

ENSI^A

FOR

DUMMIES

FROM FIRST COHORT,
TO FIRST GRADUATES
CLASS.

*A five-year
survival story*



Mohcen T. Chouireb

Unofficial edition. Not affiliated with, authored by, or endorsed by ENSIA or its administration.
“For Dummies” is a registered trademark of John Wiley & Sons. This book is an unofficial parody,
not affiliated with or endorsed by them.

A personal account of one promo’s journey through the school, written for the ones that follow.

Published 2026. Free to read and share, just please don’t sell it or pass it off as your own.

For ENSIA, the one I knew, and the one I didn't.

Chapters

Introduction	7
Chapter 0: Picking Your Path	9
Is it even worth it?	9
So where do you study it?	10
Chapter 1: Meet ENSIA	13
ENSI A	13
The Pole	14
Housing	15
Sidi Abdellah	18
How Passing Works	20
How I'm Grading the Modules	21
Chapter 2: First Year	23
Context	23
Term 1	24
Term 2	24
Term 3	26
Term 4	27
Miscellaneous	28
Chapter 3: Second Year	31
Context	31
Semester 1, Term 1	32
Semester 1, Term 2	35
Semester 2 (the new system)	37
Miscellaneous	41
What Changed After My Promo	43
Chapter 4: Third Year	45
Context	45
Semester 1	46
Semester 2	49
Building Gostu	53
Chapter 5: Fourth Year	59
Context	59
Semester 1	60
Semester 2	64
Miscellaneous	69
Finding Your FYP (and Your Binôme)	69

Chapter 6: Fifth Year	73
Semester 1	74
Final Year Project	75
Leaving ENSIA	78
The Album	81
Afterword	149
Civil Service	149
A Final Word	151
Acknowledgments	153
About the Author	155
Credits	157

Introduction



August 2025. I was out on a walk, like any other day, my mind circling the idea that this was my last year at ENSIA, and that I couldn't really remember how the past years had gone, what the checkpoints I'd crossed each year even were, or how they shaped who I became, as a person and as a professional. By the end of that walk, I'd become obsessed with retracing the timeline, winding the clock all the way back to the start, and trying to document each phase, for my own sake, not for publication or anything like that.

A few months in, with pretty inconsistent progress, I'd gathered a set of thoughts about the years that had gone by. At that point, I started thinking about the feelings I'd had through these different phases of my life, and how helpful it would've been to have someone who'd gone through the same stages, reflecting them back to me, walking alongside me on the unknown journey I set out on when I picked ENSIA. Slowly, the idea of this book took shape, as I found myself turning the notes I'd been keeping through all of it into something shareable, and finally into the version you're reading now.

ENSIA for Dummies is a walkthrough of the whole ENSIA chapter, the way I lived it and what I noticed across its levels, with one goal: to give you the sense that you're not alone, that we all felt this at some point, and to offer a hand, where and when it helps, to get you through this chapter of your life. The "For Dummies" part isn't me calling you dumb, it's the well-known idea of walking you through something from A to Z, so you come away with a real sense of it.

It doesn't matter if you're still thinking about joining ENSIA, already studying there, or just curious about it, the book goes in the same order. It starts with whether to study here at all, then the school itself, then the five years one by one, module by module, with a feel for the finals and the side quests worth chasing. A couple of detours sneak in too, the final-year-project hunt and a photo album.

As you'll see, don't expect academic language, or objective points of view. I chose to write it differently, keeping the language informal, with emojis and the odd side joke, to keep it as close as possible to how students there talk. So treat it as a piece of writing where you shouldn't stress about vocabulary, spelling, or grammar, and focus on the content instead (a very convenient disclaimer

INTRODUCTION

for my own typos, I know 🙄). It's also not endorsed in any way by ENSIA's administration (aka من الشعب و إلى الشعب). Still, if you do spot any errors, please reach out. And if you're curious who's writing this before going any further, you'll find the answer at the end of the book.

None of this means you'll grasp everything this school is about by the end, since it's just one person's account, out of many promos, and you'll only get the real depth of it by living this chapter yourself. And to you, I wish a happy read, one where you get the most out of this book, and where it helps you along the way.

Chapter 0: Picking Your Path

Is it even worth it?

The first thing you'll ask yourself, before ever setting foot in ENSIA, is whether the school is even worth it. But the answer to that question matters far less than *how* you arrive at it. So instead of handing you a verdict, I'll walk you through my own reasoning, and then offer a few things that might clear the fog for you.

For me, the pull toward computer science started a little before I sat my BAC (thanks, covid 🙌). Lockdown left me time to poke at coding and get a feel for the basics, so by the time my BAC results came out, I was already certain I'd pick something in this stream.

Now, you might be reading this a few days into making your choice, or a few years after it, quietly wondering whether you got it right. Honestly, we've all had that doubt, it comes with any decision that matters. You will always be able to imagine that option B would have turned out better than option A. You'll never fully shake that doubt, so what matters is knowing yourself well enough that the choice actually fits *you*.

That's why the single best thing you can do is try things early. Write some code. Poke at finance. Dip a little into law, into medicine. Every little thing you try teaches you something about *yourself*, and that makes every decision after it a bit easier.

Side note: this is exactly where I think our education system lets us down. It rewards parroting over real learning. But that's a rant for a whole other book.

I know experimenting early isn't a luxury everyone has, and it wasn't really available to me either. But a lot of today's tools quietly patch that gap. Take YouTube for example, and search for day-in-the-life vlogs from engineers, doctors, whoever. Don't expect that to give you the honest look at what the job really is, but you'll get a peek at it, and a peek is what's useful at that level. That's only a starting point, though, and these days there are plenty of other tools that can set you up to dig deeper into whatever catches your eye.

So where do you study it?

Picking your field isn't the last call you'll make . You also have to decide *where* to study it. In my case I was set on computer science, but I still had options: ENSIA, ESI Algiers, ESI SBA, ESTIN, USTHB, or studying MI at the university in my hometown. That's a genuinely tough decision, and contrary to what everyone assumes, it isn't settled by how high your BAC mark is. Scoring an 18 or 19 and still choosing your hometown university is a completely valid call if your reasons are there. Every option comes with its own advantages and downsides, and your job is to weigh them honestly.

The grandes écoles are where the strongest students cluster. Expect a smaller, more intense space where everyone is focused and performing. Geeking out is the standard, and being a nerd is the default. That environment isn't the right fit for everyone.

That's precisely why some people choose university instead: more time to themselves, which many treat as the real resource. Time to build other skills, start working early, or invest in things beyond class. The flip side is giving up that constant, high-pressure push, where the people, the clubs, and frankly everyone around you are trying to create and produce.

That grande école pressure can be great for self-development, but it can just as easily cost you hair, weight, and sleep to stress. That's the real appeal of a university's calmer pace. So get clear on the trade-off in each path, and pick the one your actual self is more reflected in.

And the *where*, geographically, matters too. Some people want to stay close to home, so Algiers is the last place they'd go. Others want exactly the opposite! So just make sure you account for all of it, the personal factors, not only the professional ones.

One debate you'll also see online is whether to go for ENSIA or ESI. Back then, I was dreaming of getting into ESI Algiers. I used to watch the students and their club activities online, and given my enthusiasm for CS, I saw that environment as the right fit for me. But a few days before the BAC results came out, I learned about ENSIA. It was its first year, and I understood that if I ever got accepted, I'd be among the very first promo. That idea, specifically, is what got me more excited about ENSIA. I wanted a fresh place, somewhere my impact would be more lasting and remembered, rather than ESI, where the communities were already established and it would be harder to leave your mark. A few years later, I got exactly that. We ended up co-founding a club at ENSIA, and it felt

very special to watch future promos keep joining and becoming part of the community.

NOTE

One thing that tips this comparison, as of the 2026 BAC intake, is that the pole schools, ENSIA and ENSM included, now come with a catch ESI doesn't, which is that their graduates have to commit to five years of civil service in Algeria after finishing. ESI sits outside the pole, so it isn't covered. It also doesn't affect my promo, or the ones that started before this decree, but if you're choosing now, weigh it into this decision. I will talk more about this in the afterword.

Reflecting on this, I'd say you should spend as much time weighing your options as you would revising for the BAC, because the two are equally important. So reserve some real time for it. I can't think of a better window than the one between finishing your BAC exams and waiting for the results. You won't have much time anyway once the portal opens for you to register your final choice.

So yeah, there's no perfect choice here, and nobody's going to hand you the right answer. Any of these paths can work out, or go wrong, and from what I've seen, what makes the difference is how honest you were with yourself about what you want, and how much you commit once you've picked. And please, don't spend the next five years second-guessing it, that'll cost you way more than any "wrong" choice ever could.

And if ENSIA is where you landed, or where you're seriously looking, the rest of this book is about what that looks like from the inside. So let's meet it properly.

Chapter 1: Meet ENSIA

ENSIA

As you probably don't know 🤔, ENSIA is the school Algeria created to develop engineers in software and AI. From what I can tell, it was set up largely as a political signal that the government is giving this field the attention it deserves. The goal is to produce engineers who push the boundaries of AI, whether through the jobs they take after graduation (nothing guaranteed, calm down) or the R&D projects they work on, in certain modules along the way or later during a PhD. The school also pushes an entrepreneurial mindset, nudging students to start their own companies. I don't love the word "startup" here, but more on that later, since I ended up doing it myself. None of this means these are your only options once you graduate. On the contrary, plenty of career paths open up from here.

Getting in requires around 18.50 or more in the BAC, depending on your branch. That threshold climbs year over year, so it also depends on grade inflation in your BAC year! Once you're in, expect a monthly stipend, or bourse, of 10,000 DZD during the two-year preparatory cycle, rising to 12,000 DZD across the three-year second cycle.

At first, ENSIA used a term-based system, where the year was split into four terms, two of which make up what you'd call a semester, and we followed this for the first year and a half. After that it was dropped, and we went back to a semester-based system for the rest. As for the modules, we followed a fairly modern curriculum, adapted from well-known universities, which, looking back, is another thing that sets the school apart.

Teaching happened mostly in English, since the materials were all in English, though not all the teachers followed suit. Some of them were taught, and had taught, in French their whole lives, so expect a few to lecture in French while keeping English for the "official" correspondence. Classes themselves came in a few formats you'll hear about constantly, the lectures, the TDs (travaux dirigés, the exercise sessions), and the TPs (travaux pratiques, the hands-on labs).

The Pole

ENSIA is one of five elite schools in the Sidi Abdellah Techno Pole, named after the martyr Abdelhafid Ihaddaden¹. The others are the National School of Mathematics (ENSM), the National School of Nanoscience and Nanotechnology, the National School of Autonomous Systems, and the National School of Cybersecurity. ENSIA itself is an engineering school, and if you carry on through the master's track on top of the engineering one, you graduate with a double diploma, a State Engineer degree and a Master of Science in Artificial Intelligence and Data Science. I'll get into how that works in the fifth-year chapter.



Figure 1: An aerial view of the Sidi Abdellah techno pole.

As the aerial view shows, that big open esplanade in the foreground is where students hang out. On its left is the auditorium, the domed building, where events are regularly held. On its right is the central library, the crescent-shaped one, a quiet and reserved space to read and study. Behind the esplanade, the schools stretch across the full width of the pole, and further back sit six student residences, spanning the same width. As far as I remember, each residence can hold around 1,000 students. The Students Hub is further off to the right, out of frame here, a big space with a restaurant that only serves lunch, a spot to wash your clothes, and an area where students gather and recharge.

Life at the pole is generally great. I found it a good space to blend your personal and academic life. If you're an early bird, you can hit the schools' open

¹https://fr.wikipedia.org/wiki/Abdelhafid_Ihaddaden

spaces as early as 6 a.m. and start your day. If you're more of a night owl, you can stay at the school until 10 p.m. before heading back to the dorms. The closeness of the two schools and dorms, changes everything about student life here, and I found it really handy, in the sense that it doesn't cap your productivity. Students here have the luxury of being in their own space, disconnecting for a while, going out, eating, chatting, whatever, then getting back to work. One thing I really liked about the pole is how much space there is to walk. If you're fond of long walks (like I am), you'll love it, and I'd even recommend doing it at least once a week, heading out for a walk of 30 minutes or more. It used to help me reflect, reason through ideas, and think about things I don't usually think of when I'm in the middle of the storm.

Housing

As mentioned earlier, there are six student residences, split between women's and men's dorms, and every student gets a room in one of them. That's true no matter where you're from, so even if your family lives half an hour away in Algiers, you still get a room on the pole if you want one. The one on the far left in Figure 1 is called Mahelma 06, and it's the closest to the campus entrance, on the train station and Mahelma city side. Every student from the first ENSIA/ENSM cohort remembers this dorm, since we all lived there for our very first year. It's relatively small, with a capacity of only around 500 students. The other five are Mahelma 01 to 05: Mahelma 01 sits just to the right of Mahelma 06 in the image, and Mahelma 05 is on the far right, out of frame, closest to Sidi Abdellah downtown. Not that it matters much, since all six residences are carbon copies of each other. I could drop you in any one of them and you wouldn't be able to tell which.

After one year in Mahelma 06, I spent the next four years in Mahelma 03, room C1-15 to be exact. The dorm is split into two halves. One half is the buildings themselves, where you sleep, five pavilions labeled A to E, each with a shower block underneath. Water was terrible in the very first years, but it got much better later on, becoming available more or less all day. The other half is where student life happens. There's a foyer that sells groceries and snacks, and a café that doubles as a restaurant. There's a covered indoor sports hall used for football, basketball, and handball (and yes, volleyball too). You'll also find a doctor, a study room, a library, the main administration office, a prayer room, a gym, and the residence restaurant, which usually serves breakfast in the morning and dinner at night.



Figure 2: The residence café (left) and the indoor sports hall (right).

Your room is where you'll spend a lot of time over the years. You can always change it later, depending on availability, but try to choose well at the start of the year. A few things to look for:

- **Good network coverage.** You want solid internet and call signal in your room. I found Ooredoo to have great coverage in the residence, but it can vary depending on exactly where you live. Some of us would switch pavilions and suddenly lose all signal 😊
- **Sun and airflow.** Pick a room with good exposure to both. You'll feel it in summer, when it gets hot and humid and you're begging for the slightest breeze. Sunlight matters too, since it kept me awake and alert. A fully dark room might feel cozy, but you'll probably feel demotivated through the day.
- **Electricity.** Like any building, Mahelma 03 has its quirks 😞. Some rooms have power issues where it cuts out the moment you plug in your charger. That's bad, you really don't want that.

And in case you wondered what a room looks like, here's one. It's not huge: it barely fits two people, but it's just fine for one. Yours will come mostly empty, just one bed with a mattress and blankets, a wardrobe, and a full-length glass door instead of a small window. There's a small table and one chair. You can request more, and if they don't come through, just grab any spare table or chair from around the residence (not stealing, just tell the administration afterward, since going through them directly tends to take forever).



Figure 3: An empty room in Mahelma 03.

In general, you can also bring a *résistance* to cook with and a fridge, if you want to keep food around. Plenty of people I know bring their whole setup, a bigger desk, a fancy chair, the works. That's fine, since there's plenty of space for it (assuming rooms are still individual 🙄).

As I said earlier, proximity to campus is one of the things that make ENSIA, and any other Techno-Pole school, special. So understand that this is different from what you'd get at any other elite school, and make the most of it. Your room is part of that, so make it your own, a place where you can focus, be productive, and have as much fun as possible! And when your time's up, try to leave it clean and in good shape for whoever moves in next, since the place only stays good if each of us plays their part. It's a small way of looking out for the promos coming after you.

Sidi Abdellah



Figure 4: A main boulevard in Sidi Abdellah, the “city of Legos.”

Sidi Abdellah is where you’ll hang out most whenever you’re outside the pole. As the image shows, and as you’ve probably seen, it’s a city of Legos, in the sense that it’s basically nothing but apartment blocks. That makes it a dense, social place, with upsides and downsides. The main upside I remember is the shops, where you’ll find everything, at almost any hour, and for much lower prices than in other Algiers suburbs. One thing I really recommend is walking the whole city. It’s big, but if you cover the two main boulevards and the streets around them, you’ll quickly learn the roads and the shops you’ll need during your stay (charger suddenly dies, laptop needs repairing, you need to visit a dentist, whatever, you get the idea).

An area like that, wall-to-wall apartment blocks, also has its downsides. I wasn’t a big fan of the people there. You know the kind of behavior a dense area like this can breed. Don’t get me wrong, plenty of people, especially the ones you deal with daily, like shop owners, are genuinely great. But expect to run into some bad manners too. And to women, keep your guard up a bit more, especially at night. Apart from that, it’s pretty safe, since not much went wrong during my time there. And hopefully nothing in yours either!

Getting to and from Sidi Abdellah is a mess! Unlike other parts of Algiers, which have the Metro, tram, and cable cars, this city has none of that, which can make getting in and out pretty inconvenient at times. The first option you’ll

probably use is the train². There's a station called Sidi Abdellah Université, steps from the pole, and you'll use it a lot. It's one stop from Zeralda to the west, and it can get you to Agha station in the east in around 40 to 50 minutes on average, though expect more. Agha is in the center of Algiers, and from there you can reach anywhere. The second option is buses: there's a central bus station on Sidi Abdellah's main boulevard, sometimes handier for reaching nearby areas like Zeralda, Ouled Fayet, Cheraga, or further out.

With the wonderful color palette our dear architects chose for Sidi Abdellah, both the city and the pole, it can feel pretty depressing, especially in winter when exam stress peaks. So remember to get out regularly, go see some other colors (to put it mildly) elsewhere in Algiers, and disconnect for a bit.

So that's the place, a small school tucked into a Lego city on the edge of Algiers, where you're never more than a short walk from class. For five years, this bubble is pretty much your whole world, so get to know it well, just don't let it become the only place you ever see.

²Timetables for the Alger-Zeralda line are on the SNTF site, <https://www.sntf.dz/index.php>

How Passing Works

Before we get into the years, let's sort out how the school grades you and decides whether you move up. It's what everyone stresses about, and honestly, it's nowhere near as scary as the rumors make it sound.

Each module mark is usually 40% for the work you do across the year, meaning your midterms, tests, and how you show up in general, and 60% for the final exam. Some modules run it 50/50, and a few flip it entirely, putting 60% on the year and leaving only 40% on the final. Find out which split each module uses early on, because it decides how much one bad exam day can cost you.

Every module also has what's called an eliminatory mark. The part that trips people up is that it isn't a fixed number like 10. It's relative to how your whole promo did in that module, working out to 60% of the promo's average in it (so, the promo average times 0.6). Score above that and you've passed the module. Score below it, and you'll sit a rattrapage (a resit) for that one. There's no cap on how many modules you can resit, so one rough semester isn't the end of the world.

To move up to the next year, two things need to hold. Every single module has to be above its eliminatory mark, and your overall average has to be 10 or above. If you've cleared every module's eliminatory mark but your overall average still lands under 10, you'll sit a rattrapage to pull it up to 10 and move on.

One last thing, and it's the one that matters most for the stress. You're allowed to repeat a year once during your time here, except the first year. First year is pass-or-go. If you don't make it through, it's reorientation rather than a repeat. So take that first year seriously from day one, and after it, you've got a bit of a safety net.

How I'm Grading the Modules

On top of the school's own grading, I've got my own way of flagging the modules. To keep it fun and organized, I sort each one into a tier, a rough sense of how much it deserves your attention:

SUPER-DUPER-IMPORTANT

the ones to focus on 100%. Show up for the courses, the TDs, and the labs.

RELEVANT

you can nail these just by attending the TDs and labs, mostly the math-heavy and practical ones.

MEH

a bit of revision right before the tests does it (aka the famous ليلة الرعد).

RANDOM

so random that you can pass the exam on existing knowledge alone.

That first tier, "attend the courses", is itself debatable, since not everyone has the patience to sit in the amphitheater and still absorb everything. Some prefer recordings, or other courses covering the same material, and that's completely fine, as long as you do what suits you best.

This whole thing isn't a science, just my own take. You might file a module I call **relevant** under **meh**, or the other way around, so trust your gut and give each one the weight you think it deserves. It's also not meant, in any way, to beef with the teachers or diss them. I have utter respect for all of them, and this is just feedback from one of their students.

Now let's get into what those five years feel like, year by year.

Chapter 2: First Year



Figure 5: Techno-pole esplanade in 2021, only OGs feel the vibes from this image.

Context

As you probably noticed from the book's outline, we'll cover the years one by one, and mostly semester by semester. The exception is the first year and a half (three semesters), which I'll do term by term, since that was the system back then. A term is eight weeks long. We had three or four modules per term, with midterms around week 4 or 5 and finals in week 8.

One caveat before we start: the modules I'll walk through are different from the ones taught now, in both naming and structure, and there are some they teach today that I never took (especially in first year). But I'll flag these changes whenever we hit one. Most of them are in these first two years, so don't expect many once we're past that.

I'll also try to include a miscellaneous section in each chapter (First Year, Second Year, and so on) to cover the things beyond studying that you might run into that year, like clubs, work, or other opportunities.

Term 1

These first chapters are the hardest for me to write, since I've mostly forgotten what I studied back then. What I do remember is that our first term at ENSIA was an intensive English program. We had a bunch of light modules in it, and frankly I'd file them all under **random**, since, depending on your English level in high school, you could do well on all of them on existing knowledge alone.

Of everything we studied then, the one module I still remember is ESP (English for Specific Purposes), mainly because it actually felt relevant. It taught the technical jargon, and right after it you could feel a real difference in how you expressed technical ideas.

Apart from that, we studied Writing, Reading, Listening, Translation, Creative Problem Solving and Critical Thinking (CPSCT), and Public Speaking. These were all separate modules, each on its own topic, and while we sometimes did fun ice-breaker activities, I didn't feel much benefit from spending a full eight weeks on them. Luckily, this changed with the semester system: instead of eight weeks split across all of them, they were folded into a single module called English.

So I mostly treated these two months as an adjustment period, getting used to the new place and the residence, setting up routines, and figuring out how the university works. I also explored the transport, Algiers itself, and whatever else there is to explore when you move to a new city.

Term 2

The second term is where the fun starts, and the real foundations begin to come in, starting with the math. We had three modules in total this term. The first was **Analysis I**, where we covered the basics of analysis: real numbers, complex numbers, sequences, functions, and continuity. Its color marks it relevant, since it lays down important basics you'll need later. It's also one of those modules you can do just fine in, as long as you focus on the TD sessions and solve as many exercises as you can.

The second module was **Algebra I**, where we covered the basics of algebra: logic rules, sets, relations, and algebraic structures. Again, it's a very important module, since it covers core notions you'll use later in your studies. Just make

sure you understand the content and solve plenty of varied exercises to make it through.

In both **Analysis I** and **Algebra I**, the theorems and content presented in the lectures are generally proved, and I'd strongly advise you to understand those proofs, and then try reproducing them yourself. As challenging and different as this is from the math you did in high school, it's a great way to train your reasoning and problem-solving skills. You might even get a proof-style question in the exam, so treat it as your shot at the harder marks!

Currently, first-year students take a single course called Foundational Mathematics, which, from what I understand, merges the content of both modules. So it's one broader course covering what used to be **Analysis I** and **Algebra I**.

As for revision resources I found useful, there's the well-known Exo7³. It's arguably an underrated source for revising basically all the math courses you'll take at ENSIA. They offer a wide variety of TD sheets full of exercises, each one corrected on the same sheet, and many even have an attached YouTube video of the correction, like this one on sequences⁴. I'm not sure it's still the case, but a lot of the exercises in our own TD sheets were pulled straight from it!



Figure 6: The Exo7 experience.

³<http://exo7.emath.fr/>

⁴<http://exo7.emath.fr/ficpdf/fic00010.pdf>

It's not the only resource you'll need, but it's a powerful one. Another trick I liked for finding exercise solutions was to copy the exercise text, paste it into Google, and let the crawlers do the work. The same exercise usually turns up elsewhere online, with a solution, typically in other universities' archives, alongside their exams and midterms. And if you're lucky, you'll find that your teacher drew from the same source, so you already know what the finals will look like 🤖 (yes, it happens sometimes...).

Back to the modules, the third one this term was **Academic Communication**. From what I remember, it introduces academic language into how you communicate, and trains you to write reports, papers, articles, and the like. We also had a second part of it during Term 3. I found the content pretty informative, but it was brought in earlier than it needed to be, since we had the same module, with practically the same content, in our final year (where it became relevant). The good news, students no longer take this course.

Term 3

This term was very special, as we got to study a CS module for the very first time! The one in question is **Introduction to Programming in C++** (now called Data Structures and Algorithms I), and as its color shows, this was our first super-duper-important module. It's an essential step in your never-ending (hopefully) programming journey, and it lays down basics you'll use forever. The core concept here is how you can use programming to build algorithms that solve a specific problem, and the two things that stand out are the problem and the algorithm, not the programming language. To that end, the module builds up the basics of algorithms through conditions, loops, functions, recursion, and so on.

It uses C++ rather than another language because of how close it sits to the hardware. Some problems can be solved in other languages like Python with a magical one-liner, but a beginner wouldn't fully grasp how that solution works, or how the program executes it. A language like C++ gives you the flexibility you need, and more visibility into how the system operates.

If you've never programmed before, no need to panic, since I found this module really beginner-friendly, accessible even to people who had never touched a computer! My one piece of advice is to spend as much time as possible programming, solving practice problems for the chapter you've just covered, whether from the lab sheets or online (the likes of GeeksforGeeks⁵). Once you

⁵<https://www.geeksforgeeks.org/cpp/cpp-exercises/>

feel you’ve grasped loops, for example, you can figure out how to combine them with conditions to approach other kinds of problems, and you keep building gradually until you reach more advanced topics. One thing I found helpful is to write your solution down with pen and paper before moving to your laptop to code it. That gives you more time to order your ideas and reflect on better solutions, especially when you hit harder problems later on.

And since some readers of this book will be learning to code in the age of AI, I’d ask you, please, not to lean on any coding agents at this stage to write solutions for you. At this stage, you need to know every aspect of the program you’re running, the syntax, the logic, everything. Save the AI tools for later, once you’re at ease coding projects by hand, which will come in the following years. You might feel you’re moving more slowly than your mates, but for now, just focus on yourself and your own growth, so you build a really strong foundation for whatever comes next.

I’ve probably said enough by now, but this is one of the key modules for grasping the mindset this field demands, and for training yourself to solve all kinds of problems just by spotting recognizable patterns. Spoiler alert, that’s pretty much what you’ll be doing for the rest of your career!

The second module we studied this term was **Analysis II**, the second part of the one we did the term before. This time you’ll cover functions in greater detail, Taylor expansions (what we call DL, *développement limité*), integrals, and differential equations. But I didn’t find the way you approach it any different, so solving as many exercises as you can should be enough to get you through it.

The third module was **Academic Communication II**, which I already talked about back in Term 2.

Term 4

In the last term of first year, the school kept up the programming momentum from the previous term by introducing **Object-Oriented Programming (OOP)**. In this course, you’ll learn your first ever programming paradigm, and by paradigm, I mean a formal style that guides how developers conceptualize and implement their programs. OOP is one of the fundamental ones, and learning it helps you zoom out from the little programs you wrote earlier to something more global, closer to real software. You’ll also get to build some very cool projects as part of the curriculum, obviously in C++, and you’ll see how a program can be spread across many files that you compile together into your

final work. It will push you to practice your problem-solving and coding skills even more, so put in the time if you want to really master it.

We had one more math module this term, and this time it's **Linear Algebra's** turn. It's the successor to the first Algebra course, taking you from vector spaces to linear maps and matrices, then on to eigenvalues, eigenvectors, and orthogonality. It lays the groundwork for key concepts you'll meet later in the field. In reality, though, I found it ultra-focused on drills, meaning how many exercises you can solve, warming your brain up to do as much calculation as you can (rightly so). Even so, you should also attend some lectures, or at least try to reproduce some of the proofs done there. They'll add more depth to your understanding and will certainly help you later in more advanced courses. And prepare yourself to find the eigenvectors of the 5-by-5 matrix full of 3-digit numbers you'll get on the exam!

Rounding out the term was **Probability and Statistics I**, one of the most fundamental modules you'll ever study and one you'll keep running into for the rest of the degree. It covers the probability basics, random variables (discrete and continuous), expectation, and variance, along with the basics of statistics. All of that is an introduction to what comes next year, where the heavier material, distributions and inference, gets a dedicated follow-up (more on that when we get there).

One thing worth flagging is that there are also newer modules taught now that I never took, since they came in after my promo, such as Information Technology Essentials, and Introduction to Linux.

And that's first year done. Looking back, the math, your first real programming, and even the English modules were mostly about laying the ground for what comes after. I know a lot of it won't feel like the "AI" you signed up for, and some of it will feel plain boring, but almost all of it comes back later, usually when you least expect it (yes, those eigenvectors will find you again). So don't write a module off just because it doesn't excite you right now.

Miscellaneous

In general, you'll find the first years the most challenging of the five, and the very first the most extreme. The content itself gets harder later on, but everything here is new to you, the material and the whole system around it, and that's what makes it tough. You'll learn how to approach lectures, how to get through TDs and labs, and how to revise to get the best out of each module. So on top of adapting and actually studying, you'll find yourself juggling a lot

of self-management, which is what makes it feel hard. If there's one thing to remember, it's to keep a cool head, trusting your abilities and building on them to get through your modules. And when you hit a wall, break it into smaller problems (per module, per chapter) and tackle each one on its own. Before long, you'll gain confidence and feel yourself making progress.

Outside of studying, you'll also have time to explore all sorts of activities, clubs being the obvious one. There are already plenty at ENSIA, so take some time to attend their opening events, where they usually announce their team and their program for the year. Then you might want to pick one, or two at most, and join their workshops and events. You'll learn most of your skills outside the curriculum, and club workshops and events are a great way to do that, as well as to meet people who are into the same things as you. I wouldn't advise taking on responsibilities within a club in your first year, though (if you ever get the chance), since you'll burn out from how much you'll have to take on. So for now, just focus on your own learning.

Throughout the year, and as a direct consequence of the elitism that packs people with top BAC scores into a single promo, you'll find yourself comparing with everyone around you, either explicitly, by checking their marks against yours, or implicitly, by counting the hours they spend in the library or the amphitheater. That's unhealthy, and it can breed bad feelings toward your mates, like envy. So pay attention to your own progress instead, regardless of others and whatever they're doing, and make the most of your own time and effort. I know some people really are more motivated by comparison, and that's fine if you're one of them, but they're the minority, and I wouldn't treat it as the default to adopt.

On the same theme, I've noticed a steady rise in social-media content creators around ENSIA. I've got nothing against them, they're free to post whatever they like, but with the trend of dressing everything up to look aesthetic for the feed, I do want to stress the importance of keeping it real. By that I mean the learning experience usually won't look aesthetic. It'll feel more turbulent and headache-inducing 😬, because that's what genuinely learning something new tends to feel like. So keep that in mind when you scroll past your mates' posts and reels. You're not doing it wrong, you're just living the real thing, which might hurt, but pays off if this is a field you care about.

One more thing you'll notice about the mentality here is that people are always restless, always chasing the next phase. So it's totally natural to feel that in summer, when you're out of studies and feeling a bit left behind by life. But

remember that summer is your time to think, since there's little room for it during the year. Use it to reflect on where you're headed, what you actually like, and what you want to try next. These stretches of self-reflection are what give you clarity about yourself and what you want to become.

You can also spend some of it pushing your coding skills further, or better yet, pick a domain that caught your interest during the year (web dev, mobile dev, whatever) and start building around it. Over time, you'll find yourself making mini-projects and pushing your boundaries.

Before we move on, here is the whole first year in one picture: every module, grouped by term and colored by how much attention it deserves.

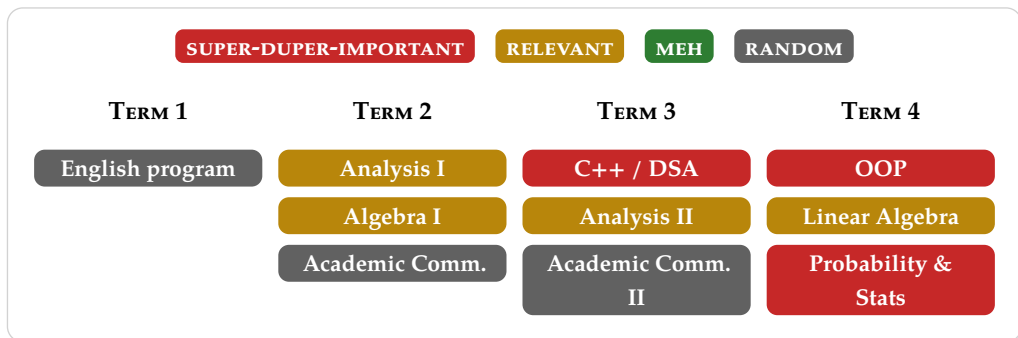


Figure 7: First year, module by module. Notice how it heats up from gray to red.

Chapter 3: Second Year



Figure 8: ENSIA/ENSM 2023

Context

On to the second year now, which we'll approach the same way we did the first. It's also the year the terms became semesters. We studied the first semester the old way, term-based, with four modules in Term 1 and the same in Term 2. Once we got into the heavier modules and felt we were moving too quickly through some of the fundamentals, with too little time to grasp the material, we asked the administration to rethink the system. That request didn't move on its own, so the promo went on a week-long grève, with delegates picked to carry the students' case and sit down with the administration to negotiate something better. That's worth knowing early, because a promo that agrees on something concrete, and goes about it the right way, does get heard here. The change landed in the second semester, when the terms were canceled and merged into one long semester, with more modules (seven or eight) but more time to sit with each one.

That change had a downside too, in my opinion. I could feel the overall pace drop. With a whole semester to study one module, you can end up cramming it into the final weeks, or even days, before the exam. That wasn't possible before, when time was short and the terms were far more focused. So, semester by

semester, I gradually started feeling a bit detached from studying. At the same time, the new system freed up plenty of time for other things, like club work, hackathons, internships, side projects, and more.

I also found this the best year I had at ENSIA, mainly because of some of the modules we studied, which opened doors in my imagination and let me see the field, and the industry as a whole, differently. Then again, I'm not sure it's still the case, because some modules have been removed or added since our promo (I'll get to those when we reach them), which might change how the year looks from a student's perspective now. Either way, the general consensus is that this year matters a lot, so focus on its material, and the years that follow get much easier.

Semester 1, Term 1

To kick off this year, we start with **Analysis III**, which continues the analysis modules you studied throughout last year and is the last one you'll have to take. It focuses mainly on multivariable functions, since that's what you'll deal with most in the AI and ML courses to come. It teaches you how to compute their gradients and find their local and global minima, and it also covers integrals and differential equations. By this point you'll already know that solving plenty of exercises should get you through it, and the resource I mentioned earlier (Exo7) still does the job!

The second module was **Data Structures and Algorithms** (now called Data Structures and Algorithms II, since last year's C++ course became DSA I), one of the foundational courses every CS student goes through. It's essentially a complete walkthrough of the classic algorithms and data structures used to solve problems. You'll learn how to take a problem, analyze and reason through it, and pick the data structure or algorithm that solves it with the least time and memory. Along the way you'll meet lists, stacks, queues, trees (not the green ones, unfortunately), heaps, graphs, and a handful of well-known sorting algorithms.

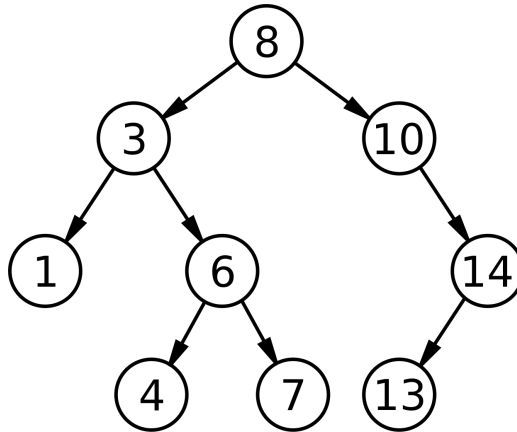


Figure 9: A binary search tree, one of the classic structures you'll meet.

When I took it, I found the content well structured. We covered the theory in lectures, and even though it got overwhelming at times, given the pace and the level, it was properly backed up by tutorial classes to practice it and lab sessions for the hands-on coding side. I found that this course really sharpens your problem-solving skills too, because you'll realize it's all about spotting which bin the problem in front of you falls into, then using the right tools the course gives you to solve it as efficiently as possible. Expect a few headaches, but by the end, you'll catch yourself sorting every problem you see into one of those bins.

I'd also really stress backing up your understanding with visualizations and animations. You need to see a data structure in action, or an algorithm step from one iteration to the next, to really grasp why these tools exist and what problems they solve. It also adds real depth. Once you can picture these tools, you can start imagining and getting creative with how you use them, which, after all, is the whole point. For that, there are plenty of visualization sites, and the one I remember using is VisuAlgo⁶.

Practically speaking, I recommend putting a lot of time into the labs by re-implementing the structures and algorithms yourself, without leaning on any library, which will push your coding and C++ skills even further. If you can, also take a look at competitive programming contests. They can get pretty challenging, though, so stick to the more accessible ones, and to problems that only cover the topics you've studied and more or less mastered (so you don't depress yourself 😊).

That was a lot on **DSA II**, but hopefully worth it, given how important the module is. Now we move to another CS course, **Computer Architecture I**, which

⁶<https://visualgo.net/en>

you study for the first time here. It's mainly based on fundamental hardware concepts, starting from Boolean algebra, which you'll need to understand and build your own circuits. You'll also study computer organization, specifically the von Neumann model, and the computer's main components. Generally speaking, I found that if you practice well in the TD sessions, you'll be fine in the exams. As a side note, I'm not sure it's always the case, but I remember our midterms and exams in this course being time-consuming, given how much calculation they required. So train yourself to calculate faster, and pick up a few tricks to reach the answer quickly, especially on the number-system problems (binary, octal, and the rest).

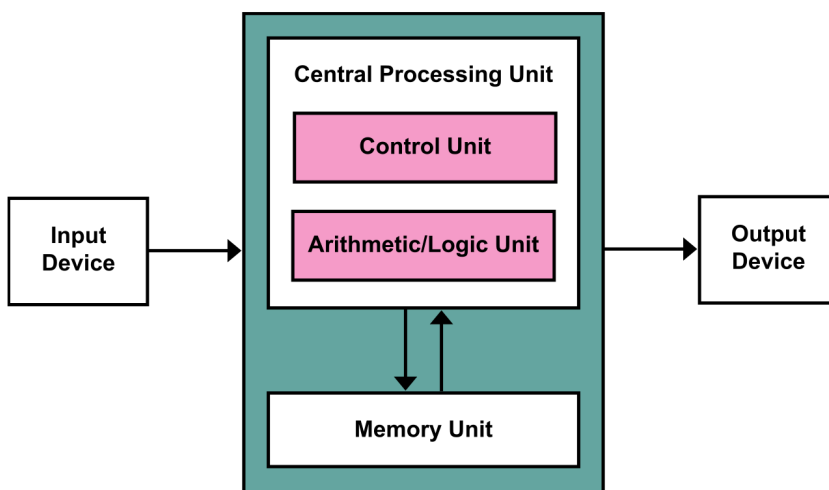
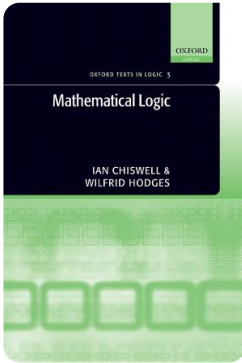


Figure 10: The von Neumann model.

The last one for this term was **Mathematical Logic**, and honestly, it's one of the trickiest, not hard exactly, but challenging, because it's so different from the math modules you've seen so far. It covers the fundamentals of logic, taking you from basics like truth tables and propositional logic, building up to predicate logic, and teaching you how to handle propositions, proofs, and formal languages. At first glance (and even years later), it might seem like a purely abstract course, which, in part, it is, given how foundational its approach is. Even so, you should see it as a tool that gives you a language and framework for reasoning about the systems you'll use and build later. Its concrete uses range from compilers checking that a program is syntactically correct, to systems verifying that a given condition always holds for them to work correctly.



Back in my time, this course was taught primarily from a famous book by Ian Chiswell and Wilfrid Hodges called *Mathematical Logic*⁷. Everything was drawn from it: the lectures, the exercises, all of it. So rely on it as your primary source for revision. You should also stick to whatever resource the lectures use, mainly for notation, because, weirdly enough, in this corner of math everyone seems to invent their own notation, all expressing the same concepts in different ways. Mixing them will only make an already confusing module worse.

Semester 1, Term 2

Next up was the second term, again with four modules. We start with **Linear Algebra II**, which has since been dropped from the curriculum, so you won't be taking it at all. I'm still covering it here, because the material itself didn't go anywhere. Essentially, it continued the matrix material from the first Linear Algebra course, going into more detail on orthogonality, QR factorization, symmetric matrices and where they matter (like in singular value decomposition), and the Jordan form of a matrix. All of that comes back later in the AI modules, so even without a course walking you through it, it's worth picking up on your own the first time you run into it. You just get to do it without a matrix-heavy exam hanging over you!

The next module was **Probability and Statistics II**, the follow-up to Probability and Statistics I from first year, and this is where things get properly involved. On the probability side you'll go beyond the basics into distributions (binomial, geometric, hypergeometric, and more), then push deeper into statistical inference, including estimation and both parametric and non-parametric hypothesis testing. It wasn't just the volume of content that made it tough, but the content itself, which took me a while to really understand.

Expect the midterms and exams to be tough and out of the box, not the kind you can ace just by grinding exercises, since the teachers who run it tend to test your understanding rather than rote practice. Still, no need to panic. As with DSA, lean hard on visualizations and animations, it's becoming a theme, I know, but seeing a distribution or a hypothesis test in action makes the tricky parts click far faster, and you'll get through it just fine.

⁷https://books.google.dz/books/about/Mathematical_Logic.html?id=JeUDUWYD5eQC

The third module you'll take is **Complexity Theory** (now called Theory of Computing, and taught in the second semester in today's curriculum). This is one of the most important courses you'll ever take, and one of the most challenging too (the hardest, for me). Here, you'll learn the abstract foundations of computer science, the ones independent of any particular technology, and tackle some simple-sounding but fundamental questions about what can be computed and what can't, and how efficiently it can be done. The course starts with automata theory, studying regular and context-free languages, where you'll notice similarities with **Mathematical Logic**, in how you define a language and use its rules to build things up. From there it moves to computability theory (decidability and reducibility), then to the complexity of problems and the classes P and NP, ending with NP-completeness.

As you can see, this is about as abstract as CS gets, and honestly, I don't have any real tricks here beyond attending lectures regularly and doing your best with the TD worksheets, especially the optional exercises that aren't covered in class, since those tend to show up again, in different forms, in the midterms and finals. This is one of those exams where you could walk out with a zero if the teacher wanted. You could grind until the last moment and still realize you hadn't understood a thing, given how challenging and theoretical it is. It's one you'll remember for the rest of your life (hopefully), and a fun one too, with important ideas to take away.



Figure 11: The Complexity Theory experience.

The last module this term was **Databases**, your first introduction to the world of databases, starting with why they're needed, what systems and structures exist, and then a focus on relational databases, covering core concepts like the Entity-Relationship model, normalization (1NF, 2NF, 3NF), and how databases handle concurrency. You'll use Oracle as the engine, with a TD session for the

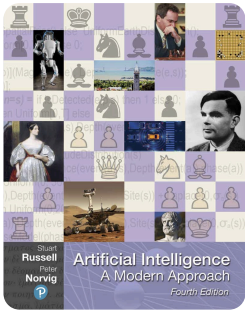
worksheet and a lab session for the practical side, mainly learning SQL and putting the material into practice.

I remember the exams having three parts: a lab part where you write actual SQL queries, a TD part with worksheet-style exercises, and a lecture part on the theory from class. The slides weren't much use, though, since the lecturer taught mostly from his own knowledge and often strayed from the official ones. As for the rest, if you practice enough, you should ace this module smoothly.

Semester 2 (the new system)

This second semester was our first real taste of the semester system, running across the second half of the year. To warm things up, I'll start with the first AI module you'll study at the school, **Introduction to AI**. In case you were eager to start training your neural networks and applying what you picked up from online courses, chill out, since this course is mostly about the foundational work behind AI. It starts as basic as defining an agent and how it behaves, interacting with the environment it operates in, the set of actions it can take, and the states it can be in. The module also involves a lot of search algorithms, spanning uninformed, informed, and adversarial search. It tackles planning and constraint satisfaction problems (CSPs), teaching you how to define and then solve them. Lastly, I remember studying knowledge representation, and how you can use first-order logic (yes, the one from **Mathematical Logic**) to reason over logical statements, mainly through techniques like unification, lifting, forward and backward chaining, and resolution. So it's mostly a foundational course, but an important one for understanding how AI got to where it is today.

Personally, I found it a fast-paced course, with lectures, tutorials, and labs (in Python ). On top of that, we had two mini-projects over the semester and periodic quizzes (three in total). One thing I noticed is that the module leaned more on the ideas of AI and on learning to reason about problems than on coding, so the labs felt like an afterthought, often little more than re-implementing the search algorithms from **DSA II** in a different language.



This course is based on *Artificial Intelligence: A Modern Approach* (AIMA)⁸, adopted worldwide as an introduction to the theory and practice of AI. It's a great place to find extra exercises to test your understanding as you go, and it has an active open-source community that maintains implementations of the book's algorithms, and much more!

And keep this pattern in mind for the rest of the degree, since plenty of ENSIA modules lean on a well-known book like this one, or shadow a widely recognized program taught at universities everywhere. Whenever that's the case, you inherit everything around it, past exams, quizzes, and the student repos maintaining its solutions, all handy for revision. I'll make sure to flag the source whenever there is one.

Then there's **Software Engineering**, another essential module, this one building the software-engineering side of your curriculum by teaching things about the industry that go well beyond just coding. It covers software processes, requirements engineering (how to draw the most out of your client to figure out what to build next), architectural design and implementation, and finally software testing (TDD based).

It's genuinely one of the best courses the school offers, because the content is so relevant that I still find myself using what we covered in class on a day-to-day basis. The biggest benefit is how it shifts the way you see programming, from a single coder's view to many, including the client's (the most important one in a product's lifecycle, as you'll learn), the project manager's, the tech lead's, the developer's, and more. You'll also spend the whole semester on a project, putting what you're taught into practice and, most importantly, working in a team, to get a real feel for how a project plays out.

In general, whether at ENSIA or elsewhere, I get that some courses you just want to skip through, doing the essentials and passing. But for this one, take the time to really grasp it and practice, both conceptually and in code. You'll see plenty of projects over the semester, so try reproducing them yourself, and do the optional exercises too!

Back to computer basics, this semester we had **Operating Systems**, a course about how a computer is run by its operating system. It starts by re-introducing some hardware components like the CPU and RAM, before moving on to

⁸<https://aima.cs.berkeley.edu/>

processes and how the OS manages and synchronizes them, raising fundamental topics like race conditions and mutual exclusion, then memory management, and finally OS security. It gives you real visibility into how a computer works.

Better still, it's backed by tutorials and labs in Java and C on GNU/Linux to help you absorb the material. On top of that, you'll have quizzes right after each lecture to check your understanding, so I found that attending lectures regularly made things clearer rather than more overwhelming. You'll also get a question box where you can drop your questions and get answers in the following lecture, and a course reader, which is a guide packed with extra examples, exercises, and resources to round out the experience. Honestly, huge kudos to the instructors for how they organized the material, which I found super helpful to study alongside.

More on computers, we come to **Computer Architecture II**, the second part of **Computer Architecture I** from the first semester. Note that the two are taught as a single module now. This part covers how a program gets executed, from the high-level view down to the instruction set, working through all the transformation levels in between. It's backed by lab work built mainly around the RISC-V instruction set, plus tutorials to practice the lecture content.

As a general tip, make sure you really understand how circuits work, something you'll have picked up in the first half of this course. That'll make your life much easier in the finals. Beyond the tutorials, there's also a resource by Nouredine Roussafi⁹ with plenty of exercises to practice from.

Stepping out of the computing bubble and back into the mathy modules, the kind you'll actually start to miss as the years go by, we have **Operations Research**. Here, you'll see the practical side of mathematics, how it's used to solve real-world problems. The module splits into two parts. The first is linear programming, an optimization method for a linear objective function under linear constraints. That sounds abstract until you see what it's for, which is getting the best outcome out of limited resources, the kind of thinking behind allocating a budget or working out a schedule. The second is graph theory, which gives you another angle on problems involving networks of all shapes and configurations. You'll have met graphs already as a data structure back in **DSA II**, and this is where you get the mathematics underneath them. You'll also have a tutorial where you solve exercises and practice the material.

⁹<https://electroussafi.ueuo.com/?page=sln>

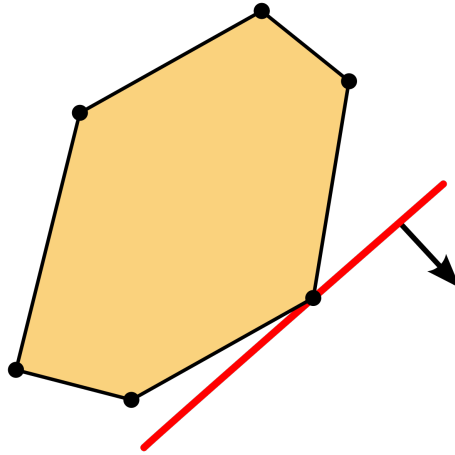


Figure 12: A linear-programming feasible region (shaded), with the optimum at a corner.

As a general note, this module always confused me. It's the kind where you feel confident you've mastered it and are sure you'll ace the finals, only to be blindsided by the exam. And just like **Complexity Theory**, it's a tricky one whose exams can go out of the box pretty easily. So my advice is to test yourself on the theory, not just the practical side, especially in graph theory, to make sure you've got a firm grasp of it.

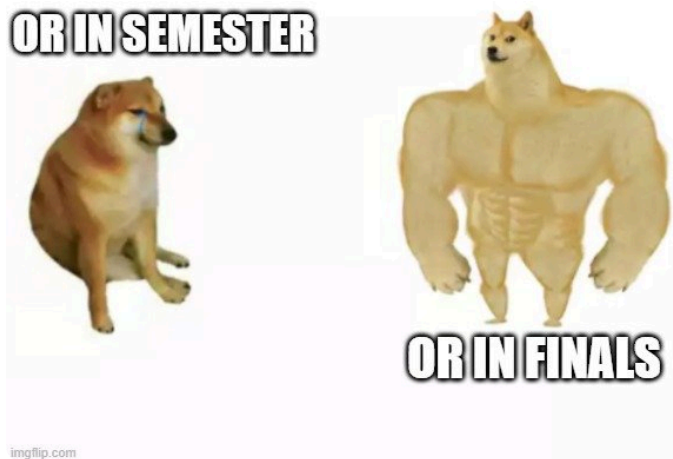


Figure 13: Operations Research, semester versus finals.

At this point, if you've been reading consistently from the start, you're probably a bit bored of module after module, but we only have two left for this year, so be patient with me 🙄.

Next up is **Web Development**, where you'll learn to build websites, from the basics of HTML, CSS, and JavaScript to more sophisticated work with PHP and

front-end frameworks like React. We also got to work on a project to practice what we'd studied.

As much as I like web dev, I filed it under **meh** because lectures aren't all that necessary for such a practical module, since you can get a long way with it, especially if you've already picked some of it up through club activities or simply on your own. The labs were worth attending, though, since they gave us dedicated time to work on the project as a team.

And now that this course is taught in the AI era, it matters even more to write the code by hand if the material is new to you. The exams demand it anyway, but more importantly, you need to do it at least once, ideally early on, to make the learning stick. I said this back in the earlier programming courses, but it holds here just as much, and in the ones still to come.

The last one is **Introduction to Business**, your gateway to becoming the next rich businessman 🤪. Honestly, I don't remember exactly what we studied here, so I can't say much about it with confidence. Still, I'm sure you'll do just fine if you revise the slides a few days (or hours) before the finals.

That said, I was rarely satisfied with the material, or the way it was delivered, in modules like this that deal with business, management, and the like. I don't think they work as lecture-based courses taught in the same format as the technical modules. They should be highly practical, since as engineers we mostly need to know how to use this stuff in practice. That wasn't really the case here, or in the business-type modules that come later.

Miscellaneous

First of all, if you made it past this year in one piece, congrats! You should be proud of yourself. It's a year packed with modules, where you get to explore math, CS, and even AI, each at a different depth.

Try to spend what little free time you have building your skills in parallel. I'd lean toward club activities, since they help a lot, and by now you'll know enough to follow even the advanced workshops if you'd like. And if clubs genuinely interest you, consider taking on some ownership, leading a cell, a department, or even a whole club, or taking the initiative to organize events, teach a workshop, or more, whether this year or in later ones.

I've done quite a lot of that during my time at GDSC (now GDG)¹⁰ and ETC¹¹, and those were what pulled me out of my own bubble, and let me meet great

¹⁰<https://gdg-ensia.vercel.app/>

¹¹<https://etc-club.vercel.app/>

people, collaborate on cool projects, and build some of the most unforgettable memories. Those two are just where I happened to be active, and the rest of the school's clubs, whatever they're oriented toward, can do the same for you. And you'll quickly find yourself learning a huge amount about managing people, and management in general.

One other activity that can sharpen your skills in a short time is hackathons, and you'll run into them a lot, whether at ENSIA, around the pole, or at similar schools. For those who might not know, a hackathon is an event where you spend a day and a half, or two, usually over a weekend, trying to solve a set of technical challenges, build an MVP, or just draft a simple prototype. Winners are usually well rewarded with prizes or money, depending on the hackathon, and I've noticed sponsors getting more and more generous with that.

The point is that even if you don't win, here or later on, you still get to practice your hard skills, especially at heavily technical events, and your soft skills too, since you might end up pitching your solution to a jury, which forces you to get better at speaking in front of people. After an event like that, you'll feel exhausted if you gave it your all, but you'll find you covered a lot of ground in a short time, thanks to the pressure of the competition. So whenever you get the chance to join one, do, some of your best memories will probably come from there!

Just as I said back in the First Year miscellaneous section, summer is the one slot where you get to think and reflect on yourself, so make sure to use it wisely! Use it to explore, or to build your skills, whether in coding or anything else, whatever you're into at that point.

You could also try landing a summer internship, preferably a remote one so you can do it from home, where the work tends to be on the lighter side. They're usually dev roles, web dev, mobile dev, at a very junior level, where you won't do much real work or decision-making, so treat it mostly as a chance to sharpen your skills. I'll be honest, there aren't many of these around (depending on your level, of course), and a paid one is certainly harder to land. Still, keep your hopes up and look for the best opportunities you can find!

Same as last year, here's the whole of second year in one picture.

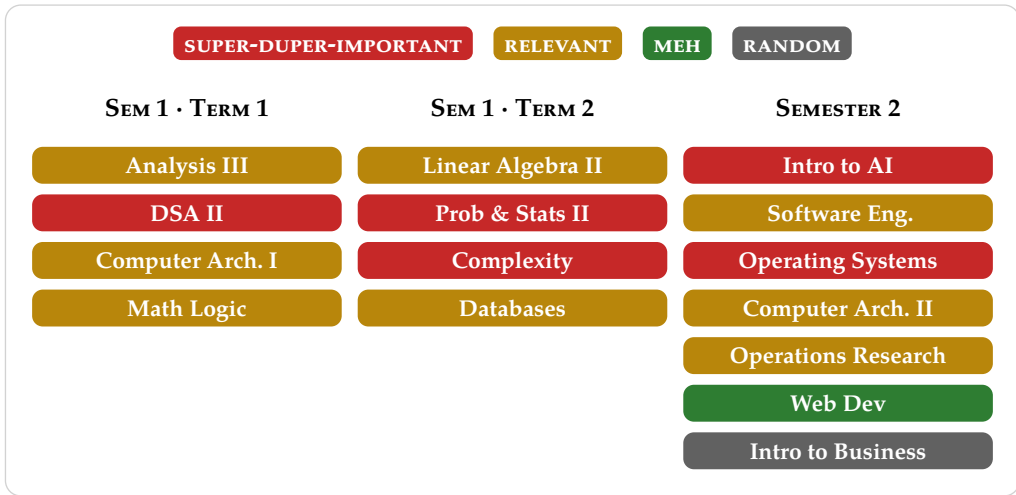


Figure 14: Second year, module by module.

What Changed After My Promo

That's both preparatory years behind us. Along the way I kept flagging modules that have been renamed, merged, moved, or dropped since we went through, and since they're scattered across two chapters now, here they all are in one place.

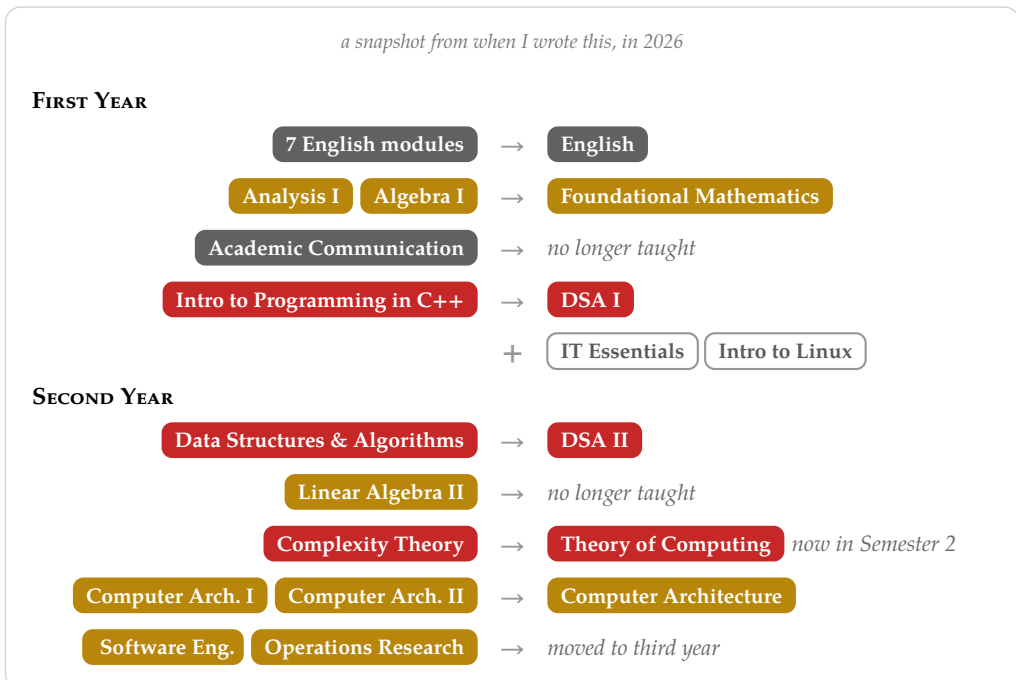


Figure 15: Every curriculum change I know of since my promo.

The outlined ones, though, are modules I never took, so I can't give them a tier the way I did the rest. And it really is a snapshot rather than a fixed truth, since

the school clearly keeps moving things around, so check the official curriculum for your own year before you plan anything around it.

What changes far less is the content itself. Even where the name on your timetable is different now, the walkthroughs in these two chapters should still map onto what you end up studying.

Chapter 4: Third Year



Figure 16: ENSIA, the newer one.

Context

Coming into third year, you'll start to feel how fast life moves, and realize how quickly you got here, from your BAC year through first and second year at ENSIA. You won't be able to resist thinking about how close you are to finishing school and moving on to whatever's next...

We'll approach this chapter the same way as before, two semesters of modules, then a Miscellaneous section for the things worth considering this year. I'll also spend some time documenting my experience building a startup, and how my team and I got into that entrepreneurial path while studying this year. I find it relevant given the entrepreneurial spirit the school pushes, and the fact that I saw many students already joining the incubator and building their own solutions.

A general trend I noticed this year, and in later ones, is that students have more of a tendency to detach from studying, now that they're familiar with

the school's pace and its rhythms (when to study, when to push harder, when to ease off). So they start looking for opportunities outside the classroom, like entrepreneurship, internships, or research projects. That's something I'll try to document, whether in this chapter or later ones.

And speaking of internships, in the summer between third and fourth year, you'll be required to do one that counts toward a module next year. I'll talk about it properly in next year's chapter, since it technically belongs there, but I wanted to mention it now, in case you're following the book's timeline chronologically.

Semester 1

For our promo, this was one of the lightest semesters we went through. The reason is that we had only five modules instead of seven, mainly because of an earlier module reshuffle (from the switch from terms to semesters), so we ended up not taking **Software Engineering** and **Operations Research** this year, since we'd already had them the year before.

Kicking it off with **Data Mining**, which is really your first data-centered module, since it focuses on handling incoming data, analyzing it, and preprocessing it with various techniques to make it usable downstream. You'll also learn how to approach the standard prediction tasks, like classification, regression, and clustering.

As an AI student, you'll find this course really fun, since you'll see all kinds of problems and data, and each time you'll try to work out what the data is really telling you. You'll also get to practice most of the content in the Python labs, with periodic quizzes on the lectures to make sure you're keeping up, so it doesn't all pile up in the final days of the semester.

Throughout the semester, you'll work on a course project, where you can either propose your own topic, something that interests you and your group, as long as you have the data (or a way to get it), or take a ready-made one, where the data and problem statement are provided by the instructors. Lately, they've started partnering with companies to turn this into an internship for students, so it's two birds with one stone, you get the project done while working on something real in the industry.

One thing I really liked was how creatively the course was taught. The labs were really useful if you wanted more practice with the lecture material, and even the midterms and finals were built from exercises that pushed your analytical thinking, rather than just testing the practical side.

Then comes a mathy module, **Stochastic Modeling and Simulation**, a course about modeling systems that evolve randomly over time. It starts from fundamentals you gained in the earlier probability modules, namely random number generation and probability distributions, then builds up through discrete-time stochastic processes like Markov chains, continuous-time processes like Poisson processes and queuing systems, and finally Monte Carlo simulation techniques.

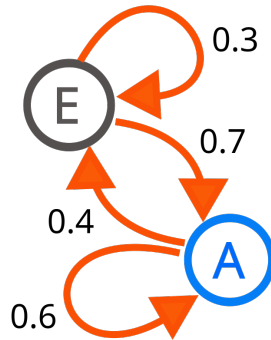


Figure 17: A simple two-state Markov chain, with the probabilities of moving between states.

Your first impression might be “what on earth would I use this for?”, which is fair, but it’s essential the moment you want to simulate and evaluate real-world events. You’ll stumble into it, for instance, when you have no data for a problem and end up generating synthetic data instead, which is just one of the use cases you’ll run into.

You’ll see the practical side of this course in the labs, which are taught in R (yeah... R ) , so that’s another programming language to add to your CV. And it’s a very practical one for statistical computing and data visualization.

One thing I used to find confusing is that I didn’t know what the finals would look like, since there aren’t many comparable modules at this level elsewhere to go by. It turns out the worksheets aren’t all coding, and it’s those non-coding, worksheet-style exercises that make up the exams, since the finals and midterms don’t involve any coding. I also found that a big part of the finals comes from the lecture material itself, whether theorems to prove or similar content. So revise the lecture content as best you can, to raise your chances overall.

Now we come to **Networks and Protocols**, a bit of a different beast, since it introduces the principles that let computers communicate over local and global networks. It starts from data transmission over physical links, moves up to the Internet Protocol (IP) layer, with notions like routing, NAT, and fragmentation,

then to transport-layer protocols like TCP and UDP, and more. The goal is to teach you how the internet really works, and how data gets exchanged.

Throughout the course you'll have both TDs, where you solve exercises to get to grips with the content, and TPs, where you apply the concepts using networking tools. In particular, you'll work a lot with Cisco Packet Tracer, using it to configure hosts, switches, routers, and the like. And you'll have some fun analyzing MAC addresses, creating VLANs, and setting up inter-VLAN routing.

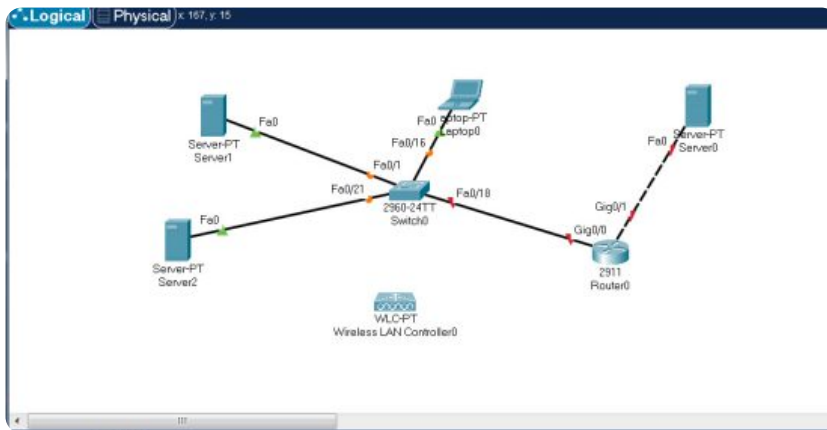


Figure 18: A small network built in Cisco Packet Tracer, from howtonetwork.com.

The finals and midterms are no different from the worksheet content you see regularly, so grinding through those will see you through just fine.

There's also **Mobile Development**, where you'll learn what you need to build mobile apps from the ground up. It goes over the basics, like usability and user-interaction design, then turning that design into a working app, first with Java/Kotlin and Android Studio, then with Flutter, where you'll pick up more advanced concepts like handling state and linking to third-party APIs to build something useful, all the way to testing your app and publishing it on the stores.

You'll get to practice every week in the labs, and you'll have the whole semester to apply it all on a group project. You pick an idea you care about, because I noticed this one is more about the idea than the code, then build it and pitch it at the end of the semester.

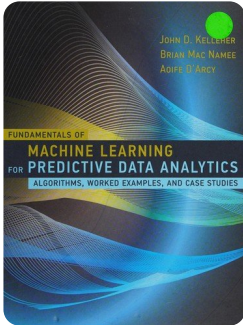
That was a fun one, nothing much more to add, except that it's very hands-on, so practice as much as you can.

Last one here is **Entrepreneurship and Innovation**, which tries to frame the innovation process and how to operate as an entrepreneur while building

something. That’s about all I remember from it, which is a bit ironic given what I spent the rest of this year doing. More on that at the end of the chapter.

Semester 2

Coming into this semester, we start with **Machine Learning**, one of the most essential modules you’ll take at ENSIA. It’s an introduction to the fundamental algorithms, with a big focus on their underlying principles, how you can use them, how to validate their results, and how to keep them working over time.



It’s primarily based on the textbook *Fundamentals of Machine Learning for Predictive Data Analytics*, so it follows the book’s division of ML algorithms, from information-based learning (decision trees, random forests, and so on), to similarity-based learning (for clustering problems), to probability-based learning (the Naive Bayes classifier, Bayesian networks), to error-based learning (linear regression and the like).

You’ll also touch a bit on deep learning, through backpropagation and how it’s used in neural networks, applied to CNNs and RNNs as examples. The final part covers evaluation and some advanced concepts like concept drift.

Overall, it’s a very rich course, one that needs your active involvement to really pick up. It supports you through tutorial sessions, where you practice the concepts from the lectures, and labs in Python, where you code up the different learning problems you come across. You’ll also have periodic quizzes to keep things fresh, and a semester project whose topic and dataset are assigned by the instructors. As you can tell, it’s a lot to take on this semester 🤖, but it’ll push your knowledge and skills a good deal further.

Coming next is **Time Series Analysis and Classification**, a whole course on time-series data, split into two parts. The first covers time-series analysis, introducing some well-known time-series models and touching on topics like time-frequency representation. The second covers time-series classification, which leans more into pattern recognition and detection, with a few other techniques for approaching classification on this kind of data.

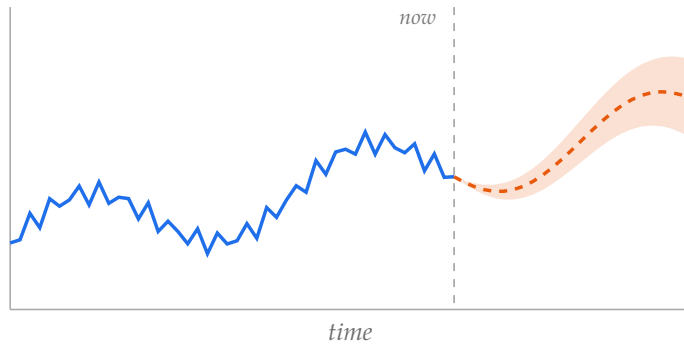


Figure 19: A time series (blue), with a forecast into the future (orange dashed) and its uncertainty band.

In this course you'll have problem sheets with exercises to solve, and coding-based ones that help you visualize and understand many of the concepts you come across in the lectures. You'll also work on two projects. The first is individual and focuses on the first part of the course, where you find your own data, from any source, ideally a long series with plenty of observations, so you can apply the analysis and techniques you've learned.

The second focuses on classification, is provided by the instructors, and you do it in a group. Here you focus on building a time-series classification model that can accurately predict the class labels of unseen instances. For both, you won't have too much to worry about, since you're given all the tools you need to make them work, so make the most of both.

Next is another mathy course, **Numerical Methods and Optimization**, where you'll learn how established numerical methods are applied to solve all sorts of problems. It opens with the notion of error and how approximate methods work to reduce it, then moves on to solving linear systems, optimization techniques, and finally well-known metaheuristics for classic problems.

This is another interesting course you'll use a lot later. The only reason I filed it under **meh** is that most of its content had already come up in earlier modules like Analysis, Algebra, and some of the AI ones, so it felt repetitive most of the time. That said, I think it's kept in the program mainly for the practical side, we got to use GNU Octave¹² (an open-source alternative to MATLAB) and practice and visualize the concepts from the lectures.

One thing to note is that, for this module and **Time Series Analysis and Classification**, the finals and midterms are mostly based on the non-coding exercises you see in the labs. So focusing on those and understanding them well should get you through both just fine.

¹²<https://octave.org/>

In this semester you'll also have **Advanced Databases**, a natural successor to the one you had previously. This one, though, tackles the bigger issues modern DB systems face. It starts from the bottlenecks relational DBs hit when data volumes grow, and the performance optimization techniques used to deal with them. From there it moves to distributed DBs, where data and processing are spread across multiple sites, then beyond the relational model entirely to NoSQL databases and their different paradigms, before finally concluding with data warehouses and how they're used in business intelligence to support analysis and decision-making.

In my time, this and the previous DB module from second year (**Databases**) were taught by the same instructors, so the same remarks I had still apply, mainly around how the finals are put together and how much the lecture slides weigh in. One more thing to note, to get through the labs smoothly you'll need to install a lot of software over the semester, and most of it is pretty heavyweight, so take your time setting it all up properly, ideally on the school's Wi-Fi, so nothing slows you down later on.

One more this semester, **Computer Networks and Security**, a course that looks at information systems security from different angles. You'll start by studying the vulnerabilities and security threats in IT systems, along with some fundamental security services and protection mechanisms. A large part of the module then covers cryptography, the use of mathematical techniques to secure communications in the presence of adversaries. From there it moves to network security, where you'll revisit some of the notions from **Networks and Protocols**, this time from a security angle, before finishing with OS security.

As rich as it looks, you'll get plenty of practice, especially in the TD sessions for the cryptography part. You'll also get hands-on in the lab sessions, doing some packet sniffing with Wireshark and going through well-known attacks like eavesdropping and spoofing. And you'll get to check your week-to-week understanding through the quizzes the instructors share, so you'll have everything you need to ace this module.

During my time, this course and **Operating Systems** also shared the same instructors, so you'll notice the same methodology in the course's structure and organization, specifically when it comes to how the question banks are used and the course reader. I used to find that this kind of approach reduces a module's entropy, so you walk into finals at ease, and you actually retain most of what you learned, even years later. Funnily enough, the disorganized modules are the ones you won't remember much of down the line 🤔

The last module this year is **Project Management**. As you'll probably get used to, you'll always have a similar module each semester. In this one, you'll go over some notions around how projects are organized, executed, and managed. I remember it also involved some calculations around how schedules are built, how resources are allocated to keep the work moving, and how costs are broken down, specifically using PERT and Gantt charts.

I said the previous module was the last one, but that's not completely true, since you'll also have another project module called **Group Project**. In this one, you'll be given a set of projects depending on the teacher in charge of your class, you'll form groups (of 4 or 5 people), and pick the project that interests you the most. You'll have the full semester to work on it, then pitch your solution in a dedicated session right at the end, just like you'll defend your FYP (your final-year project) two years later.

I'm not sure how project ideas are formed right now. I remember we did a brainstorming session where we came up with a few ideas, listed them, then groups started applying to each of those. This is at least better than getting assigned something at random. That said, what you'll work on is still capped by your teacher's expertise. So you might end up in a situation where other classes get better¹³ projects, simply because those happened to fall within their teacher's expertise. This is obviously a kind of selection bias, which is understandable, but it can also affect how the project goes if you end up working on something you're not fond of. Either way, keep this in mind, since you'll almost always do better on a project you're excited about.

That's it for this year's modules. As you can see, it's far from AI-only, you also get security, networks, and mobile dev in the mix, so the program ends up a lot broader than the school's name makes it sound.

As usual, the table below gives you the year's snapshot:

¹³Better here means an idea you'd have liked to work on, not a project that's objectively better.

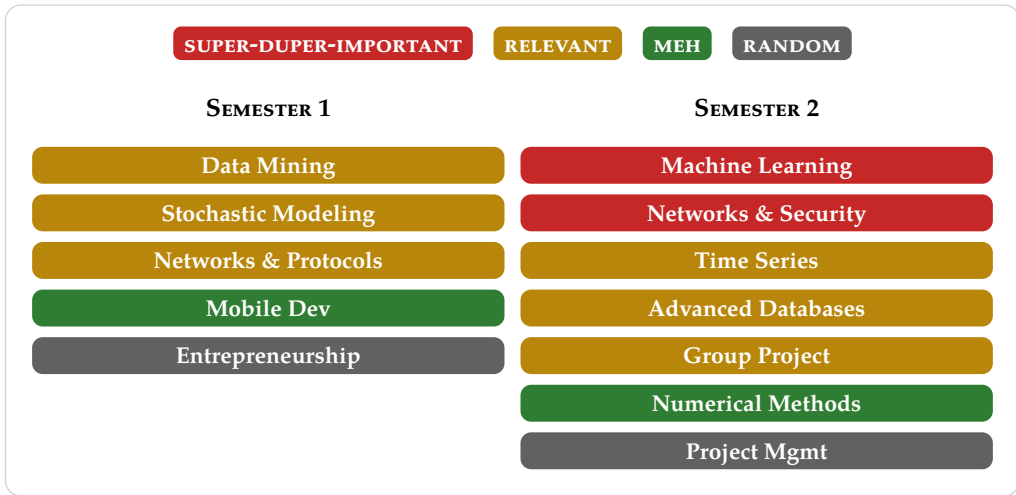


Figure 20: Third year, module by module.

Building Gostu

As I mentioned earlier, as the years go by, you'll find yourself more and more tempted to try new things beyond just studying. By this time, you'll find that you've picked up a set of skills, and that, with a bit of motivation, you can work on getting better at them and putting them to proper use, whether through internships, entrepreneurship, side projects, and the like.

One of those things, for me, was starting a company. And before I get into that story, I owe you the explanation I promised earlier, about why I'm not a fan of the word "startup". For me, it sets the wrong expectation. It carries this whole image of a rocket heading straight for unicorn status, some world-changing thing that's about to blow up overnight. That framing quietly messes with your head, because you catch yourself chasing the size of the vision before you've even checked whether anyone actually wants what you're building. What we were doing was simpler, and more honest, than the word makes it sound. We were building a product, trying to solve a real problem, and running a small company around it. So whenever I say startup in these pages, that's all I really mean, a few people building something and hoping it sticks.

Anyway, back to the story. I was in exactly that position in the summer of second year, when I was approached with the idea of building the first startup at ENSIA, one that opened the way for many others in the years after. That year I co-founded Gostu, with a great team of people. It was an all-in-one app for BAC students, built around a forum where they could discuss things with peers, ask questions, and debate different topics. If you're old enough, you'll remember

something similar we had before, called Mountadayat Djelfa¹⁴. The forum's idea came from there. We just wanted to make it more focused on BAC topics, and more visually appealing and practical.



Figure 21: Gostu logo.

Gostu wasn't only that, though. As part of its all-in-one vision, it also had a courses area, where teachers could open online courses, say Math, and students could subscribe and study online through Zoom¹⁵. We also built a resources section, where students could browse past exams, course readers, and more. All of this was built to tackle one pain point we'd spotted among students, which is distraction. We wanted to give them a single place with everything, so they wouldn't have to keep jumping from one app to another.

We ran it through our third year, since most of the team was in the first promo, and we managed to ship a lot of features and move fast, live on both web and mobile, all before the era of agentic coding tools, so I feel like a coding demon looking back on what we pulled off. However, we stopped one year into the adventure, for various reasons, but mainly missing product-market fit (at least in my opinion), which I'll come to later on.

Starting a business at a young age isn't an obvious choice, so I've tried to put together a few reasons that might motivate the decision:

- **More Time.** At this point in your life, even though you're busy studying and getting on with everything else, you'll find you still have plenty of free time for side projects. Entrepreneurship being just one example, you'll experience investing your time in something you genuinely care about and have skin in. It's also something you can do later in life, but time is priced higher then, and investing it in a project is riskier, which brings us to the next point.
- **No Risk.** At this stage of life, you have fewer dependents and fewer commitments, which makes risk far more affordable now than later on, since failure itself is more manageable and recoverable. I also feel it's more enriching to fail at your own project than at someone else's. When it's your own project,

¹⁴منتديات الجلفة, <https://www.djelfa.info/vb/forumdisplay.php?f=170>

¹⁵<https://www.zoom.com/>

you're exposed to every decision that got made along the way, which makes it easier to backpropagate through the breaking points that kept it from being sustainable, and learn from them. And failing at your own thing sits with you more easily than failing at someone else's. "No risk" is a bit of a misleading label, though, since the time you put in is itself a valuable asset you're spending on the project. Still, I think if you're willing to learn, you have to pay a price for it, and if that price is time or effort, then so be it, as long as it's rewarding!

- **Network.** One thing I used to love about the school is the access to a network, which might seem like a luxury at first, if you're not in the market yet. But once you're live, you'll find yourself needing help from all sorts of parties, whether legal, financial, or even technical. Having access to people working in these domains (and others) is a real necessity for the projects being built at ENSIA. I'd also like to point out how kind the people we reached out to there were, and how willing to help. I learned that sharing as much information and experience as you can with others is the only way we move this community forward, and that's probably the reason I'm writing this book.

That said, there are plenty of things to watch for before and after you jump in, and I'll try to list some I drew from my own experience:

- **A Team.** The people you'll be working with on this journey are the ones you're locked in with for most of your time, so you want to be as careful about who you pick as you'd be about the problem you'll be tackling. Being just a friend or a nice person won't be enough here, and being good at what you do isn't the strongest criterion either. Ideally you're looking for motivation, paired with technical ability, and someone who stays a good person in the worst moments, not just the good ones. I'll take this chance to thank the people I worked with at Gostu. I genuinely enjoyed collaborating with every single one of them during my time there.
- **Product Market Fit.** PMF is the degree to which the product you're building satisfies the demand of the target market, and it's the one you'll be working to meet for much of your early days, since it's the only feedback mechanism that tells you whether the problem you solve is painful enough for clients to pay for, and how much they're willing to pay to start using your solution. Until you reach that state, you shouldn't be focusing on anything else, because as long as your solution isn't fitting anywhere, there's little point talking about how big your vision is or how good you are at what you do! This, in my opinion, is why I dropped Gostu a year into building it, since we had no clear

signs of reaching PMF, and it was clear we'd either tried to solve a nonexistent problem, or hadn't gone about it the right way.

One other thing to note is that, at this stage of your life, you're less likely to have FMF (Founder Market Fit), which is how well your skills and background as a founder align with the market you're trying to serve. So most of the time you'll know little about the industry you're aiming to sell to, and in that case, you need to spend most of your time researching potential clients and gathering as much intel as possible, to make your solution problem-fit and to have strong arguments in your next pitches.

- **Time.** As both a builder and a student, your time is very tight, so within the free time you do have, your moves need to be as calculated as possible. You might spend the next month or two building some big feature, only to be faced with the harsh reality that nobody asked for it. You'll also have stints where you won't be working on your project at all, like exam periods. These are dark periods that eat into your project time without adding much value, so plan around them. I'd also point out that agentic coding tools may have sped up the feedback loop, since coding is less and less the bottleneck these days, but you have to use your budget wisely too 💰
- **Flexibility.** This comes as a direct consequence of poor FMF. The initial idea you start with often isn't the one you end up building and pitching to customers and investors. As you put more time into the project, understanding the market and trying to make money from it, that idea tends to shift. And part of the game is accepting those changes and adapting to them, as long as you have strong signals from your client base that the change is the right one.
- **More Input, More Feedback.** The nature of building your own project might have you spending a long time heads-down, building before going public, and that's generally good for focus. But don't limit yourself to the feedback from your team, or the corridors of the school. Try to step outside that and seek feedback from external parties too, ideally straight from your end clients. Getting lots of input, from all kinds of people, teaches you to look at your product from different angles, and catch perspectives you wouldn't have found elsewhere. And don't be afraid to expose your idea even in the early phases, out of fear of losing it or having it stolen. I believe the more people you expose to trying to break your claims, the stronger those claims become. Otherwise you'll face that stress-testing later, when you've already sunk a lot of time into it.

- **Less Talk.** If you're going to market a product, make sure the few people you convert through your marketing channels in the early days meet a product that works! So spend less time marketing and more time making sure what you're marketing actually works and is properly usable, UX-wise, even if that means pushing deadlines back to get the product working, so that you don't end up trading people's attention for loose timelines.

One other note: make sure to attend events that actually matter to the market you're serving, and don't put too much weight on the random events usually organized at the pole just to show that students are being entrepreneurial, since those are generally political showcases, nothing more. It's fine to have them, but you'll want to spend your time in places where you can spot weaknesses and find areas to improve in your solution.

- **Stress.** One last thing. In your day-to-day life, you'll be juggling a lot, trying to study and pass your year while working on your project and meeting deadlines, so it can feel very stressful at times. Even so, you shouldn't project that onto your team or teammates. Make sure to protect them before anything else, since the whole point of this experience is to enjoy scaling your product, not to create beef or stressful working environments.

These were a few key takeaways I had in mind, from that experience and from watching other students go down the same path in the years after. I can't stress enough that I could be wrong or biased on some of the personal-experience points here, so learn when to trust your gut on those, because you generally won't find a single recipe for scaling a business, and you'll necessarily be doing things differently to make your product work.

Chapter 5: Fourth Year



Figure 22: The dome, a familiar sight by now.

Context

Now, coming into fourth year, for those of you who won't go on to a master's degree, this is technically your last year at ENSIA. And that feeling from last year, of how fast life moves, has fully sunk in by now. Looking back on everything that's happened, you realize that no matter how hard it all seemed, you got through it and moved on to whatever came next. I find this makes the future easier to face, since it's fulfilling to know that you can get through anything, or take it on, if you approach it properly and with the right state of mind. But before talking about any future, you need to know what that future looks like. You can't head toward a destination you haven't picked. That's what I found myself mostly battling with, back then and honestly right now too! And I have the feeling you never really finish understanding yourself, and trying to take the right step given what you know, and come to realize, about yourself. Still, I'll try, later in the book, to document the kinds of decisions you can make here, and share my own experience with them.

As we'll see in a bit, this year is really packed with challenging yet interesting modules. You'll get to study AI in more depth, and go through some of its well-known subdomains like NLP (Natural Language Processing), CV (Computer Vision), and more...

By the end of this year as well, you'll find yourself searching for the right binôme (your project partner) and FYP for next year, and I'll try to dedicate a big part of this year's Miscellaneous section to that, and to the different options you can pick from as a student at this school.

Semester 1

During this semester, we started studying in October, as opposed to September like in past years. This was mainly because of the internship students are asked to do during that time. All in all, you'll be asked to find a place where you can do what we call a discovery internship, one that should relate to AI, data science, software engineering, or any domain from your curriculum. However, it won't be one where you're asked to produce something or solve some kind of problem. Its primary purpose is to put you in a professional setting where you can practice what you've learned.

In case you're wondering how you'd land that internship, and where, the answer is that it mostly depends on you, since you can go look for a company that interests you and is willing to take you on for a one-month internship. It can happen either over the summer or in September, so you get a bit of freedom there. If you can't find a host company, your fallback is what the school offers, since it periodically uses the companies it has agreements with to host students during September. You can find a list of those on the school's website¹⁶.

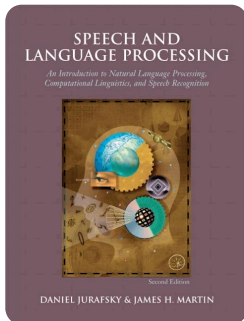
By the end of this internship, you'll be asked to prepare a report on your experience there, what you were involved in, and what came out of your work, with your supervisor at the company signing it to make it official. As mentioned earlier, you won't be asked to build something big, since this is mostly a discovery internship. However, you'll want to get the most out of this short experience, so try to join a company where you'll do real work, without being overworked in the process.

You'll learn, from what you will see, that there are plenty of companies where you'll end up going in for nothing, since the workers themselves have nothing to do. So you'll want to avoid those. At the same time, you might land somewhere with plenty of work and planning going on, so try to be as curious and proactive

¹⁶<https://www.ensia.edu.dz/cooperation/>

as you can, and build something in that environment that ends up used by your peers, or even beyond the school. There are plenty of examples of that around ENSIA.

Starting with the courses you'll have this semester, we begin with **Natural Language Processing (NLP)**, which tackles the treatment of language in AI as a whole. It starts by laying down some fundamental NLP concepts, before moving to logical and statistical algorithms for processing linguistic information. It then ramps up to more advanced concepts, using ML and DL for NLP tasks like the well-known text generation, among others.



This module is based on the book *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*¹⁷ by Jurafsky and Martin. It's a well-known reference in the NLP field, packed with exercises and algorithm implementations on top of the theory.

You can also google its name and find plenty of repos where people solve the exercises or publish code you can build on later.

You'll also have lab work, where you reproduce the concepts you cover in the lectures. My favorite part was redoing the well-known NLP models, from RNNs and LSTMs to the attention mechanism and transformers. You'll also get the full semester to work on an instructor-suggested project, one that's pretty challenging. For instance, in my year's batch of projects, a lot of them dealt with the practical use of LLMs for teaching vocabulary, reading comprehension, and Algerian history, or for task detection and classification, and were mostly focused on Arabic, so you might find yourself doing a lot of fine-tuning to push past state-of-the-art (SOTA) results. These topics were super trendy at the time, so you'll have plenty of fun this semester 😊

We also had **Deep Learning** this semester, a course on yet another hot topic in the AI world lately, and one you'll approach differently from the online courses you've probably seen. You'll start with understanding data, specifically in very high dimensions, and how to handle it with techniques you already know like PCA, before moving on to learning from that data, starting with logistic

¹⁷<https://web.stanford.edu/~jurafsky/slp3/>

regression, laying the ground for the error-reduction principle, then moving to neural networks, backpropagation, CNNs, and sequential models like RNNs.

You'll find this course overlaps with a lot of what you studied before, mainly **Machine Learning** and **Data Mining**, and I personally felt that either its content or the other courses' could have been adjusted to make this one feel more distinct. I also noticed the labs are the same as the ones from the well-known Deep Learning specialization on Coursera¹⁸, which means you can use that same resource, mainly to revise through their quizzes.

The next course is **Introduction to Mobile Robotics**, which, despite the name, is really a course in control theory. In it, you'll study the theory and practice of controlling dynamical systems. It starts from system modeling using differential equations and transfer functions. From there, it introduces classical controllers such as PID (Proportional-Integral-Derivative), followed by state feedback and optimal control using LQR (Linear-Quadratic Regulator). It also covers non-linear systems, where controllers are designed for non-linear dynamics, before finally getting into intelligent control through fuzzy logic.

You'll have both TDs to solve practical exercises and lab sessions where you'll try to develop some robotics applications, relying on the concepts from the lectures. You'll primarily use ROS (Robot Operating System)¹⁹.

You'll find this module very different from anything you've seen in school before, since you'll be grappling with physics concepts that ENSIA doesn't cover in earlier years, which makes it slightly harder to digest. I also noticed the course's pace doesn't help, since it was fast-paced and gave little time to properly understand the content. The reason is that it was supposed to be two courses instead of one, and ended up merged and sped up, which is something that should be reviewed, in my opinion.

¹⁸<https://www.coursera.org/specializations/deep-learning>

¹⁹<https://www.ros.org/>

I also found this course to be inspired by Stanford’s CS223A²⁰, for which the full course is available online for you to follow along, with plenty of assignments to practice from. It’s also presented by one of the pillars of this domain, Oussama Khatib²¹, which makes it quite interesting to follow!



You’ll also take **Wireless Communication Networks and Systems**, which focuses on how information can be transmitted efficiently and reliably through real communication channels, giving you the physical foundations of wireless communication systems. It introduces the principles of digital and wireless communication, covering signal representation and analysis, Fourier transforms, bandwidth and data-rate limitations, then channel impairments and practical transmission techniques such as encoding, synchronization, and error control.

You can effectively see it as a continuation of what you studied before in networks and the like. Unfortunately, though, in my time we studied it in bursts rather than week by week. The instructor was often out of town, so sessions got batched into whatever days he was around. There was one week where every free slot in our timetable got patched with this module, and then long stretches where we barely touched it at all. That’s what made it messy to revise and retain, because you’d take in a big chunk of it in a few days, then leave it alone for weeks. If it still runs that way for you, the answer is to keep your own rhythm between the bursts instead of following the timetable’s. Luckily, I can point you to some resources with plenty of problem sheets to practice from, which the course instructor shared with us. One is from Northeastern University’s Khoury College of Computer Sciences²², another from Carnegie Mellon University’s School of Computer Science²³, and one more from Bilkent University in Turkey²⁴.

²⁰<https://see.stanford.edu/Course/CS223A>

²¹<https://khatib.stanford.edu/>

²²<https://www.khoury.northeastern.edu/home/rraj/Courses/G250/F07/ProblemSets/>

²³<https://www.cs.cmu.edu/~prs/wirelessS24/>

²⁴<https://www.cs.bilkent.edu.tr/~korpe/courses/cs515-fall2002/>

Next on the list, we have **Human Computer Interaction**, which covers how people interact with different types of computer systems. It focuses specifically on designing interfaces that are intuitive and easy to use for your end users. Through it, you'll work on a team project, where you design any kind of application (web, mobile, and so on) using the design principles you've studied, and apply some user-testing concepts as well. Later, in the finals, you'll have an exam split into two parts: a theoretical one, where you answer questions on the lectures, and a practical one, where you get more hands-on tasks, like looking at designs and spotting strengths and weaknesses using the principles you studied, and how to approach user-testing in practical situations, all things you'll have gone over in the labs.

I found this module very informative, regardless of the little importance you'll give it, since it's not very technical or mathy. However, it only ended up in **random** because I think modules like this shouldn't follow the strict lecture-and-lab system, and could instead break that routine with something more creative that makes the most of their hands-on nature.

Last in this semester, we have **AI and Ethics**, which debates the role of ethics through its lectures, and explores the various ethical issues and challenges facing the people and institutions involved in AI development. If you're into exploring the legal and regulatory frameworks around such technology, you'll find this one interesting.

Semester 2

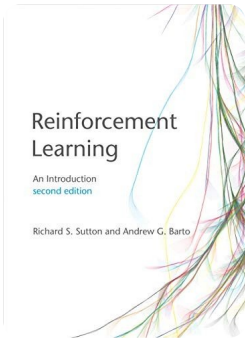
To kick off this semester, we have **Computer Vision**, which aims to teach you the techniques used to make computers extract information from the visual world, and use that information for more advanced tasks. It's quite foundational in its approach, since it starts with how cameras form and sense images, then moves through image processing, feature detection, geometric vision, 3D reconstruction, motion, and visual perception, before introducing machine learning and neural networks for vision.

It'll be accompanied by a lab session, where you'll dig into the concepts taught in lectures using Python, which you should be familiar with by this point, given the journey you've had with it so far. One thing I noticed, whether about this course or others at this level, is that instructors expect you to be actively using agentic coding tools. You can tell from how fast they move from one section to the next in the lab, and if you're not using any, you might find

that annoying. I also hope this isn't happening at lower levels, which need more time with code, as I mentioned earlier in this book.

This course also draws heavily on Columbia's First Principles of Computer Vision²⁵, which is also available on Coursera²⁶. So you can always make use of GitHub repositories that post walkthroughs and quizzes to practice from, like this one²⁷.

Now for a very spicy one, **Reinforcement Learning**, which tackles teaching an agent to perform actions through its interaction with its environment. It'll cover Markov decision processes, value functions, and dynamic programming, before moving to learning directly from experience through Monte Carlo and temporal-difference methods such as SARSA and Q-learning. It also covers the case of large state spaces through function approximation and policy-gradient methods.



Browsing similar courses online, you'll quickly realize this course adopts the same methodology proposed in the famous *Reinforcement Learning: An Introduction*²⁸ by Richard S. Sutton and Andrew G. Barto, which has since been picked up by numerous courses, such as Coursera's²⁹ and David Silver's at UCL³⁰.

During my time, the course presented at ENSIA was also heavily inspired by the one on Coursera, since it reused its labs, quizzes, and assignments for the midterms and finals... So, unless the adoption strategy changes in the coming years, you might find repositories like this one³¹ helpful when revising .

Moving to **Speech Processing**, which was more about signal processing than about speech itself, since we studied how signals are represented, analyzed, transformed, sampled, and processed to extract information. It builds up through discrete and continuous-time signals, and also tackles Fourier analysis

²⁵<https://fpcv.cs.columbia.edu/>

²⁶<https://www.coursera.org/specializations/firstprinciplesofcomputervision>

²⁷https://github.com/ansariparvej/First_Principles_of_Computer_Vision_Specialization_Coursera

²⁸<http://incompleteideas.net/book/RLbook2020.pdf>

²⁹<https://www.coursera.org/specializations/reinforcement-learning>

³⁰<https://davidstarsilver.wordpress.com/teaching/>

³¹<https://github.com/CharlesYuan02/coursera-reinforcement-learning>

and the Laplace transform. At the very end, it touches on some applications of signals, speech in particular, but it wasn't really focused on that.

Not much to say here, since it was mostly a practical course, so solve as many exercises as you can and you'll be fine in the assignments. You'll also have a section of lecture-based questions, which carried a big part of the grade in the exams, so don't skip revising the early lectures, which cover the process of speech production, on top of the usual exercises, which you'll solve plenty of in your lab sessions.

Coming next is **Big Data Analytics & Visualization**, a very interesting course where you'll learn how to handle massive amounts of data through advanced techniques and technologies. You'll learn about the MapReduce paradigm, and how to dispatch big jobs across your workers to get them done in a shorter amount of time, which you apply extensively in Hadoop . You'll also spend some time learning about HDFS (Hadoop Distributed File System) and its architecture, and getting into YARN (Yet Another Resource Negotiator), the core resource-management and scheduling layer in Hadoop.

The course also covers data analysis, where you'll learn about Apache Pig , a scripting layer for running your MapReduce jobs, since no one wants to do that in plain Java. It makes your life much easier when querying insights from your data.

At this point, you'll start to notice that the difference between this module and **Advanced Databases** from last year is that this one is more technical, and covers some of the standard practices and technologies actively used for big data. For instance, you'll revisit the data-warehouse concept you met earlier, and see how to apply it using Apache Hive .

This course had both lectures and labs, but the labs were far more interesting and challenging, since we had some good examples to practice on, and used Java for most of it. You'll learn how to set up your own Hadoop cluster, but you'll also be given access to the Hadoop cluster on the school server, which you SSH into to work on an example dataset throughout all the labs.

For the finals, though, I don't remember them being very focused on the technical side, since I don't recall us being asked to write any code, but they were fairly challenging on the theoretical parts, and we were properly examined on that side of the lectures.

One other module you'll have this semester is **Enterprise Computing**, a very interesting one that walks you through the standard best practices for creating an enterprise-level solution from A to Z. You can think of it as a natural continuation of **Software Engineering**, since you'll start from the software architectures you studied there, before focusing on microservice architectures (MSA), and how to link services together through REST, SOAP, message queues, and so on. You'll also cover authentication and security in enterprise software, and how to properly QA such systems through different techniques. At the very end, you'll study DevOps, and how to set up continuous integration and deployment (CI/CD) pipelines, and how to set up monitoring and logging to gain observability into your system and ensure performance at scale.

This module is also heavily based on Java, since it starts from the assumption that enterprise software is mostly built on it, so you'll find it a bit challenging, especially during labs where you'll be asked to use it for everything. But overall, you'll find it a good refresher in reading and writing actually good code, regardless of the beans ☕ you'll end up tangling with 😊.

We were also asked, throughout this course, to build a DMS (Data Management System) as part of the course project, which was a good experience, drafting an SRS (Software Requirements Specification) for such a complicated system, and building it feature by feature, using strictly the notions from the lectures.

As for midterms and exams, we were given plenty of MCQs to test the notions from the slides, with the occasional Java code snippet to assess that side of the course too. A good way to prepare, apart from reading the slides, is to generate MCQs from them using ChatGPT or any other tool, making sure the questions are genuinely spicy and challenging, then trying to solve them. You'll end up using this trick a lot, in any ENSIA course that has an MCQ section.

Another one is **High Performance Computing**, in which you'll try to understand how computationally demanding problems can be solved efficiently through parallel hardware. It starts from the fundamentals of parallel computers, before progressing to shared-memory and distributed-memory paradigms, and parallel programming with OpenMP³², with a big focus on the principles of communication and workload distribution. It also covers Ray³³ for higher-level distributed computing and data-intensive computations.

³²<https://www.openmp.org/>

³³<https://docs.ray.io/en/latest/ray-overview/index.html>



Figure 23: The Jaguar supercomputer at Oak Ridge National Laboratory, one of the machines that made high-performance computing what it is today.

This is another relevant module in your AI foundation, since you might end up optimizing large computational systems to run in shorter time. So think of this course as your starting point for tackling such problems, and maybe even specializing in them in the future, since it's a fairly hot topic in research. As for the finals, they came in an MCQ format just like the previous module, so make sure to understand the slide content well to ace them.

One other module we had this year is **Selected Topics in AI**, which is a rundown of some topics where AI can be used to solve problems. The first course in it was **Blockchain**, where we covered the basics of blockchain and its components, cryptocurrency, consensus mechanisms, and smart contracts. You'll also get to practice building a basic blockchain and Merkle trees, doing simple transactions, all in Python.

The second course was **AI for Earth Observation**³⁴, where we studied remote sensing, image processing and feature engineering for it, how to handle scarce data, and how to fuse data from different sources in the context of GeoAI.

All in all, many would argue this module should be considered random, but I can see the value in having such courses, to open you up to different perspectives on applying AI. That's really important, especially ahead of your final-year project, where you'll probably apply AI to a new domain, different from the ones you've already come across.

As for their exams, expect an MCQ here too. You'll be an expert at handling those by the end of the year.

Last one for this year is **Academic Communication and Research**, which tackles the academic practices behind writing articles, reports, dissertations, and the like.

³⁴This is a name I came up with, since as far as I know there's no formal name for it.

And with that, you're done with modules for good, unless you're doing the master's, in which case a handful of lighter ones are waiting for you next year. A lot of what you picked up this year will come in handy for your FYP, so don't rush to forget it all the moment the finals are over 😊

Miscellaneous

By this level, you'll be well tuned to face such courses and find your way through them. That won't eliminate the stress, in my opinion, but it does give you a lot more confidence, year after year.

Overall, the year is packed and diverse, and the curriculum is a satisfying one, even if it still has a few gaps here and there. The main one, in my view, is that it could be kept more up to date, and refreshed every so often with newer topics. To put things in perspective, I'm no curriculum expert, but in practice you come across many concepts and catch yourself thinking it would've been cool to cover this or that in one of the courses, if only to give the program more credibility in the AI space. For instance, I'd like to see agentic systems covered more and more, with concepts like MCP (Model Context Protocol) taught in one of the courses. I'd also like voice modality covered properly somewhere, since it was given very little weight, in my opinion, among other things.

As usual, here's the board summarizing both semesters:

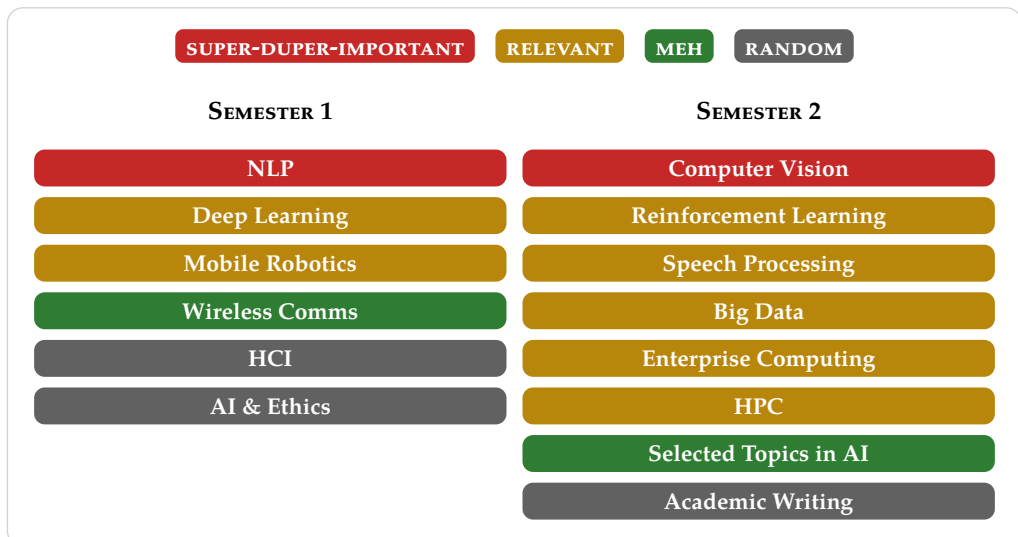


Figure 24: Fourth year.

Finding Your FYP (and Your Binôme)

As I said earlier, I'll try to give a walkthrough of the FYP at ENSIA, and the options you can pick from once you're at this stage.

One thing to note first is that ENSIA doesn't approach the FYP logistics any differently from other schools or universities, as far as I know, so there are the usual common regulations to follow. The school invites students to work on:

- A fully internal project, where a supervisor proposes a theme and then picks a binôme to work on it, following their own selection criteria.
- An external project with a company, university, or research lab based in Algeria. Here, you'd need a supervisor from that external party to suggest the theme, plus a co-supervisor from the school on the team, to supervise the work scientifically and represent the school's position throughout the project.
- An external project with a university or research lab based abroad. This works similarly to the previous one: you'd need a supervisor from that party to suggest the theme, and an internal co-supervisor to follow the project's dynamics.
- Your own startup, under ministerial decree 1275, where you build your own company and defend it as your FYP.

As for an external project with a company based abroad, that one won't be possible, since, according to the school's administration, it isn't allowed under Algerian law. So the external party has to be a research-oriented one (a university or lab) for you to pursue it.

When it comes to how projects are structured and supervised, the school has a website where FYPs are posted, by either a supervisor or a co-supervisor, with all the project details and so on. Fifth-year students have access to those and can express interest in the ones that appeal to them. If you're in a binôme, only one of you needs to express interest. Right after that, you'd be contacted for an interview or an exchange session with the people responsible for the project, before finally settling on the one you'll work on next year.

Timing-wise, people generally start looking from the end of fourth year through to the start of the next, since the administration sets a September/October deadline to pick your project and get started. So you can use that long window to find the right project setup for you.

You're also not actually required to work in a binôme, you can go solo with no big administrative issues. That said, the school encourages working in a binôme, mainly because of staff shortages in covering all the projects, whether for supervision or for the project defenses later next year. This might be a good starting point for your FYP search, working out who you'd want to team up with. The main argument for picking someone is splitting the workload and being able to move faster, as well as bringing an extra perspective that makes the work firmer.

On the flip side, finding the right person can be tricky, since, just like with the startup team I talked about in third year, being friends isn't enough, and neither is being the most proficient. So for the best project dynamics, it's better to pick someone who complements you, someone able to see things differently and give the project a fresh look, and a restart whenever needed, with consistency and focus. Similarly, whatever you expect from your partner, you should expect from yourself too, so make sure to hold yourself to the same standard. Overall, if you end up with a better choice than mine³⁵, you'd be good to go³⁶.

The same goes for your supervisor(s), maybe even more so. Do a bit of background-checking before you commit, ask around about how they are to work with and whether they're actually responsive, since you'll be tied to them for the whole year. The right one keeps the project's direction and expected outcomes clear, instead of leaving them so vague you end up lost.

Now, on how to land one. If you have an idea you want to develop as your FYP, you can reach out to any supervisor directly. Most opportunities come the other way around, though, where you'll see what each supervisor is suggesting, either internally through the school's channels, like the internships website, or externally through Algerian company offers, or even academic institutions based abroad.

For the ones abroad, you might find something on the FYP website. If nothing comes out of that, you'd need to actively reach out to contacts there yourself, and spam, honestly just spam, a ton of emails or LinkedIn messages, with different proposals tailored to each lab's needs, until one of them agrees to collaborate. Just make sure what you're applying to matches your research areas and interests, otherwise you risk ending up on a project that doesn't interest you. A good move is to also focus on the labs known to accept Algerian students from similar schools, or ones hosting Algerian PhD students and professors (by tribalism, lol), so you increase your chances of getting selected.

Once one of them says yes, make sure you get your full rights as an intern during your FYP, since you'd want either to be compensated in money for your work (the same goes whether you land something locally or abroad), or to have them host you there for a few months, putting that compensation toward accommodation. It'd be a good experience to see how other labs operate, and you might come away inspired to pursue a PhD, or discover other branches you'd be interested in once you're done with your FYP.

³⁵My binôme, [Abderrahim](#).

³⁶Just kidding, man... you know how much I enjoyed working with you 🥰

One other thing to watch for is the country hosting your FYP, which can create administrative issues when signing your agreements, since some countries might be under boycott due to sensitive political issues at the time. So make sure to get out of the bubble and watch the news to avoid unnecessary headaches!

I won't lie, this part can get stressful, since you're trying to balance an equation with way too many variables, the binôme, the project, the supervisors, and so on. But you'll learn a lot from it, especially about going after the opportunities you want, and that's something you'll keep doing long after your FYP.

Chapter 6: Fifth Year



Figure 25: ENSIA promo, 2021 to 2026.³⁷

As I said in the previous chapter, for those of you who won't enroll in a master's degree, last year was your last year of modules, and you won't have to pass the ones required for a master's. And as you'd expect, you'll have the full year dedicated to your FYP, which is one semester more than what other institutions typically allocate. So you'll have more time to pour into it.

On the other hand, if you opted for a master's, you'll have some courses to take during the first semester. That said, you won't have to deal with any deeply technical module, apart from one, and even that isn't very technical. The rest are oriented toward business, marketing, law, and academia, and you'll only be in for two days a week. By the end of the year, you won't be asked to prepare another dissertation for this master's, but to put together a poster for a poster session held later in the year, which counts as your pass for the master's degree. Weird, right? But that's how it went in our year, and it partly makes sense, since this master's isn't the main degree, it's an add-on to the state engineering degree you'll get after you defend.

³⁷Photo by [Yasmina Medjri](#).

Given how different this year is, we'll structure this chapter a little differently. We'll start with a walkthrough of the master's modules, before moving to the FYP section, sharing what the process looks like through the year, up to the defense and graduation at the end. I'll also share the opportunities you might pursue once the year wraps up, both through the school and externally.

Semester 1

During this time, you'll have a total of five modules to cover. Starting with the most technical one, **Advanced Research Topics**, which walks you through a few prominent research topics, three of them in our case. You'll start with the use of AI in biomedicine, accelerating drug discovery and analyzing biological data to improve clinical diagnosis and treatment. The second topic covers multi-agent AI systems, which is about how multiple AI agents interact and coordinate, whether cooperating or competing, to solve problems together. The last one, which got an intensive week of its own, is Deceptive AI, which looks at how AI systems can learn to deceive, and the safety and trust concerns that come with it.

You'll find this relevant if you plan to research in one of these areas, especially since you'll have a team project, where you try to produce a piece of scientific work and aim to publish it in one of the suggested journals, on one of the first two topics only. That's good training for the dissertation and paper you'll end up writing later in the year, but I find it pretty ambitious to aim for a publication at that level, given the short time you're given and how new the assigned topics are to you.

Then comes **Marketing Research**, which covers marketing in general, its process, the study of micro and macro environments, industry analysis (SWOT, Porter's five forces, and so on), before moving to consumer behavior analysis, and finishing with marketing strategy evaluation (segmentation, targeting, and positioning). And looking back on it, it was honestly well presented, with a solid practical methodology and strong real-life insights.

The exam for this course came in two parts, an MCQ with theoretical questions from the lectures, and a practical section where we had to solve a case study, similar to what we did in the TD sessions.

We also had **Business Law**, which introduces you to the legal jargon around companies, their creation, what legal status to pick, and how to keep them compliant. As practice, you'll be asked to set up a company as a project, walking

through all the stages you studied and arguing for each choice you make, legally speaking. That ended up being the exam in our year.

There's also **Introduction to Research Methodology**, which picks up from the Academic Communication and Research module you had last year, where you'll learn about research methodologies, their stages, and the objectives you aim for at the end of the research process.

And the last one is **Algerian History, National Identity, and Citizenship**, which is, well, another module you'll end up taking .

Final Year Project

The biggest change you'll feel at this level is probably the detachment from school. You go from having to attend lectures, TDs, TPs... to nothing, you're not even asked to be on campus anymore. You'll fill the void with FYP work, other activities or side projects if you're into any, and settle into this new rhythm.

You can always come to school and work there if you want to keep that lifestyle, you still have access to the open spaces and the library, and you'll even get exclusive access to some rooms dedicated to FYP students only, where you can work, hold meetings, or whatever else.

Depending on the work mode your project requires, you might be asked to meet your supervisors in person, or work in a hybrid mode, clocking some in-person hours at the place you're interning at. You might also find yourself in a fully remote setting, where you'd have the freedom to work anytime, anywhere.

Either way, you'll need to hold frequent meetings with your supervisor(s), weekly, twice a month, or more or less often, depending on everyone's availability and schedule. Make sure not to miss those, and to communicate your progress as clearly as you can, what they should expect from you next time, and what you should expect from them. And whenever you're stuck, don't hesitate to ask for help. The FYP is basically a year of hard problems, so you will get lost at some point, the trick is to work your way out using whatever you have at hand, your supervisor(s) included.

On that note, I'd like to thank my supervisors, Dr. Hakim Ghazzai³⁸ and Dr. Nouredine Lasla³⁹, who made sure we were never blocked at any level, that we worked smoothly toward what we'd planned, and made the whole experience genuinely fulfilling. I can only be grateful to them.

³⁸<https://cemse.kaust.edu.sa/profiles/hakim-ghazzai>

³⁹<https://www.ensia.edu.dz/biography/nouredine.lasla/>

One skill you'll find particularly useful this year is anticipation, learning to work out what you'll need ahead of time, and when you'll need it. It could be data, maybe compute access, anything really, that won't be available to you until later in the year, so your planning ends up following that availability. You'll find yourself making the most of what you do have, and of the time you have with it.

As for the timeline, you'll have a checkpoint in the middle of the year, around January in our case, where you're assigned to a jury and given a session to present your progress in about 10 minutes. While it's good to show in detail what you've been working on, there's no need to over-stress, since the session isn't about judging your pace, but more about making sure you have no administrative issues and that you're actually working on something. It's also a good opportunity to report any issues you're facing with your binôme or supervisor, and to raise any problems in general.

Fast forward a few months, and you'll be scheduled to defend your work, in our case, around early-to-mid June. You'll be informed of the date and the jury, which is made up of three members with different roles.

The first is the jury president, who takes the lead on defense day, representing the jury, asking about your work and your approach, and later assigning and announcing your mark after the jury's discussion. The second is the examiner, responsible for reading your dissertation from A to Z and spotting any issues, whether in how the ideas are formulated or in the content itself. His job is to make sure your manuscript is free of errors, and he'll do a post-defense revision to check you've addressed his comments before finally approving your work. The last is your ENSIA supervisor, who advocates for your team within the committee, making sure you get the recognition you deserve, testifying to the importance of the work you delivered, and making sure it's well understood and its outputs are firmly grasped by the audience.

Now for a few logistics. You'll be asked to bring two family members, though you can bring more, since in our year it wasn't a hard rule. You'll also be asked not to bring any food, but that one ended up loosely enforced, applied some times and not others, all depending on which of the school's door guys you end up facing 🤔. You can also invite teachers or your external supervisors to come share the moment with you, and even someone to film the session if you want.

On the day itself, you'll prepare a presentation to give in the first 20 minutes of the defense session. Then you'll give a five-minute business pitch, an extra step where you wrap your solution or research findings as a product and try to sell it. Not every FYP is a natural fit for a pitch, but you do it anyway, since

it's part of the defense ritual at ENSIA. Right after, there's another five minutes for the poster presentation, where you present your whole work at a glance, the problem, the methodology, and how your solution works. And that wraps up your 30-minute oral presentation.

After that, the audience is asked to leave, and you'll be left tête-à-tête with the jury. You'll start with a code demo, presenting your software if you have one. If you don't, and your work is mostly theoretical, you get to choose what to present, I'm not totally sure here, but you could run simulations of your work, or anything else that helps the jury visualize your output. Depending on the jury, you also get to choose between a live walkthrough or playing a recorded video of your software, you never know what can happen when you demo live⁴⁰.

Then comes the Q&A session, where each jury member gets 15 minutes to ask questions, request more explanation, or point out gaps in your work. The comments can be harsh at times, and you have every right to respond within your time limit, but don't argue too much with them. If you end up debating someone who doesn't have the deep view of your work that you do, in just 15 minutes you'll most likely end up talking past each other. So know the limits of the discussion, and when to stop, for the sake of your mark!

You'll then be asked to step out and leave the jury to their discussion, where they settle on your final mark, about 10 minutes, before you're called back in, with the whole audience, to hear your mark, and that you're finally engineers!

It's crazy how five years can be summed up in one moment, huh? Despite the little advances you make each day, each month, each year of those five, you only realize you've finally crossed the bridge at that announcement moment, that you're officially off to start a new phase. So make sure to enjoy it. I hope you all live it with a deep sense of your achievement, and know it won't be the last one, there's more to come!

After the defense, you'll need to address the examiner's reviews, then, once you have a revised version of your dissertation, you submit it again. The examiner then checks you've made all the fixes, before notifying the jury president, who gets your work administratively approved, and you're cleared to receive your diploma. Speaking of which, you'll receive your diploma for the state engineer title by the end of the year, and a second one for the master's if you took that track, along with your final-year transcript, concluding a five-year run.

⁴⁰<https://www.urbandictionary.com/define.php?term=demo+effect>

In our batch, we also took part in a graduation ceremony, where we were asked to invite two family members, and to celebrate the first promos of both ENSIA and ENSM. It's also where this chapter's opening photo ([Figure 25](#)) was taken. That said, I do want to mention that, if this is organized again in future years, I hope it's run better, with the graduating students kept at the center of it, not empty speeches, ill-placed award ceremonies, and pointless political showcases. Many of the graduates had family members travel in from different places, on a working day, only to never see their graduate stand on stage and be properly recognized for the work they did. I find that unacceptable. I hope it was just an oversight, and that it gets addressed in future editions.

Leaving ENSIA

NOTE

As of the 2026 BAC intake, incoming students are asked to sign a commitment to five years of civil service after graduating, working at a state-owned company or, at the very least, staying and working in Algeria. It doesn't apply to my promo, or to those that joined before this, but if you're joining from the 2026 BAC onward, it reshapes a lot of what follows. I'll talk more about this in the afterword.

Leaving this phase brings back the same questions you had right before entering it, questions about choice, and which option actually fits who you are. Back in Fourth Year I said I'd share my own experience with this decision. I'm going to half-break that, since I'd need to figure out my own before offering anything to others. Still, I'll try to give some visibility into the choices you can make. Before that, though, I want to be explicit that I'm not aware of every option, there are practically endless ones out in the real world. So make sure to invest enough time to know yourself first, and adapt accordingly, putting your time into the best path to reach what you're aiming for.

Before I get into the options, there's one thing I want to be upfront about. I'm a firm believer in the concept of Tawfeeq (التوفيق)⁴¹, that we, as humans, can only manage the set of variables within our reach, while others stay beyond us, things we can't directly see or change. A big part of that has to do with our future, and how we sometimes look at where we are and wonder how we even got here. That sense of guidance through the good moments and the bad is what Tawfeeq is about, and what carries a person to the places they're reaching for.

⁴¹<https://orlandparkprayercenter.org/the-essence-of-success-what-is-tawfeeq/>

That doesn't undermine your hard skills and overall ability, but I like to think it's combining the two that gets you the furthest, since that's what ends up shaping the options you get, and the choice you make between them.

If you're into spending more time in academia, the first option that pops up is doing a PhD, where you'll work on a problem for around three years, before finally defending it to earn the title. ENSIA offers a PhD program you can get into through a national exam, which in our year was held around July 2026, open to ENSIA and external students alike, with the next batch picked by rank. The 2026 round had around 50 seats, most of them given to ENSIA graduates as a priority. ENSM and the cybersecurity school are options too, as they also have a PhD program, I'm not sure how many seats they have, but along with the other schools at the pole, they're worth considering if you want a PhD in a slightly different domain and setting than the one you've been in. And if none of those work out, you can always find other PhD programs nationally, at other universities and the like.

Going for a PhD abroad is also an option, in case you'd rather pursue your studies outside the country. In that case, you can go through the options made available during your FYP, and if yours wasn't done abroad, you can go through the same search process, reaching out through LinkedIn, other websites, and referrals. That said, landing a PhD at a great venue is a bit harder from Algeria, so one common alternative is going abroad for another master's, using that time to look harder for a PhD, or even a direct job if you're no longer interested in academia. That time you spend looking is also an investment in getting to know yourself well, since a PhD is a long commitment, so you'd better pick the right setting (supervisor, topic, venue, and so on) to make the most of it.

If you're fed up with studying and academia, you can pick a more hands-on career, putting your engineering skills and knowledge to work wherever you land, whether as a SWE or in more of an AI role, in any company that can make use of your services, locally or abroad, whether as a contractor or freelancer.

To be direct here, contrary to what our sweet news outlets keep parroting, ENSIA won't guarantee you a job, and rightly so, it shouldn't. The school's role is to give you the right tools and mindset to open up those options yourself.

That said, expect some state-owned companies to reach out for interviews, particularly those that have agreements with the school. You might consider them if that's what you're looking for, since they offer relatively stable jobs. Personally, though, I find their adoption of technology to be quite limited, let alone their AI strategies. You'll probably notice this fairly quickly when you

meet their HR representatives and managers. I'm hoping this changes in the future, because these are the companies that citizens interact with on a daily basis. They could benefit enormously from greater digitization, and I say that as a firm believer in technology.

The private sector is another option. I've seen quite a surge in job offers on LinkedIn in recent years, and some are genuinely interesting to work for, given their product, the environment they operate in, and the growth plan they have for you as a fresh graduate. So check for these factors, and more, to make sure you're picking the right one. You could also work on your own company, if it ends up doing well, with clear signs of profitability along the way. I know it's the model the school likes to promote, though I haven't seen many pull it off so far.

Freelancing is also on the table, if you're after something less stable but with more income (in theory). The catch is that it takes some communication skills alongside the technical ones, since you'll need to land clients, manage them, manage your own time and effort, and turn it all into an actual business. I won't detail how each step works, since I'm far from an expert here, but I've come to realize that your network, the people you meet at the school, might be your best bet for landing new clients, or any other opportunity. So prove your technical chops, one way or another, and lean on the people you know to land your next opportunity, if this is something you're into.

Those are the options I had in mind and wanted to document, but as I said earlier, you're free to explore on your own, at your own pace, and make up the plan that fits you best, which might be none of what I just listed.

The Album



Before publishing this, I reached out to the promo asking them to share random shots from the years they spent at the school, whether from the school, the rooms, anything really that would reflect the experience they had, aesthetically. I was hoping to get from that 30 images at most, to fit around 10 pages or less. What I did not expect was people jumping in to gather over 240 images, leaving me with a hard job deciding what to put in the final version, and what to leave out. However, after a lot of thinking about how the book size would balloon, I opted to show all of them, out of respect for the people who collaborated, and to show you what their lenses captured for the memory!

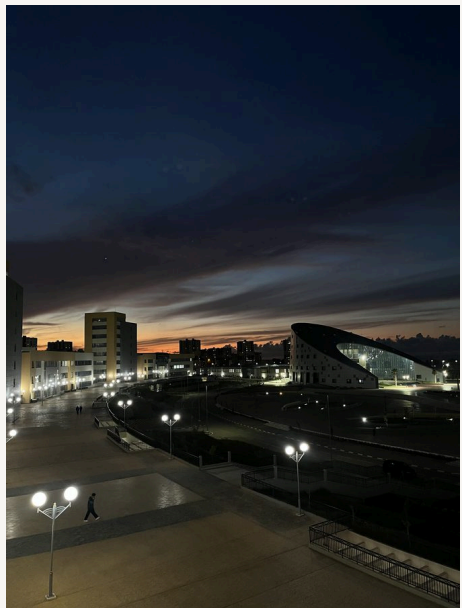
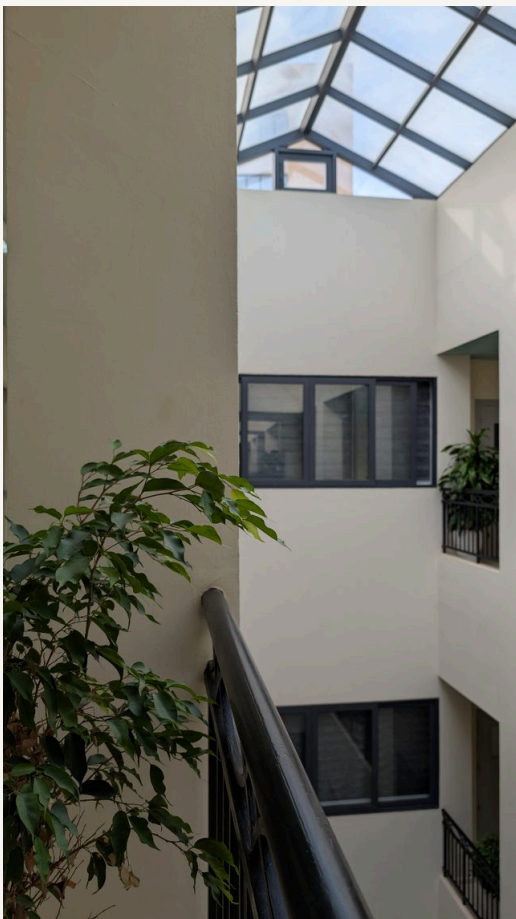
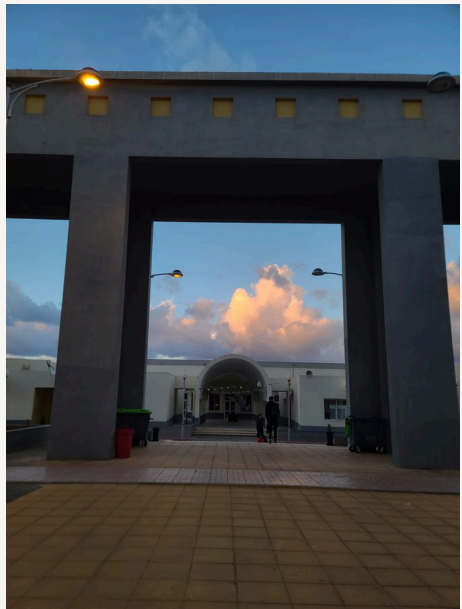
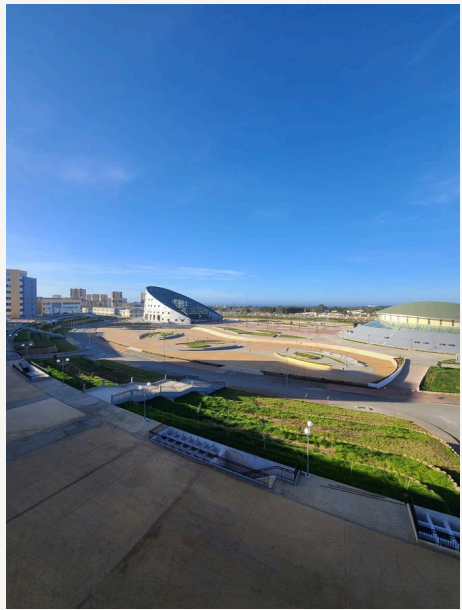
Before going there, I would like to thank them, one by one, and all the promo for their support with this. That said, sharing a photo here does not mean they agree or disagree with anything in this book, as none of them were shown its content beforehand.

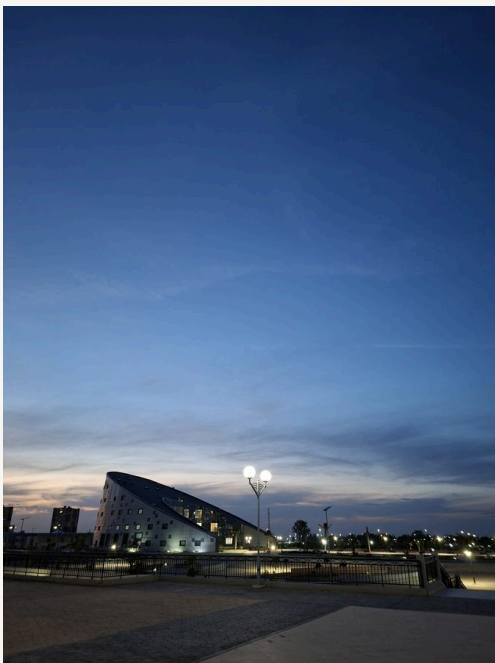
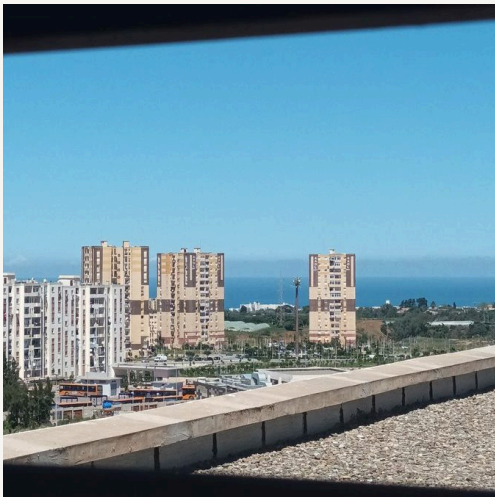
Here they are, in alphabetical order:

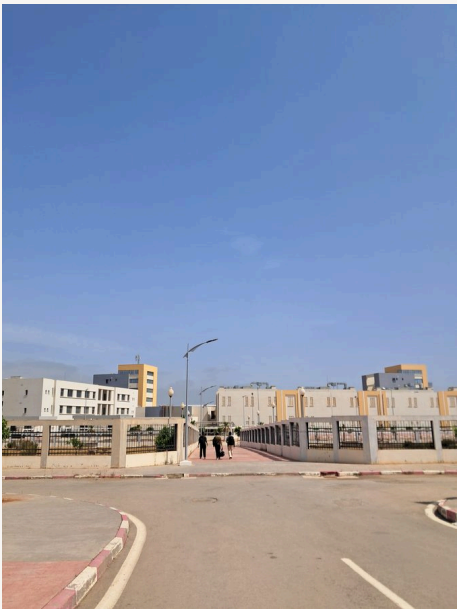
Abdenour Bouziane
Anfal Korichi
Bochra Mira
Chiraz Litim
GDG Club Admin
Haroune Rezki
Larbi Said Chikh
Loukmane Daoudi
Maab Chaoui
Mohamed Abderraouf Bendi
Mohanned Kadache
Nour El Yakine Guendouz
Rayan Zakaria Hassici
Sarah Mahmoudi
Selma Khelif
Sohaib Mousselmal
Younes Barmaki

I leave you with 66 pages of their making!



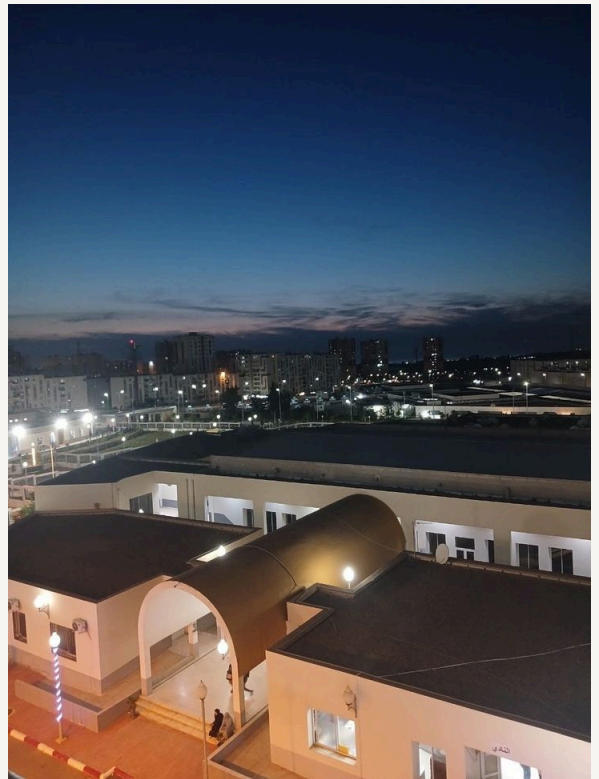




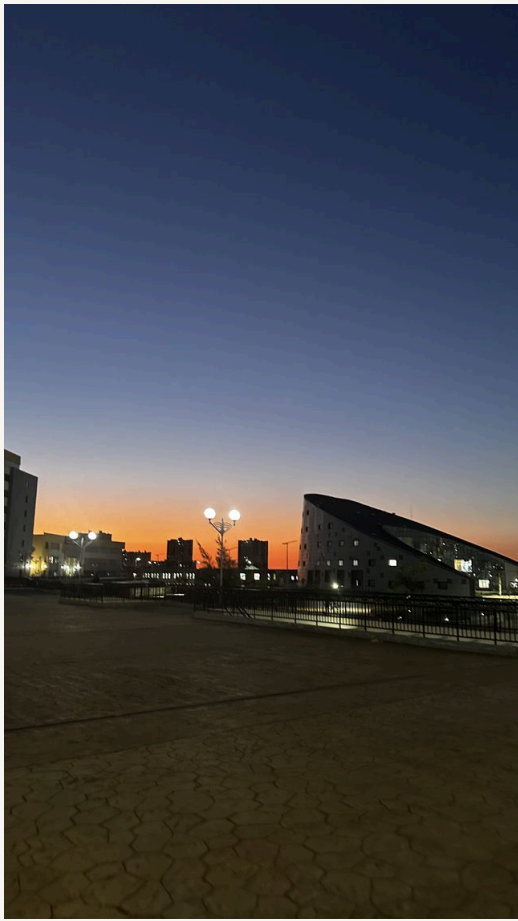


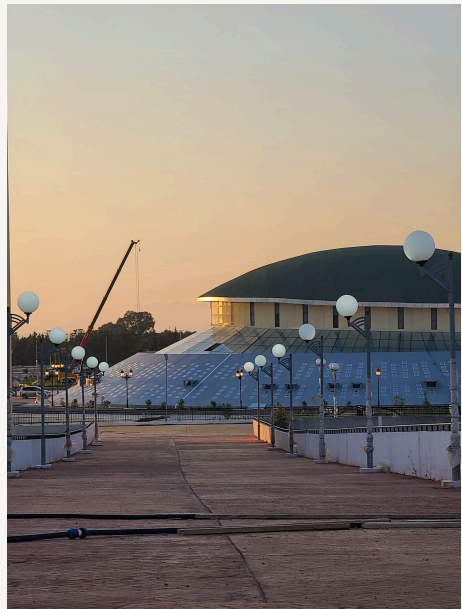
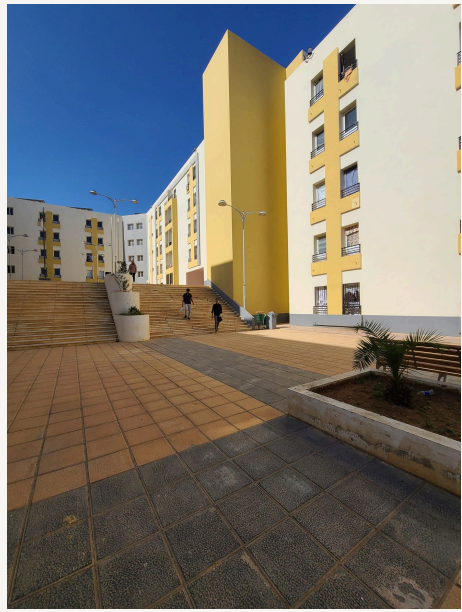
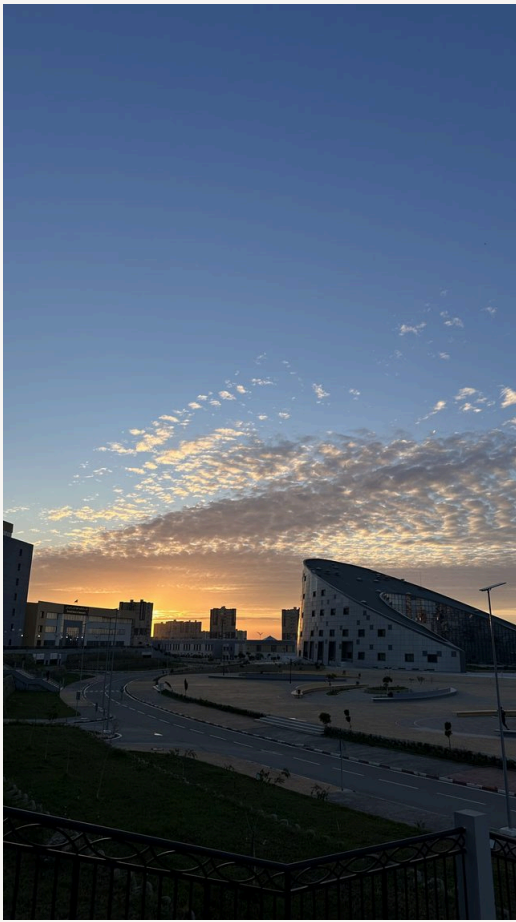










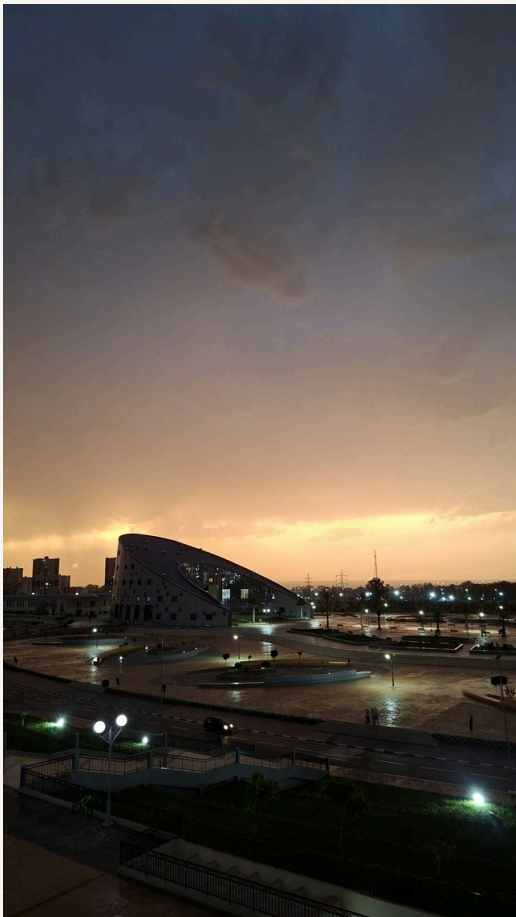


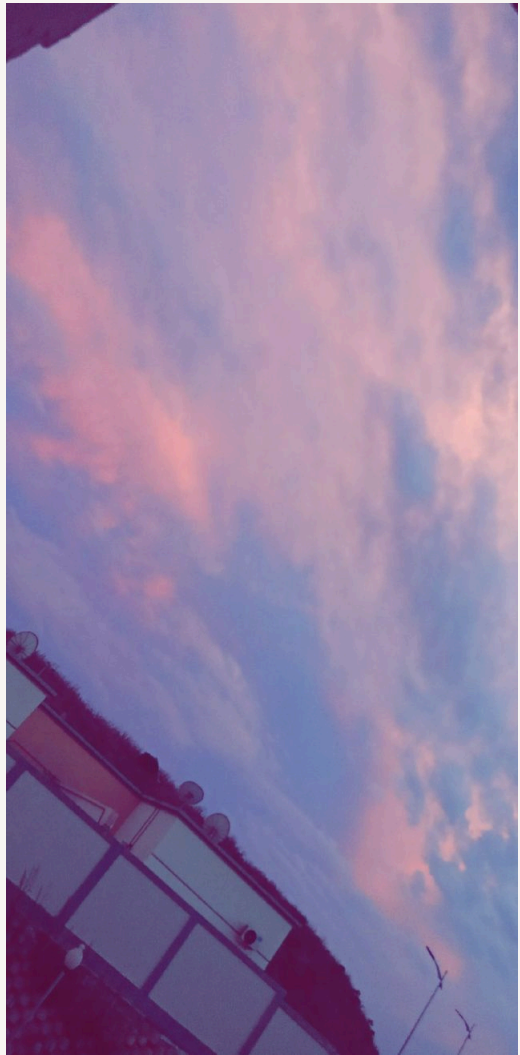
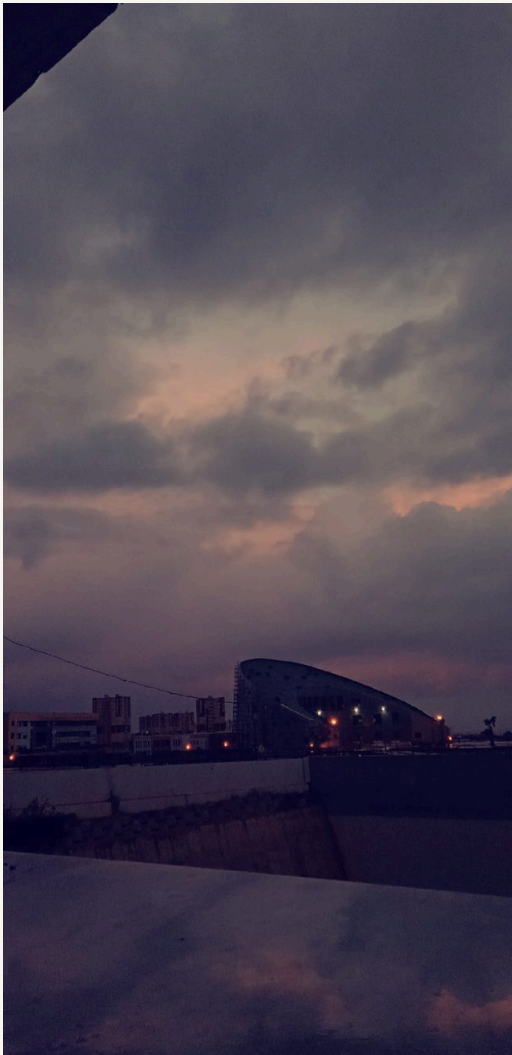


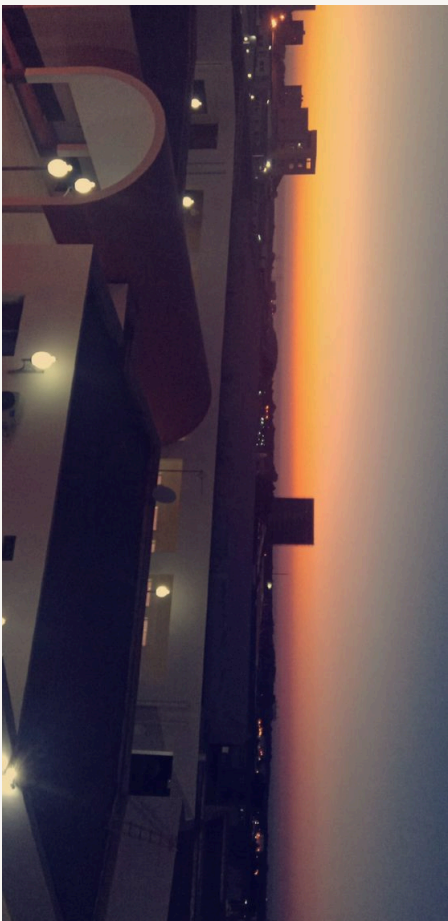
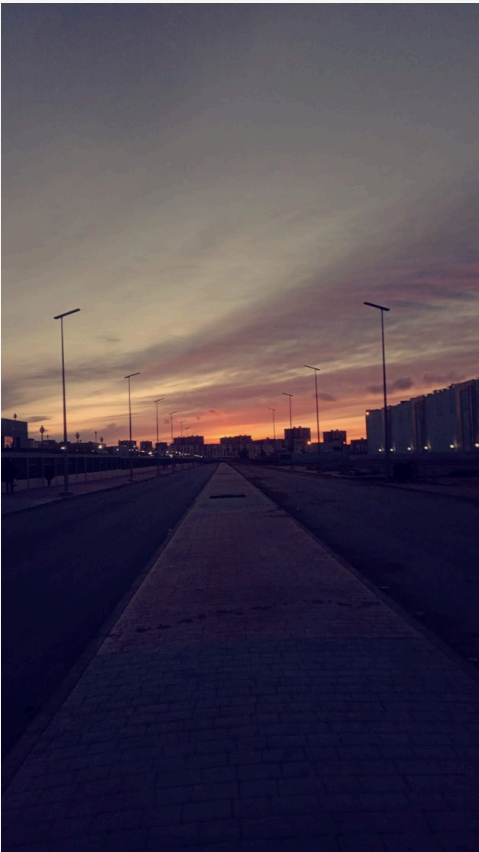
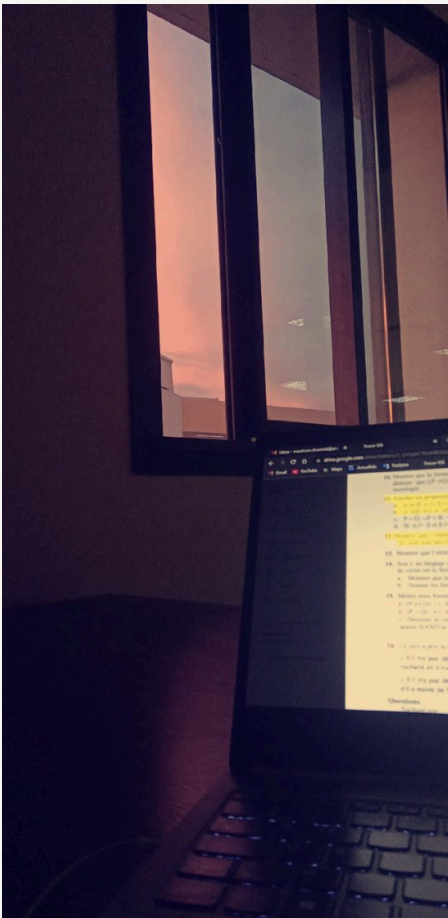


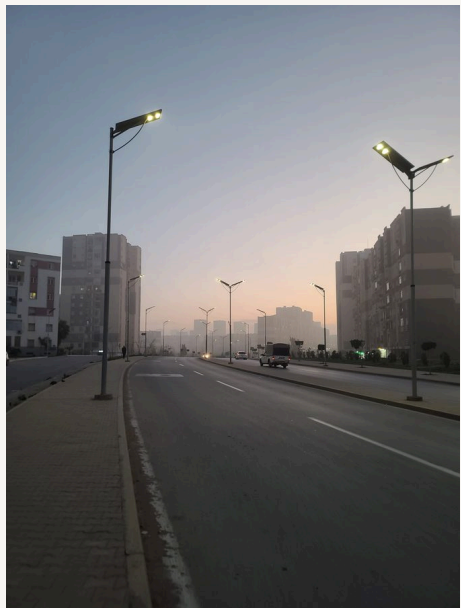
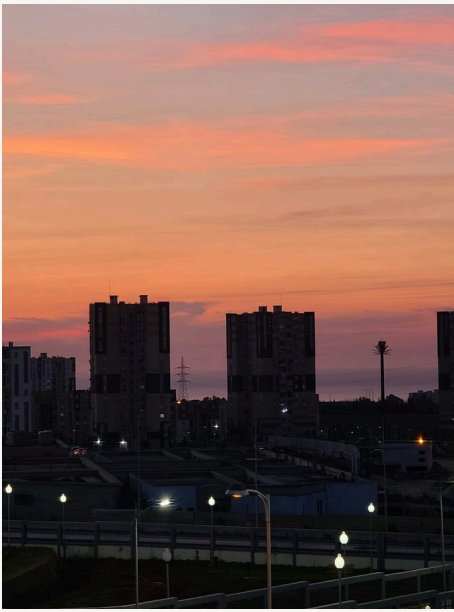


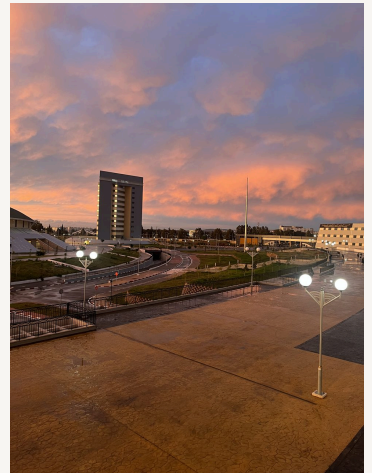
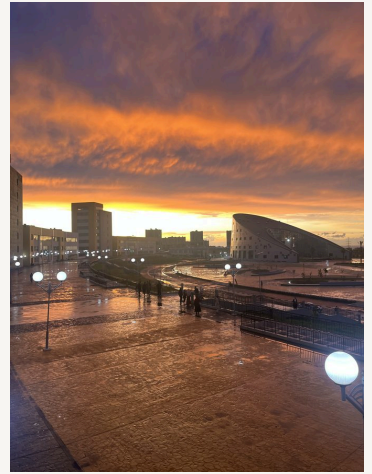
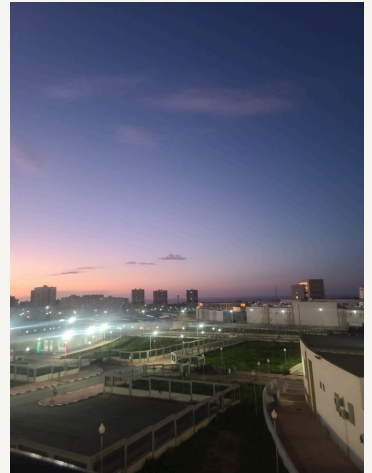


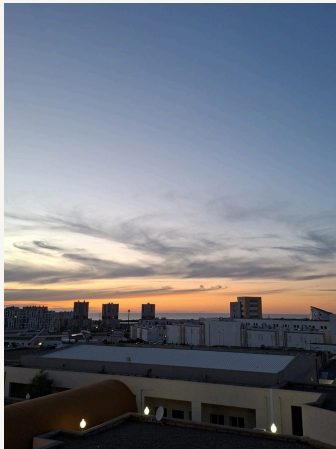






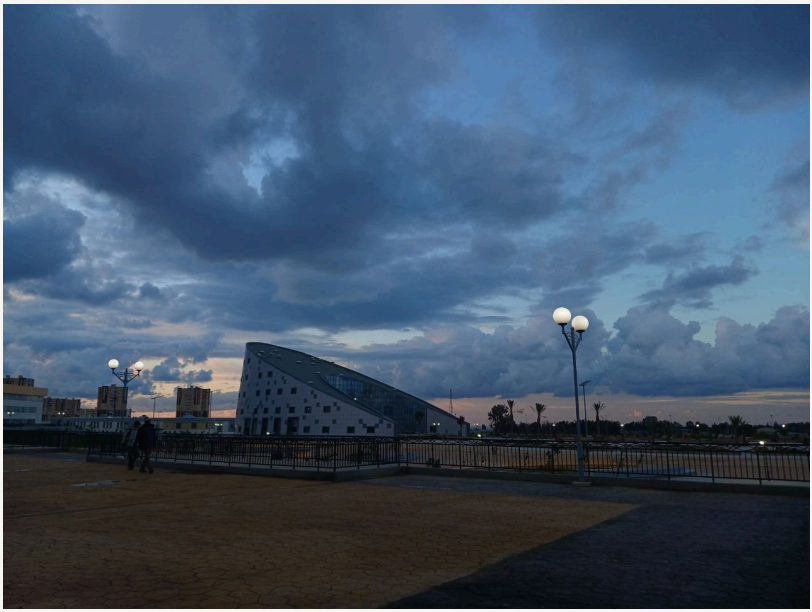


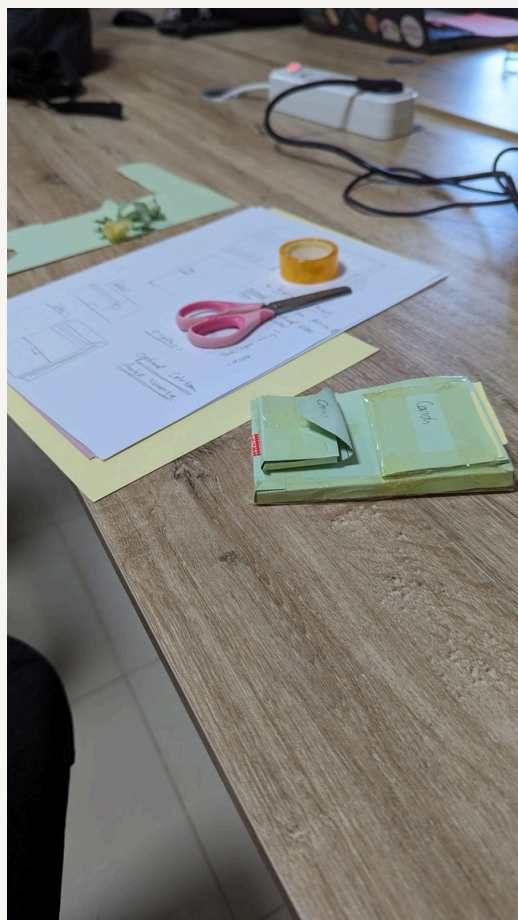












الجمهورية العربية السورية
وزارة التعليم العالي والبحث العلمي
مديرية التعليم العالي والبحث العلمي
إدارة القبول والتسجيل

الاسم الكامل :
رقم الامتحان :
المادة :
التخصص :
الدرجة :

الخامسة عشر

طلب قبول في الإقامة الجامعية
Demande d'admission en Résidence
Universitaire
POUR ANNÉE 20... à 20...
معلومات عامة

NOM :
Prénoms :
Date et lieu de naissance :
Nationalité :
Adresse :
Nom du père :
Num. de la mère :
Profession du père :
Profession de la mère :
N° de Tél. des parents :
E-mail :

الخامسة عشر

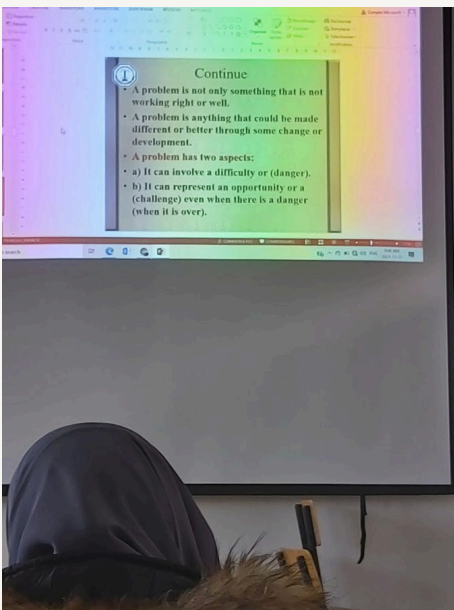
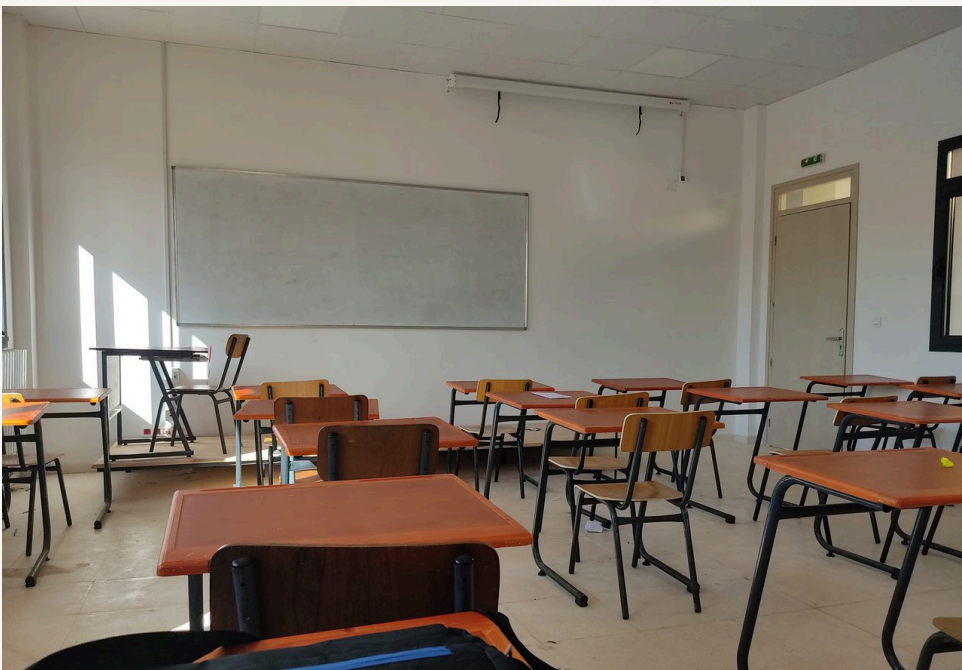
Année d'étude :
Année de BAC :
Filière d'étude :
Année d'étude :

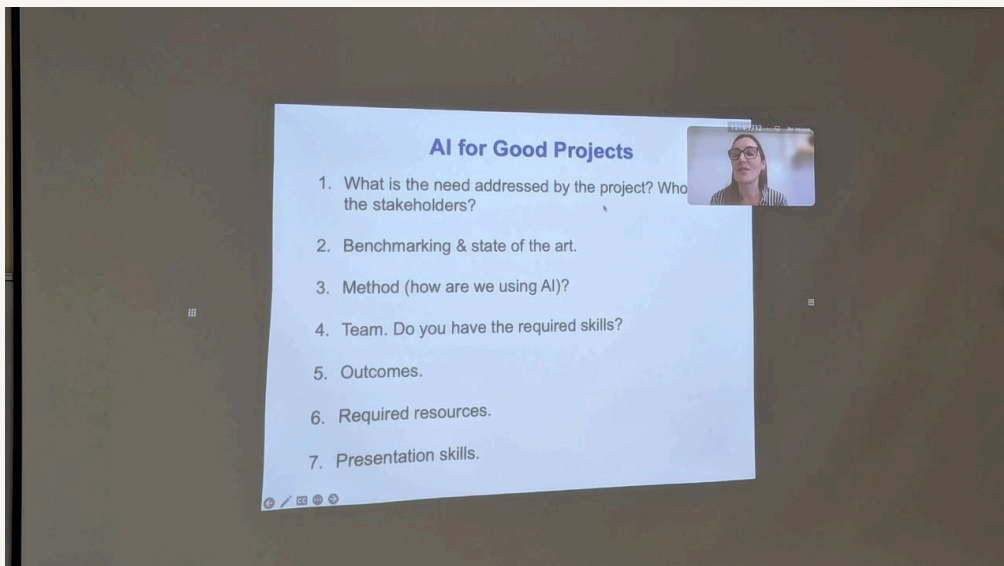
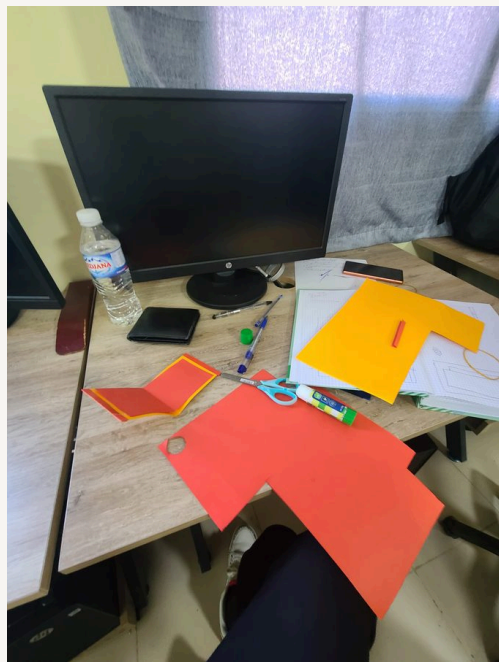
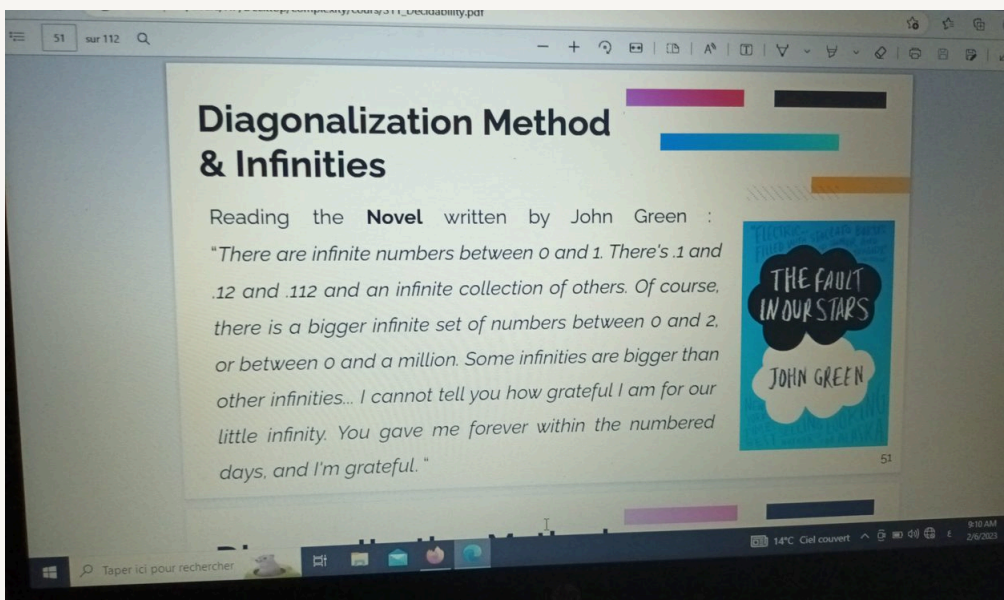
الخامسة عشر

Je soussigné Mlle :
أؤكد صحة المعلومات المعطاة أعلاه وباحترام القانون المعمول به في
إقامة الجامعة الذي سأقيم بها واحترام القوانين المعمول به في
إقامة الجامعة الذي سأقيم بها واحترام القوانين المعمول به في

إمضاء الطالب (ة)

[illegible]



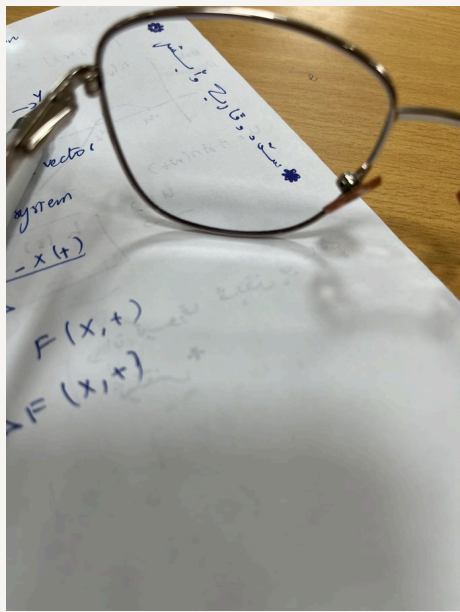


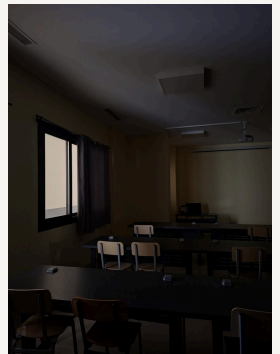
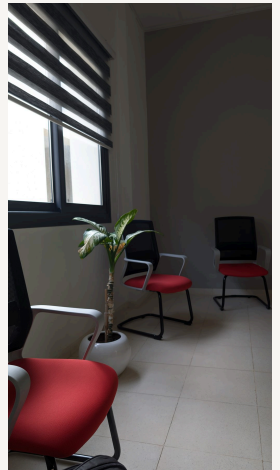
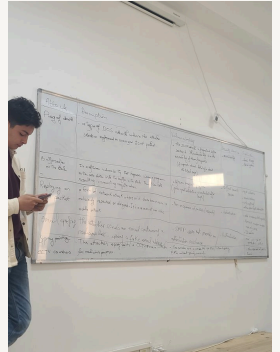

```
all Fedora Media W x LAB_8_Testing.ipynb x Colab Notebooks - Co x LAB7
https://colab.research.google.com/drive/1w7Dlame-1n0p8cabbLr-ey4G0gZimdsrallto...

Help All changes saved
+ Code + Text
print("Overall accuracy on test dataset: (overall accuracy:22f) ")
else:
print("No valid images found across all students. ")

Processing student: Selmani Rym
Accuracy for student Selmani Rym: 59.00%
Processing student: Ilmain Hiba
Accuracy for student Ilmain Hiba: 12.73%
Processing student: HAMMAD Zineb
Accuracy for student HAMMAD Zineb: 38.00%
Processing student: Selma Mani
Accuracy for student Selma Mani: 59.00%
Processing student: KHEZZANE Dina
Accuracy for student KHEZZANE Dina: 20.00%
Processing student: Mohmoud Badlis
Accuracy for student Mohmoud Badlis: 11.00%
Processing student: Abdeljalil Taibi
Accuracy for student Abdeljalil Taibi: 39.00%
Processing student: ATAMNA Alaeddine
Accuracy for student ATAMNA Alaeddine: 47.00%
Processing student: KARA LAOUAR Manel
Accuracy for student KARA LAOUAR Manel: 44.00%
Processing student: Baghdadi Kamar HibatAllah
Accuracy for student Baghdadi Kamar HibatAllah: 35.00%
Processing student: Bilal Chouche
Accuracy for student Bilal Chouche: 24.00%
Processing student: Kaouther Bensedira
Accuracy for student Kaouther Bensedira: 43.00%
Processing student: Mahmoudi Sarah
Accuracy for student Mahmoudi Sarah: 34.00%
Processing student: Sohaib Mousselmel
Accuracy for student Sohaib Mousselmel: 26.00%
Processing student: Lina Ahlem Boulam
Accuracy for student Lina Ahlem Boulam: 63.00%
Processing student: SAIDOUN Seif Allah
Accuracy for student SAIDOUN Seif Allah: 46.00%
Processing student: Souila BOUADJINA
Accuracy for student Souila BOUADJINA: 23.00%
Processing student: Abdoun Ahcene Rayan
Accuracy for student Abdoun Ahcene Rayan: 37.00%
Processing student: Yasmine Mediri
Accuracy for student Yasmine Mediri: 65.00%
Processing student: ZERROUKI Fella Manel
Accuracy for student ZERROUKI Fella Manel: 29.00%

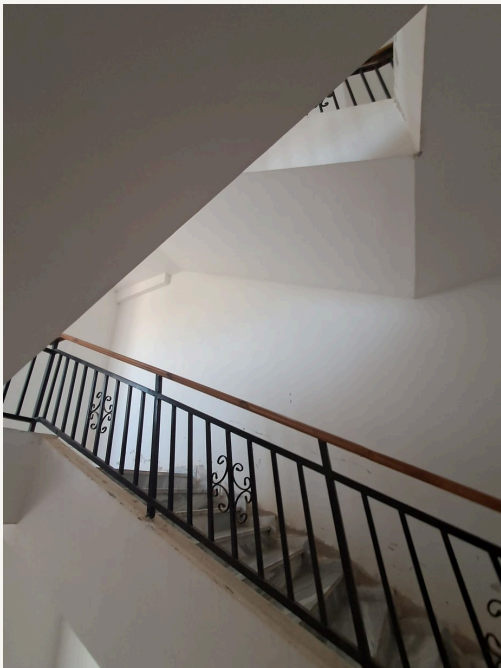
Overall accuracy on test dataset: 37.31%
2m 10s completed at 9:47 AM
```

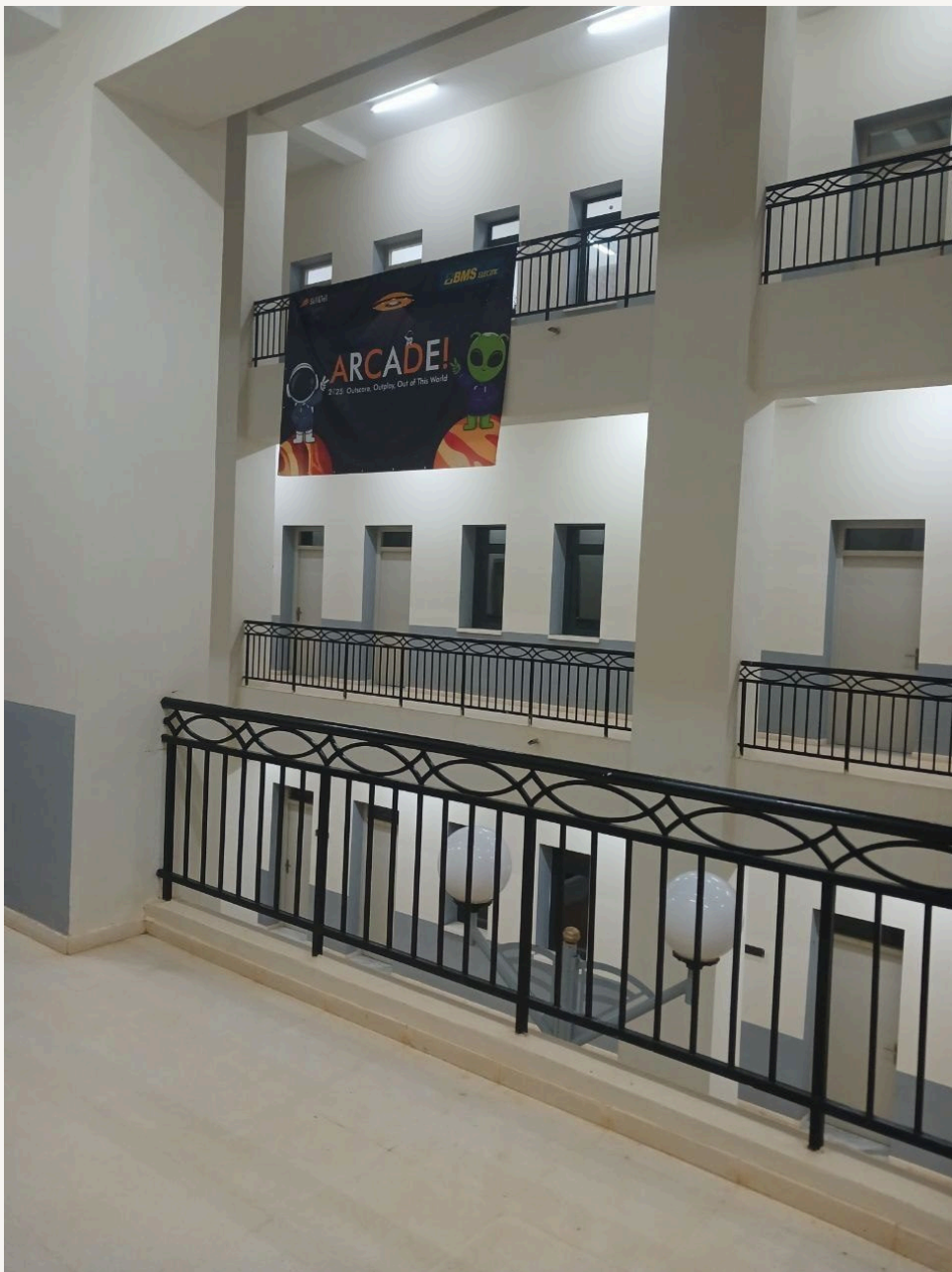
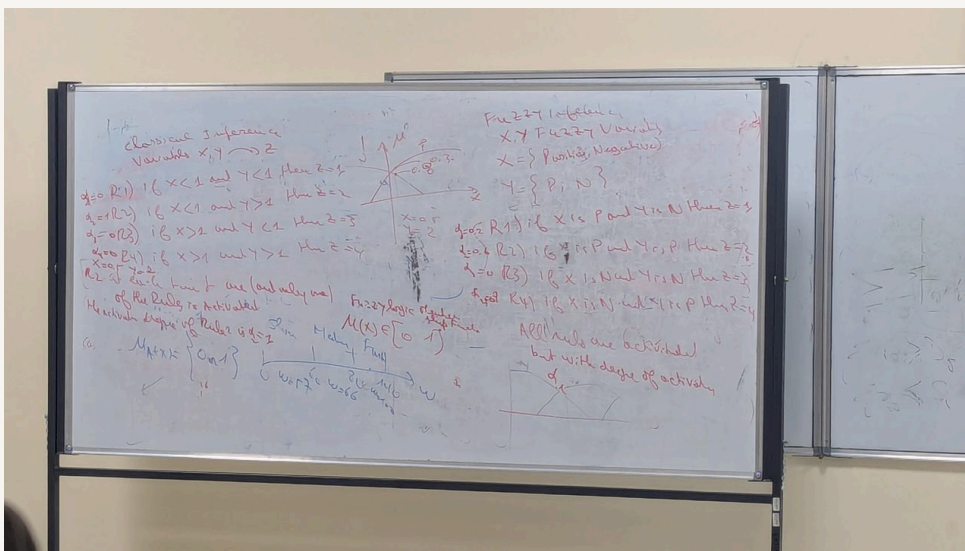


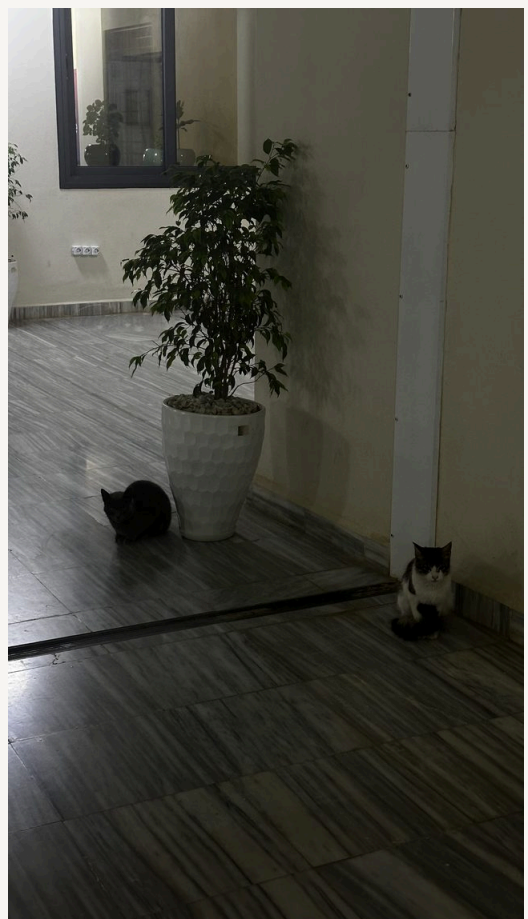
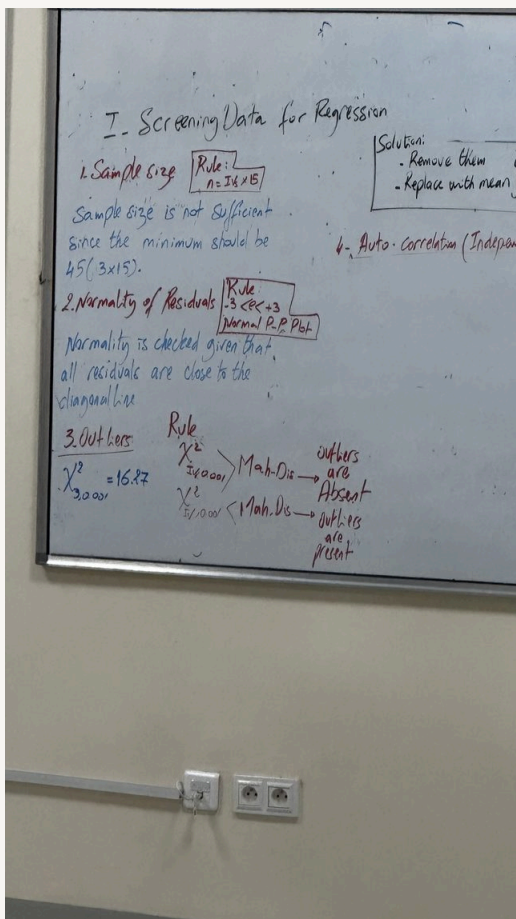
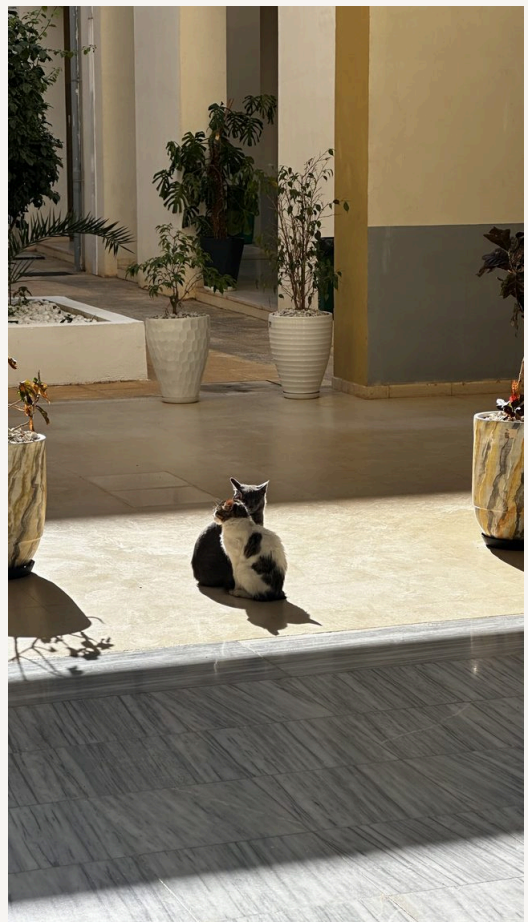
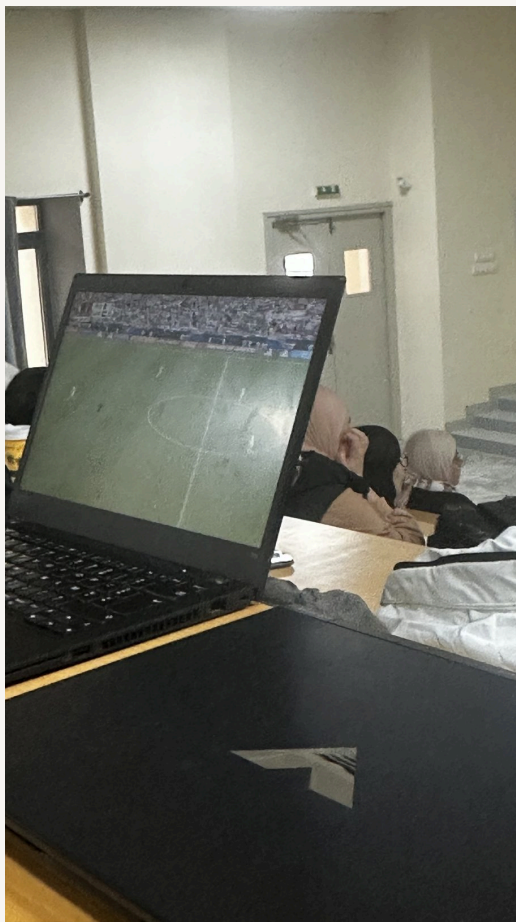


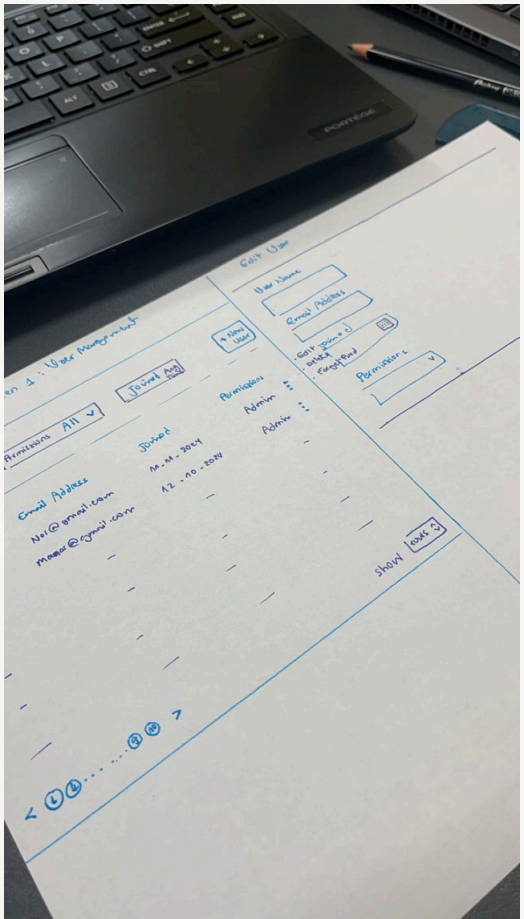
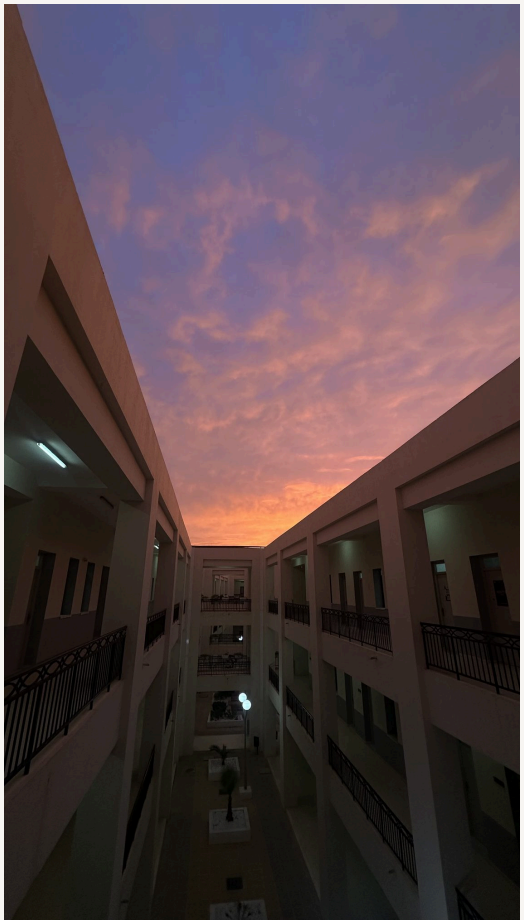
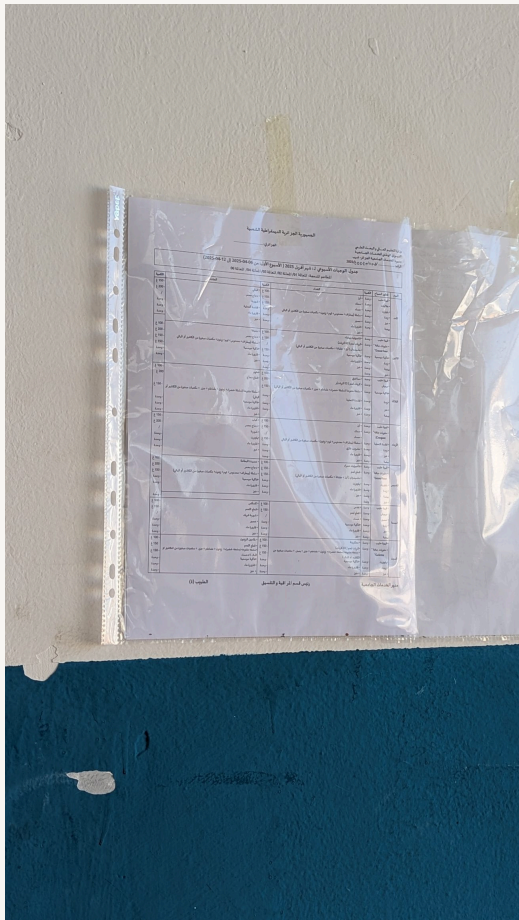


0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9

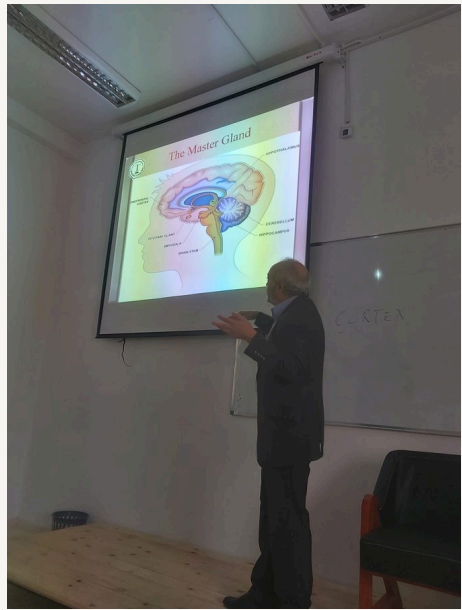




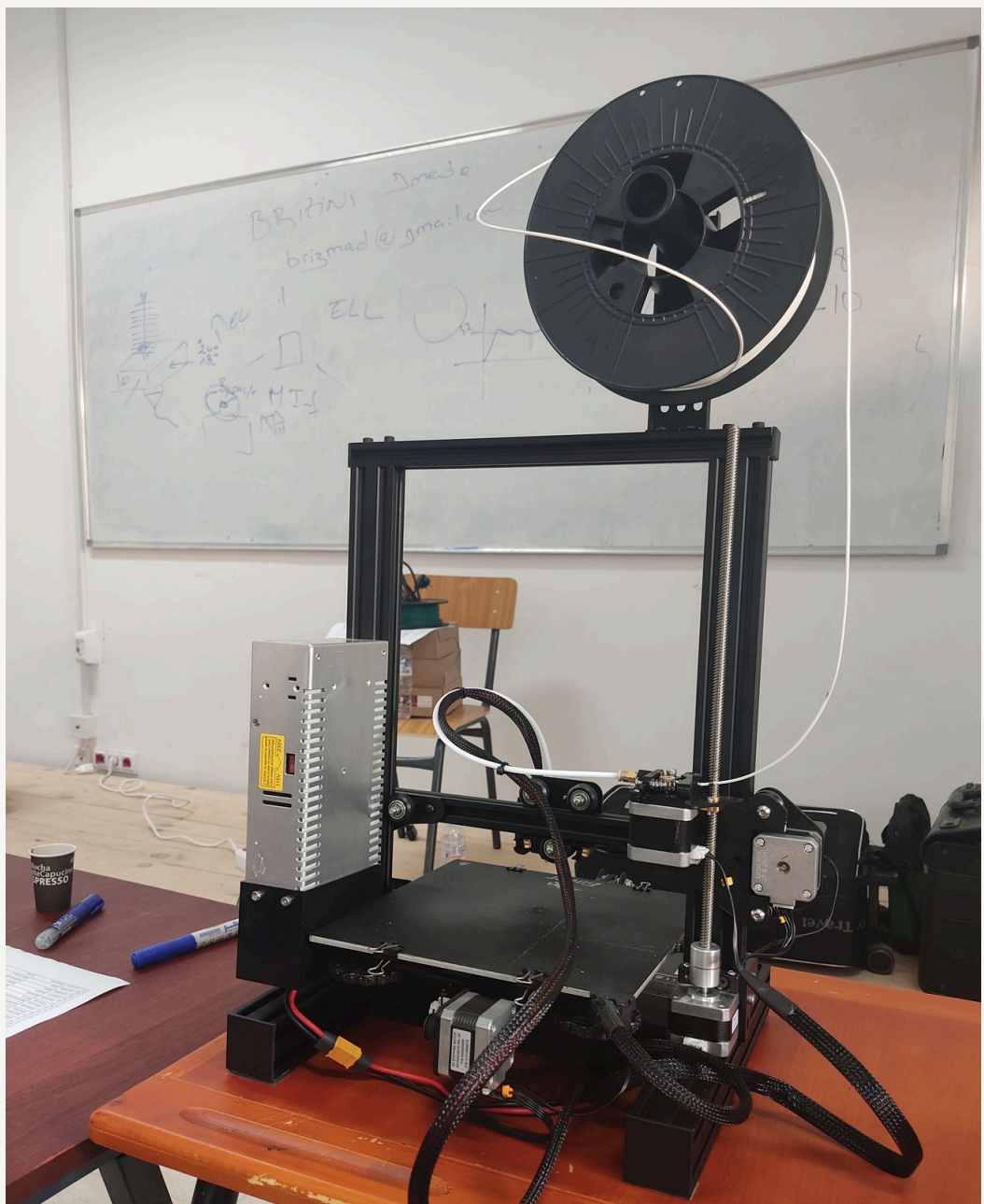
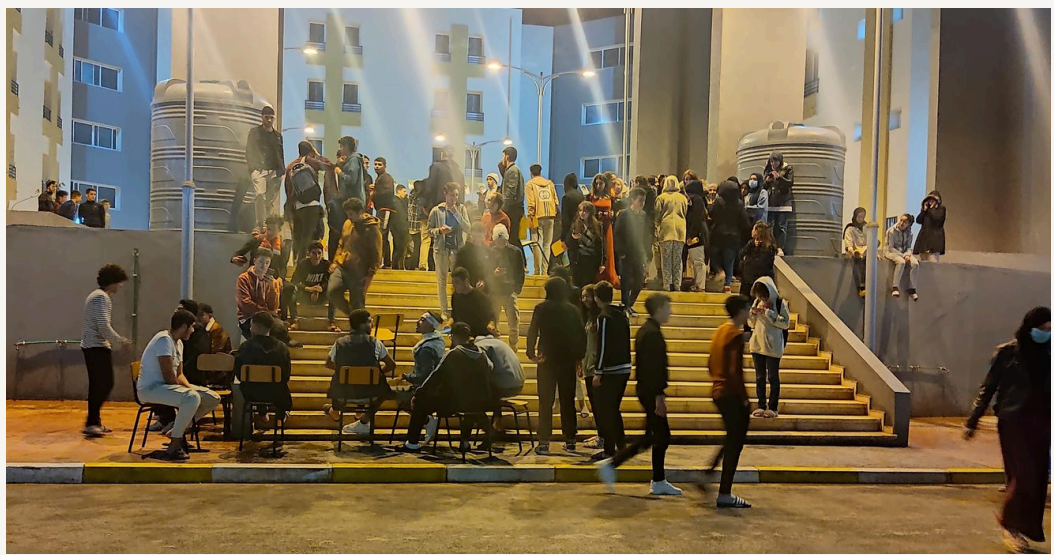


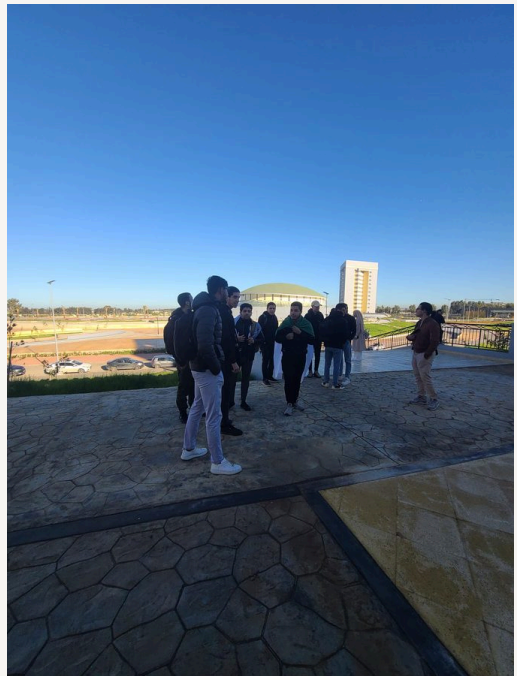


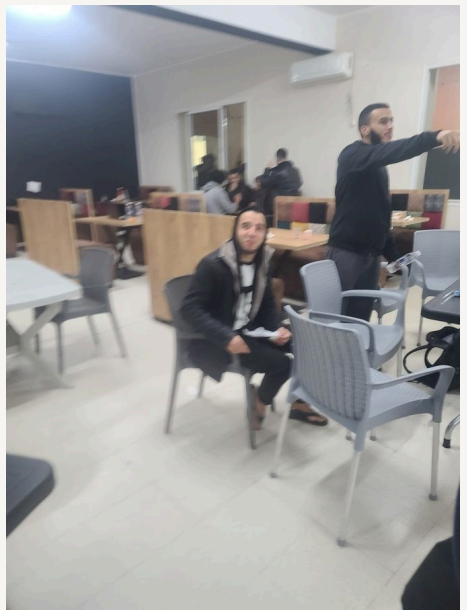


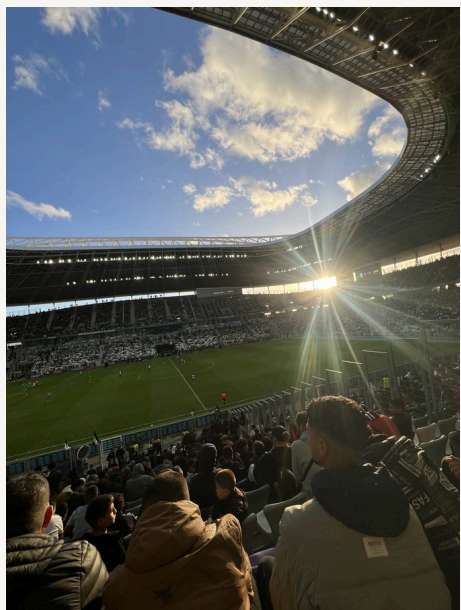
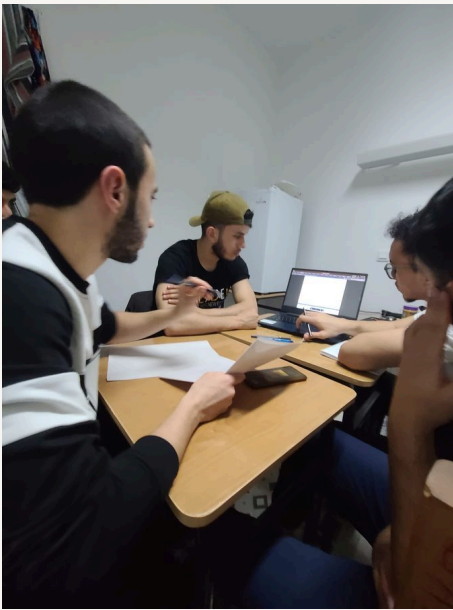


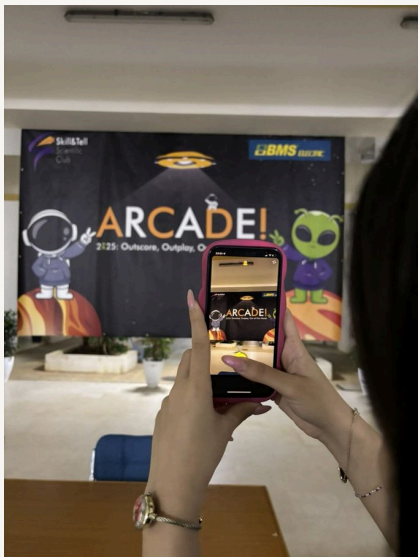




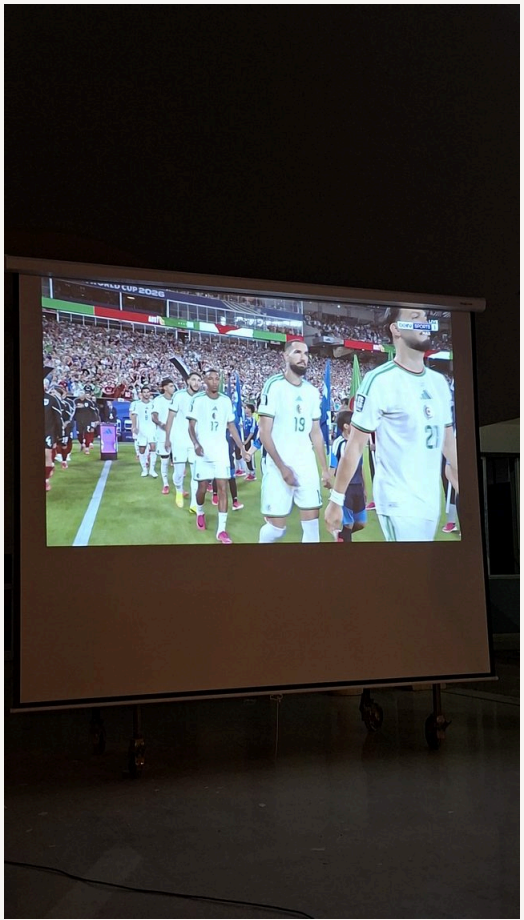


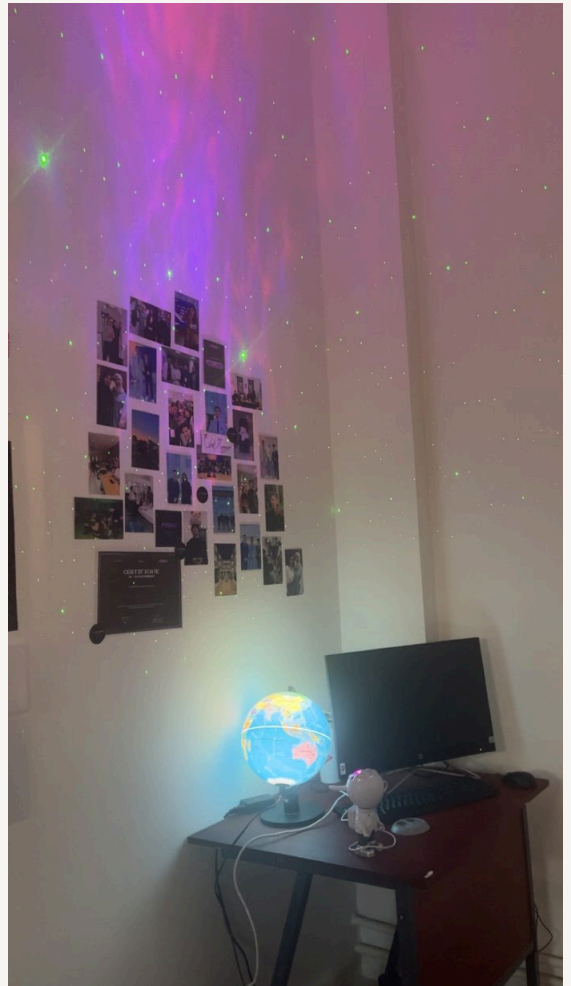
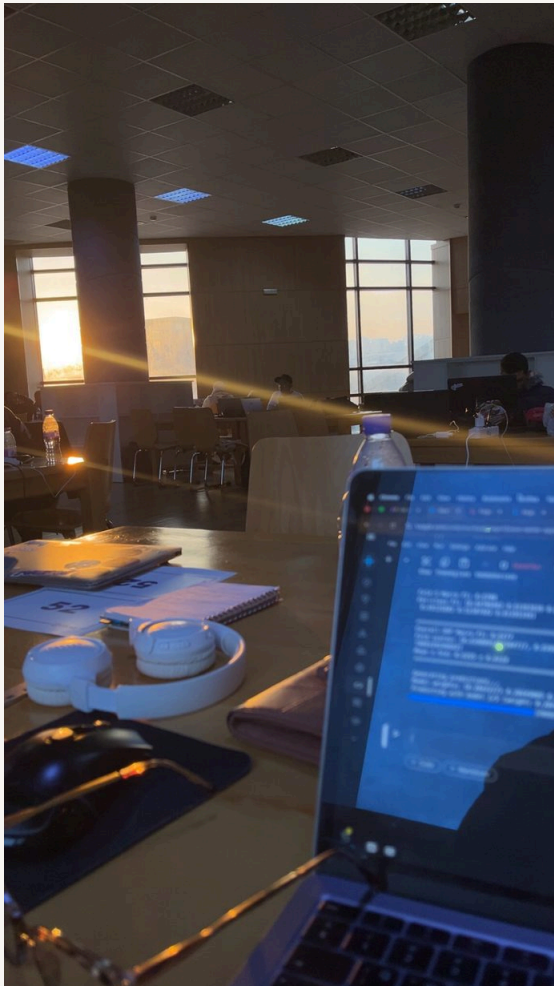
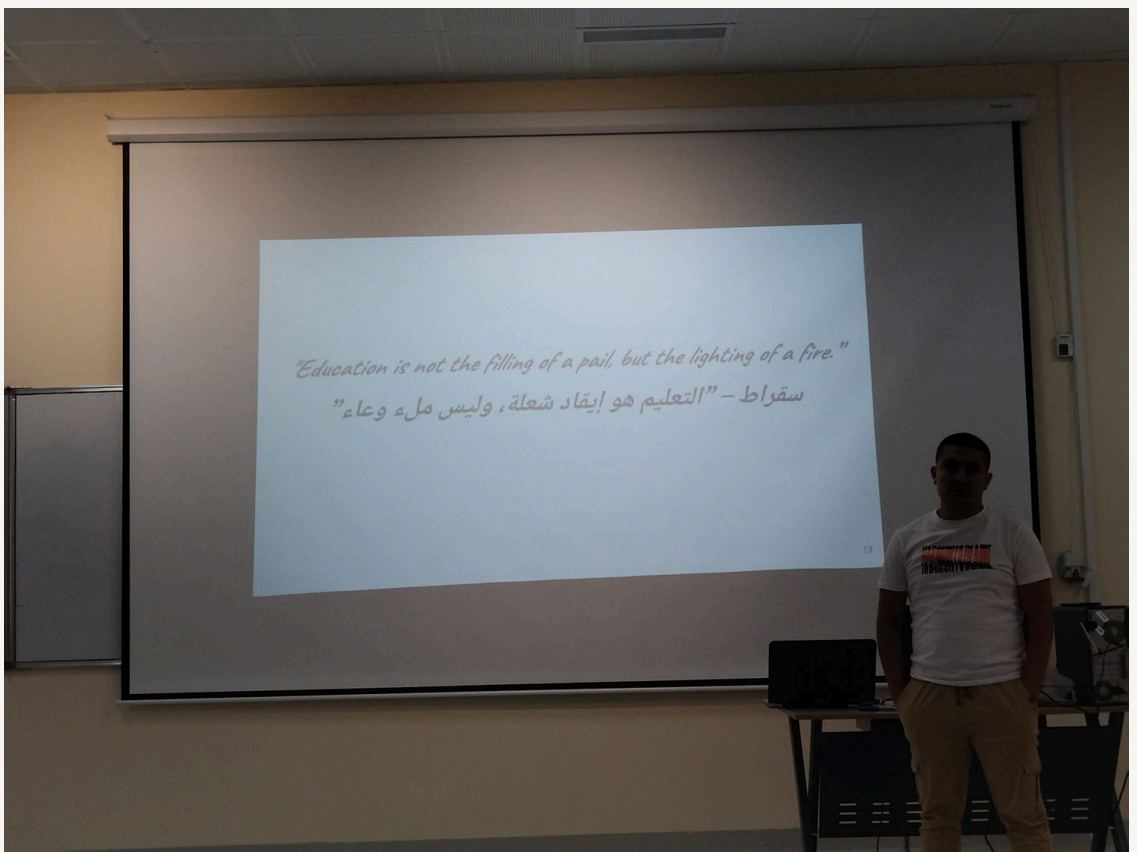




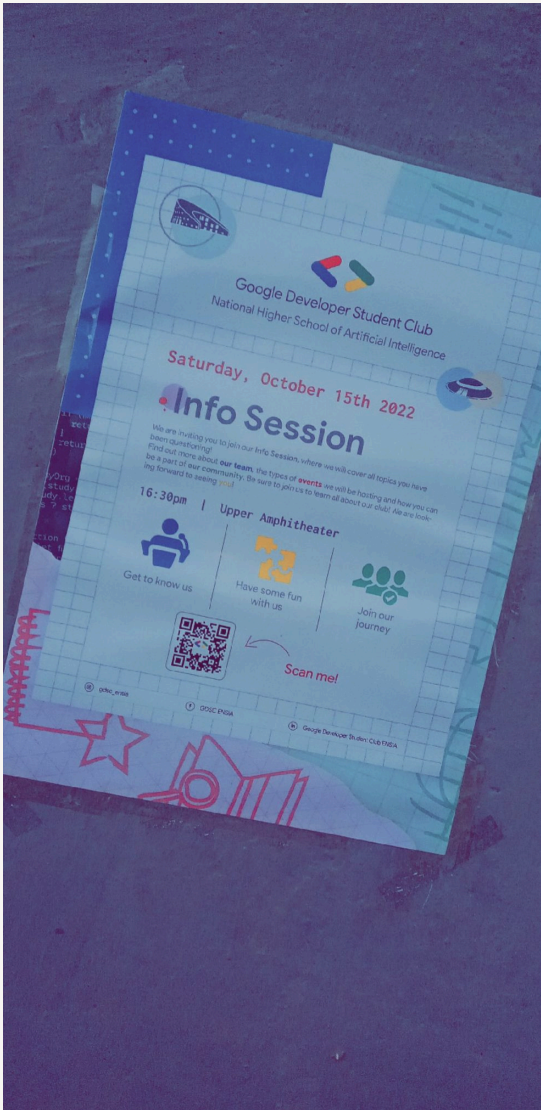








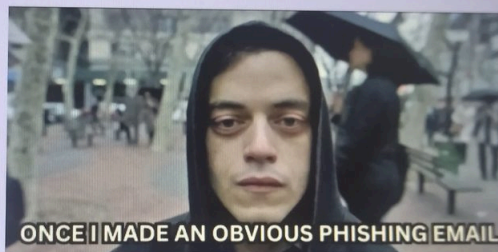


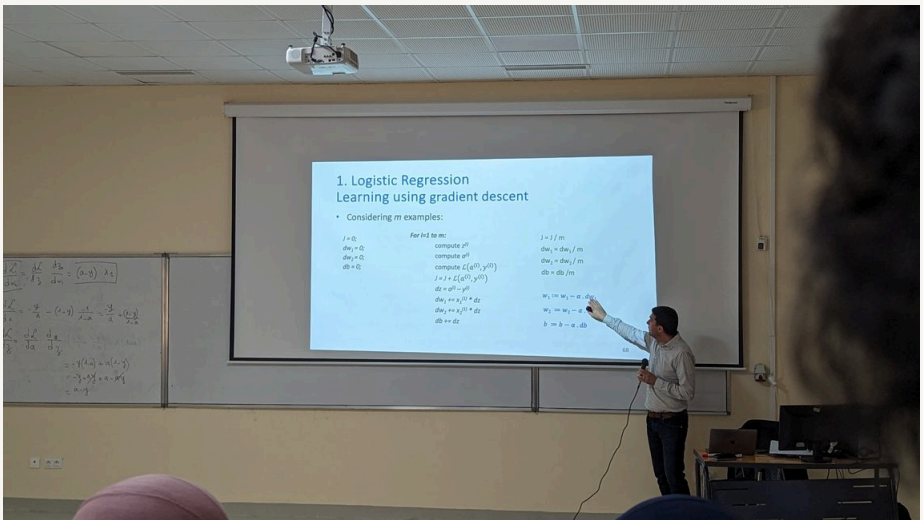


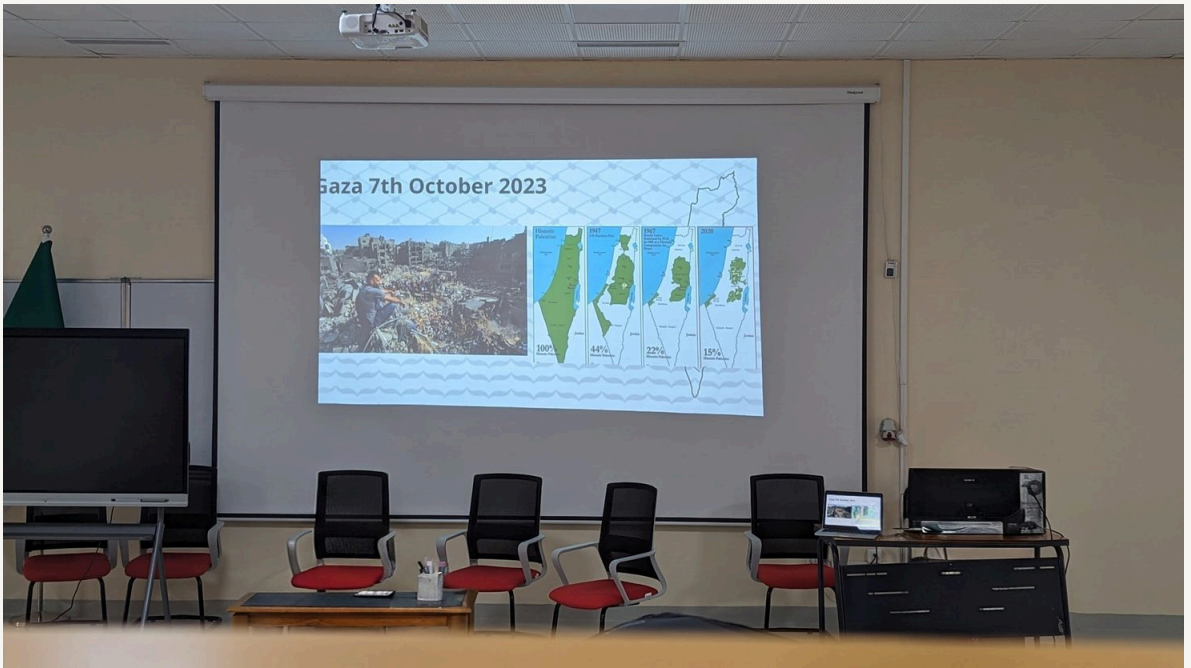


- ② **Social Aspect.** The users should be aware, trained, and sensitized about attacker, attacks, and their consequences:

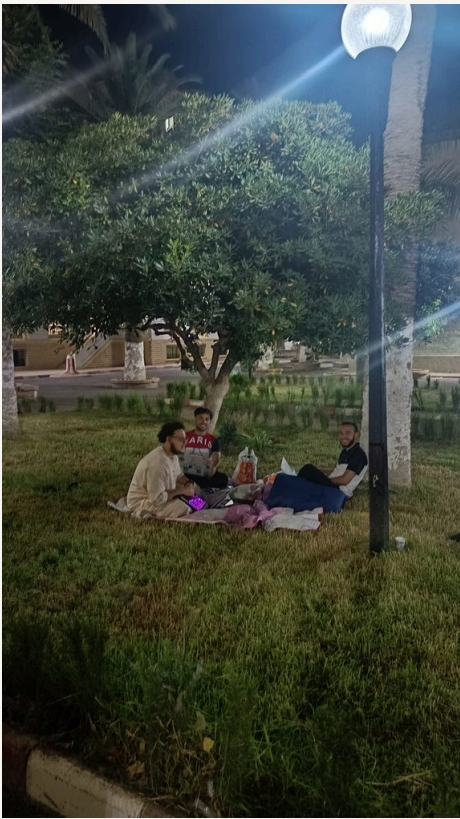
Hackers apply social-engineering techniques to compromise the social aspect. It is all about being smart to gain assets.





















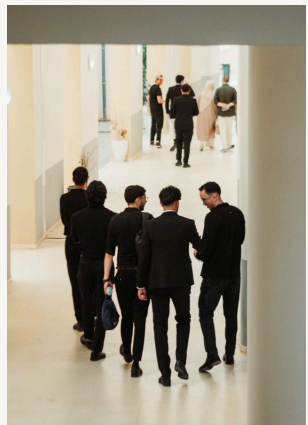




















ضرك يا الشبيبة الخدمة الوطنية تنادي
وبعدها ربي يسهللكم

1 h J'aime Répondre

THE ALBUM

Afterword

Civil Service

While I was writing this, a new presidential decree⁴² landed that changes part of the picture I painted in *Leaving ENSIA*, and even the choice to enter in the first place, though only for the promos coming after mine. As of the 2026 BAC intake, incoming students are asked to sign a commitment to five years of civil service after graduating, meaning working at a state-owned company or, at the very least, staying and working in Algeria. The decree isn't clear yet, given how recent it is, it hasn't been seen in action, and no loopholes are known for it either.



Figure 26: The decree in the official gazette, with the red arrow marking the five-year service clause.

Honestly, I'd been expecting something like this for a while, given the number of people leaving, promo by promo, and how many end up going abroad or refusing to work at local companies, mainly because the market isn't mature enough yet to offer high-quality roles or attractive pay.

It can also be understood from the government's point of view. Given the environment they fostered at Sidi Abdellah, the relatively high stipend, and

⁴²Presidential decree amending 23-416, *Journal Officiel* no. 44, 18 June 2026.

the overall quality of the students, they'd expect at least a return on their investment. That said, there are a huge number of issues with this new decree, and with the whole pattern of reaching for a sudden plug-pull when facing deep, complex problems, like why the best students tend to leave Algeria in the first place.

The biggest threat this brings is that it quietly filters who ends up joining the school. What I used to love most about these schools is that they bring together the brightest people from all around Algeria. We can still debate whether a BAC average is an honest measure of someone's level, but in my promo, the people who came were really brilliant, and it was a real challenge to keep up with their level and mindset.

The point is that with a hard limit like this, the best students, the ones really into CS and AI, won't choose ENSIA. They'll opt for schools that give them more freedom over their future. Because at that age, and in a field this new to them, people don't have much confidence in what they're picking, let alone in committing to where they'll work for five years after graduating.

Decision-makers also need to understand that people, at every level, choose their path based on opportunity, everyone wants to get the most out of where they are. So if so many students still aren't choosing the local market, we should be asking what's missing in the ecosystem, what still isn't appealing about it, and fixing that. Otherwise, we're just treating the symptoms instead of the actual cause.

It's also, unfortunately, become a habit to answer hard problems with quick, half-baked decisions, and I'm genuinely worried that ENSIA could become a place that fosters boomers⁴³, rather than the talent it should.

⁴³ *Kohoul* (كهول), in the local sense.

A Final Word

If you've made it this far, thank you.

I wrote this book because I wish I'd had something like it before starting at ENSIA, not a brochure or an official guide, but one person's honest attempt to explain what the place actually feels like, what deserves your attention, and what probably doesn't.

But this is still just one person's map, and yours will look different, since the school keeps changing, and the field even faster. Some of the things I was completely sure about will probably make less sense to you than they did to me, and that's fine. Keep what helps, and don't think twice about ignoring the rest.

If this book does one thing, I hope it saves you a few unnecessary headaches. And when you eventually leave ENSIA yourself, I hope you'll have something of your own to pass on to the people who come after you.

Before I sign off, I'll leave you with something Prof. Ahmed Guessoum⁴⁴, one of the greats, told me after graduating: never stop learning with the same ambition you had at the start of your journey, and especially with the same humility. It's always worth reminding yourself that getting into this school, or graduating from it, was never the goal in the first place, but a transition to something bigger, with more to take from it.

Good luck, take care of yourself, and make the most of the years you have there.

Salam!

⁴⁴<https://www.ensia.edu.dz/biography/ahmed.guessoum/>

AFTERWORD

Acknowledgments

This is the part where I'm supposed to thank everyone, for everything, and still end up missing someone, for some reason. I'll do my best not to.

When I was writing this, I didn't want mine to be the only pair of eyes on it. So I brought in people whose taste and opinion I trust, to get as many POVs as I could before putting it out.

Let me kick this off by thanking the reviewers and the early readers, who took the time to jump in while this was still rough. You'll also find, next to each name, the thing they're better than me at. So if you ever get stuck in one of those corners, reach out to them, and say I sent you, for whatever that's worth 😊

Starting with the reviewers, I'd like to thank [E'Salah](#) (AI and SWE), [Yasmina](#) (AI and DS), and [Larvi](#) (AI, DS, and Product Design). These guys had the nerve to read the whole thing, write down several rounds of feedback, and help shape the version you've just read.

Next, the people who got access to this earlier, who took the time to read it, shared their perspectives, and more importantly, helped me feel less cringy about the whole thing. Huge thanks to [Krimo](#) (Mechanical Engineering), [Brahim](#) (Automation Engineering), [Abdellah](#) (AI and DS), and [Wouroud](#) (all things English).

I'd also like to acknowledge what the album contributors brought to this. I asked naively for a few images and did not expect anything like what came back. So, really, thank you. I hope the moments your lenses caught along the way stay a trace of you at this school, for now and for the years to come.

A school is really its people, and the people at ENSIA were always at the center of the whole experience, from the students and the teachers to the administration, the clubs, and everyone working behind the scenes to keep the place running. Even if we didn't always agree, I'd like to thank them for their effort and dedication. They've been a big part of this, and I wish them all the best in what's to come.

The last thank you goes to my family, who were behind me through these years, and long before them. They've been the greatest support I've ever had, and I hope that never changes.

ACKNOWLEDGMENTS

About the Author



Hey, I'm Mohcen Taher Chouireb, and I'm into AI, software, and building things that solve real problems. I spend most of my time either developing new solutions, maintaining existing ones, or getting into side quests in the field, like this one, and others that I will try to document down the line.

I joined ENSIA as part of its very first promo, in September 2021. Back then the school was new, so new that the teaching system wasn't even settled yet, and all through our time as the inaugural promo, we went through constant changes: how the modules were placed, the curriculum, the whole setup, all of it tweaked to find the best combination for the promos to come. That made our experience a bit unique, since the batches after us had a somewhat different one, which is why the book flags these kinds of differences here and there. I tried to tell it from my own experience, since that's the only thing I can speak about with any confidence, and to point it out whenever I knew things were different for other promos.

These days, I'm a full-stack AI engineer, a role I grew into through internships and industry collaborations, where I've built and deployed everything from recommendation systems and Retrieval-Augmented Generation (RAG) assistants to document-understanding platforms, real-time voice agents, and graph-based machine learning frameworks. That work has come through partnerships with a number of companies, research labs, and public institutions, across educational technology, enterprise AI, conversational agents, and crowd analytics.

A stylized, handwritten signature in black ink, consisting of the letters 'MTC' in a cursive, flowing script.

ABOUT THE AUTHOR

Credits



Unless noted here, the photographs in this book are my own. Everything listed below belongs to whoever made it, and appears here for commentary, criticism, and identification, with no claim of ownership on my part.

The Album:

- Most of the photographs in the album were contributed by my promo mates, who are credited by name on the album's opening page. Those stay theirs, and are here because they agreed to share them.

Course textbook covers (shown to identify each module's main reference):

- *Mathematical Logic*, by Ian Chiswell and Wilfrid Hodges (Second Year).
- *Artificial Intelligence: A Modern Approach*, by Stuart Russell and Peter Norvig (Second Year).
- *Fundamentals of Machine Learning for Predictive Data Analytics*, by John D. Kelleher, Brian Mac Namee, and Aoife D'Arcy (Third Year).
- *Speech and Language Processing*, by Dan Jurafsky and James H. Martin (Fourth Year).
- *Reinforcement Learning: An Introduction*, by Richard S. Sutton and Andrew G. Barto (Fourth Year).

Photos from other sources:

- The Cisco Packet Tracer screenshot (Third Year), from howtonetwork.com.
- The ENSIA promo photo (Fifth Year), by Yasmina Medjri.
- The Jaguar supercomputer (Fourth Year), by Oak Ridge National Laboratory, licensed CC BY 2.0, via Wikimedia Commons.
- The portrait of Oussama Khatib (Fourth Year), by Andrew Lih, released under CC0, via Wikimedia Commons.

Logos:

- The Apache Hadoop, Apache Pig, and Apache Hive logos are trademarks of the Apache Software Foundation (Fourth Year).
- The Gostu logo belongs to Gostu (Third Year).

Memes and reaction images (used for parody and commentary, several made with imgflip):

CREDITS

- The P versus NP (Oppenheimer), “if I speak” (Mourinho), module-versus-exam, and panik/kalm images.
- Tom, from *Tom and Jerry* (Warner Bros.).
- “Hide the Pain Harold.”
- Pepe the Frog, created by Matt Furie.

Documents:

- The civil-service decree scan, from the *Journal Officiel de la République Algérienne Démocratique et Populaire* (no. 44, 18 June 2026).

Student-run resources (built and kept running by students, not by the school):

- The students’ Discord server is where most of the day-to-day chatter happens, and you can join it at discord.gg/FKAxe5Tea.
- Two student-maintained libraries of course material, ensia-hub.netlify.app and ensiapedia.onrender.com.



website: m0hc3n.com

linkedin: [mohcen-taher-chouireb](#)

instagram: [@m0hc3n_](#)

twitter: [@tahermohcen](#)