MAKEMYTECHNOLOGY

WIRELESS & COMMUNICATION ENGINEERING

MMT-GNU

Learn Wireless Communications by Doing

Student Handbook & Getting-Started Guide

210 hands-on experiments • 15 topic areas • 16 chapters

Real software-defined radios • Built-in simulator • XP, badges & leaderboards

Built on GNU Radio

1. Welcome to MMT-GNU

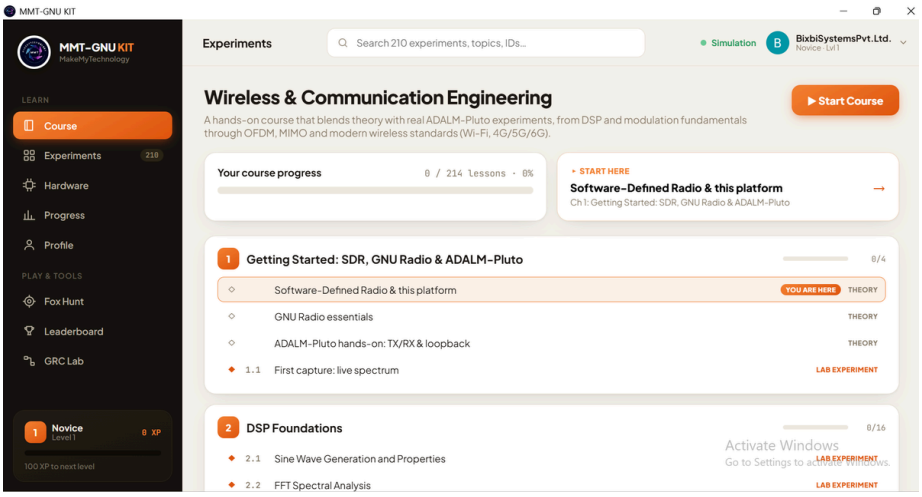
MMT-GNU is a complete learning platform for wireless and communication engineering. Instead of reading about modulation, coding, or MIMO and hoping it sticks, you build and run each idea yourself — on a real radio when one is connected, and on a faithful built-in simulator when it is not.

The platform is organised as a single course of **210 experiments** grouped into chapters. Every experiment is self-contained and follows the same rhythm: a little theory, a visual animation, a slide deck, a hands-on lab, a short quiz, and an auto-graded challenge that scores your result. You earn XP, level up, collect badges, and climb a class leaderboard as you go.

> **In one sentence**
> MMT-GNU turns a wireless-communications syllabus into 210 playable, measurable experiments — each one something you run, see, and are scored on.

What makes it different

- **Hardware-real.** 183 of the 210 experiments run on an actual software-defined radio (SDR) through a transmit-to-receive loopback — you transmit a signal and receive it back over real RF.
- **Works without hardware.** No radio? The built-in channel simulator runs every experiment on plain Python, on any operating system.
- **Learn by doing.** Each experiment is a guided, full-screen tutorial that walks you from concept to a working result step by step.
- **Gamified.** Profiles, XP, levels, badges, 64 auto-graded challenges, a class leaderboard, and even a Spectrum Fox Hunt keep the practice engaging.
- **Two ways to use it.** A full desktop app (with real-radio support) and a browser-based web edition you can sign in to from anywhere.



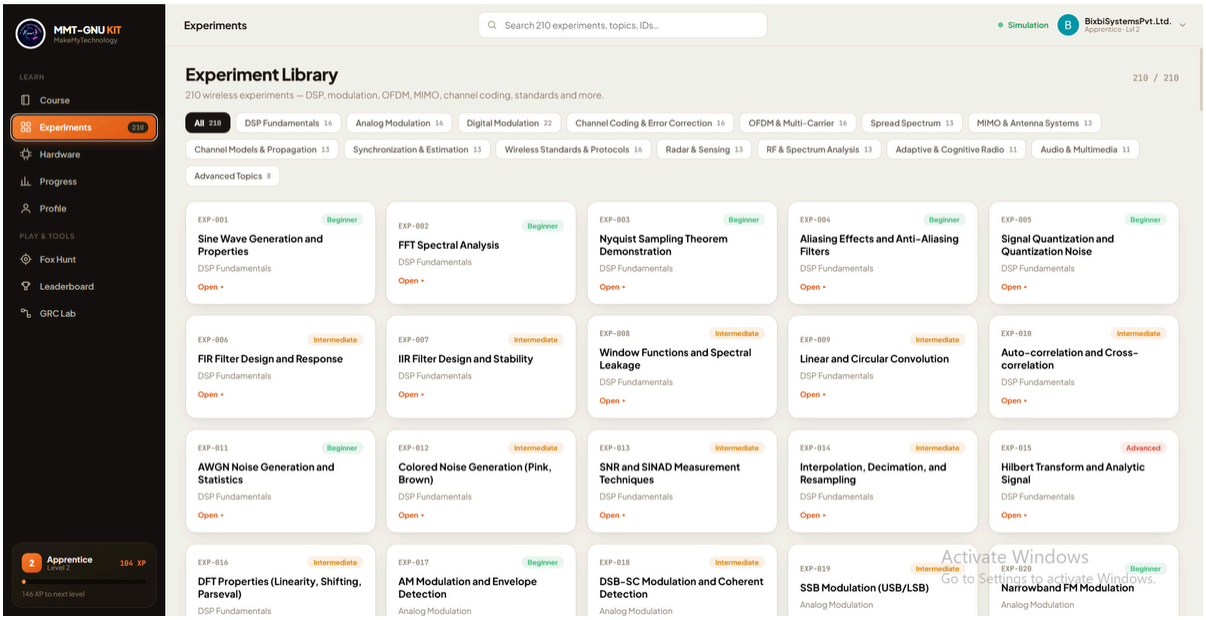
The MMT-GNU KIT app — your course, the experiment library, hardware, progress and the Play & Tools (Fox Hunt, Leaderboard, GRC Lab) all live in the left sidebar.

2. What You Will Learn

The curriculum spans 15 topic areas, from the fundamentals of digital signal processing all the way to modern standards and advanced research topics. Here is the territory you will cover, with the number of experiments in each:

Topic area	#	What you explore
DSP Fundamentals	16	Sampling, filtering, FFTs and the building blocks every radio depends on.
Analog Modulation	16	AM, FM and the classic analog schemes that carry sound and signals.
Digital Modulation	22	PSK, QAM, FSK and the digital schemes that carry data.
Channel Coding & Error Correction	16	Error detection and correction that keep data intact over noisy links.
OFDM & Multi-Carrier	16	The multi-carrier technique behind Wi-Fi, 4G and 5G.
Spread Spectrum	13	Robust, low-interference techniques used in GPS and CDMA systems.
MIMO & Antenna Systems	13	Multiple antennas for higher speed and reliability (true 2×2 on hardware).
Channel Models & Propagation	13	How signals fade, reflect and degrade in the real world.
Synchronization & Estimation	13	Locking onto timing, frequency and phase to recover a signal.
Wireless Standards & Protocols	16	How real-world systems (Wi-Fi, 4G, 5G, LoRa…) combine these pieces.
Radar & Sensing	13	Detection and ranging built from the same signal-processing toolkit.
RF & Spectrum Analysis	13	Reading the radio spectrum, power, and RF behaviour.
Adaptive & Cognitive Radio	11	Radios that sense their environment and adapt.
Audio & Multimedia	11	Capturing, processing and transmitting sound and media.
Advanced Topics	8	Deeper material that builds on everything above.

Across these 15 areas sit 210 experiments, arranged into a 16-chapter course so you always know what comes next.



The Experiment Library — filter by category and open any of the 210 experiments, each tagged Beginner, Intermediate or Advanced.

By the numbers

Figure	Meaning
210	Total experiments in the course
15	Topic areas / categories
16	Chapters in the guided curriculum
183	Experiments that run on a real SDR (176 single-antenna + 7 MIMO)
27	Experiments run as faithful simulations (antenna arrays, multi-node, capacity curves)
64	Auto-graded EVM / BER / SNR challenges

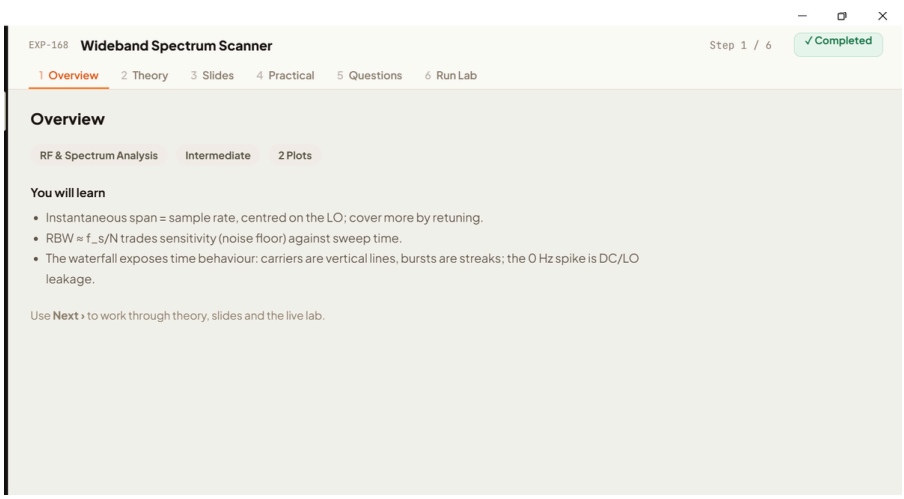
3. How Each Experiment Works

Every experiment opens as a guided, full-screen tutorial. You move through the same clear sequence each time, so once you have done one, you know how to do all 210. On the desktop app the flow is:

Step	What happens
1. Overview	A short orientation: what this experiment is about and why it matters.
2. Theory	A focused HTML lesson with proper maths (rendered with MathJax).
3. Slides	A slide deck summarising the key ideas.
4. Flowgraph	A generated GNU Radio flowgraph (.grc) you can open in GNU Radio Companion.
5. Practical	The hands-on instructions for the lab itself.
6. Questions	A short quiz to check your understanding.
7. Run Lab	You run it — on a real radio or the simulator — and watch it work.
8. Results	Your measured output, plus the auto-graded challenge score.

Along the way you get inline canvas animations, MathJax-rendered lessons, and a real `.grc` flowgraph you can open and tinker with in GNU Radio Companion.

> **On the web app**
> The browser edition uses the same idea with a course-first flow: Overview → Theory → Slides → Practical → Questions → Run Lab, with your lab results streamed live as plots while it runs.



Inside an experiment — the stepped tabs (Overview, Theory, Slides, Practical, Questions, Run Lab) carry you from concept to a working, completed lab.

4. Real Radios and the Built-in Simulator

MMT-GNU is device-agnostic: the same experiment can run on different radios, and on no radio at all.

Hardware mode — transmit-to-receive loopback

When an SDR is connected, 183 of the 210 experiments run for real. The radio transmits a signal and receives it back through a **TX → RX loopback**, so you measure genuine RF behaviour — 176 experiments on a single-antenna (SISO) radio and 7 on a 2×2 MIMO radio.

Simulation mode — no radio needed

The remaining experiments — true antenna arrays, multi-node scenarios and capacity curves — are faithful simulations, because they describe situations a single desk radio cannot physically reproduce. And when no radio is attached at all, the platform falls back to a built-in channel simulator that runs every experiment on plain Python, on any operating system. You lose nothing of the learning flow; you simply trade real RF for a modelled channel.

> **Good to know**
> Real-radio experiments run on the desktop app — a cloud server has no radio attached. A “bring your own SDR” local agent for the web app is planned, so watch for it.

5. What You Can Do

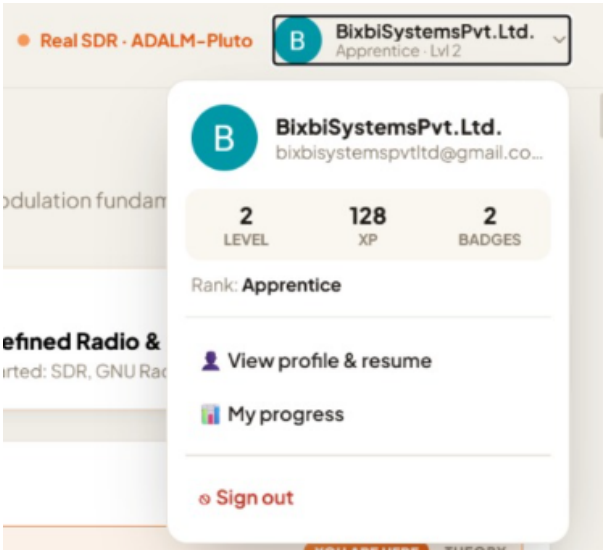
Beyond working through experiments, MMT-GNU gives you a set of things to build, earn and chase. Here is what is yours to do as you go.

Watch your signal come alive

Running a lab is not a black box. As an experiment runs, the platform draws what is happening — in the web app, live plots stream straight to your screen so you can watch a constellation tighten, a spectrum light up, or a bit-error curve settle as conditions change. You see cause and effect in real time, not just a final number.

Build your profile

Every student gets their own profile. As you complete experiments you earn XP, climb through levels, and unlock badges — so your progress is visible and yours to keep, not lost the moment you close the app.



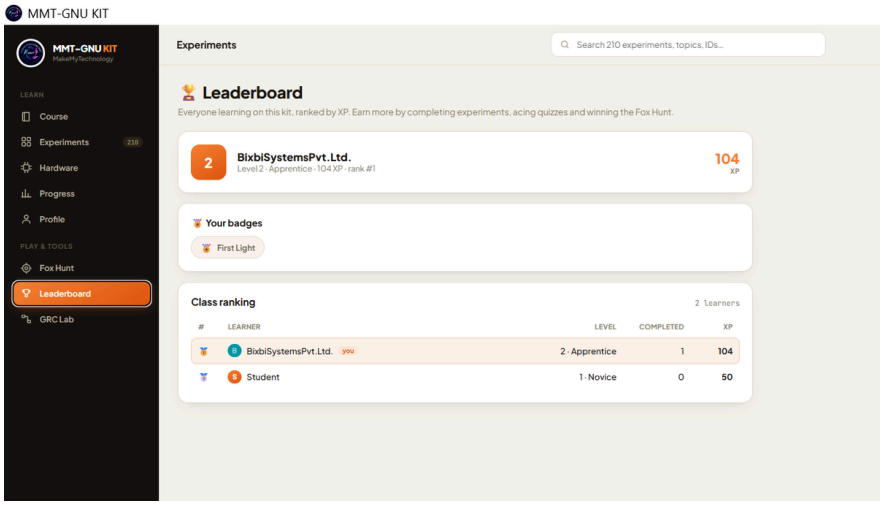
Your profile at a glance — level, XP, badges and rank, with quick links to your profile, résumé and progress.

Take on the challenges

Most experiments end in an auto-graded challenge **64 in all**. These do not just check that you clicked through; they score the real quality of the signal you produced, measuring things like EVM, BER and SNR. A strong score means you genuinely got the radio working. It is the difference between “I read about it” and “I made it work.”

Climb the class leaderboard

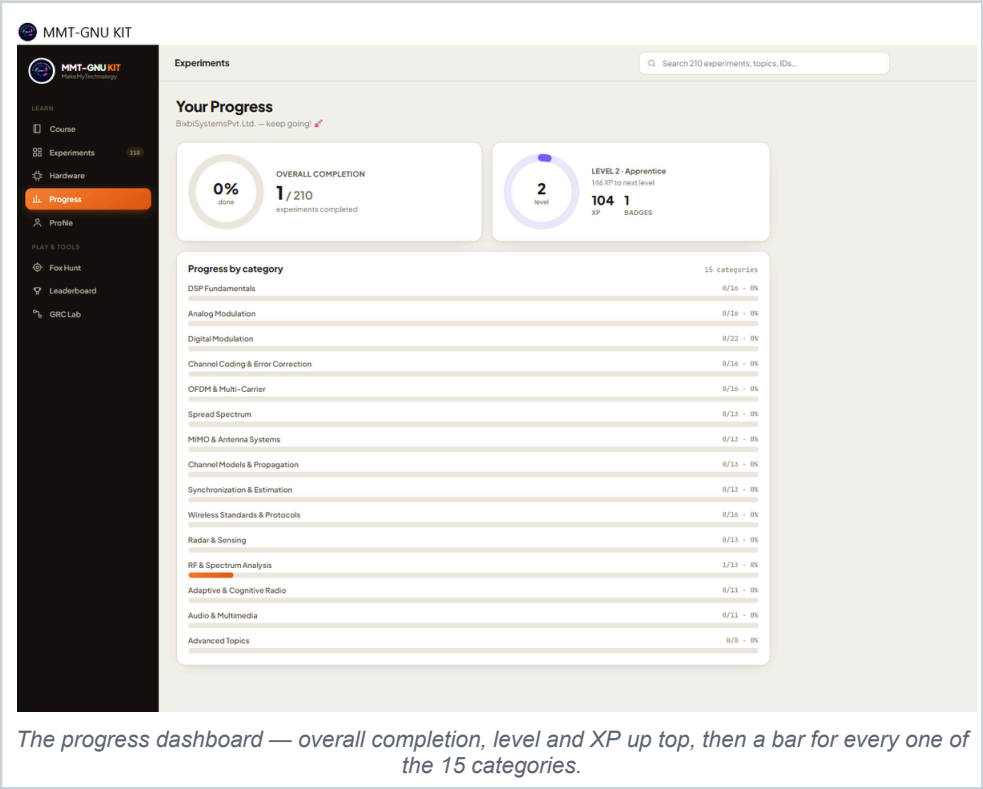
Your results feed a class leaderboard, so you can see exactly where you stand against your classmates. You earn your place by completing experiments, acing quizzes and winning the Fox Hunt — it turns a quiet evening of practice into a bit of friendly competition.



The class leaderboard — everyone on the kit ranked by XP, with your badges and standing front and centre.

Track your progress

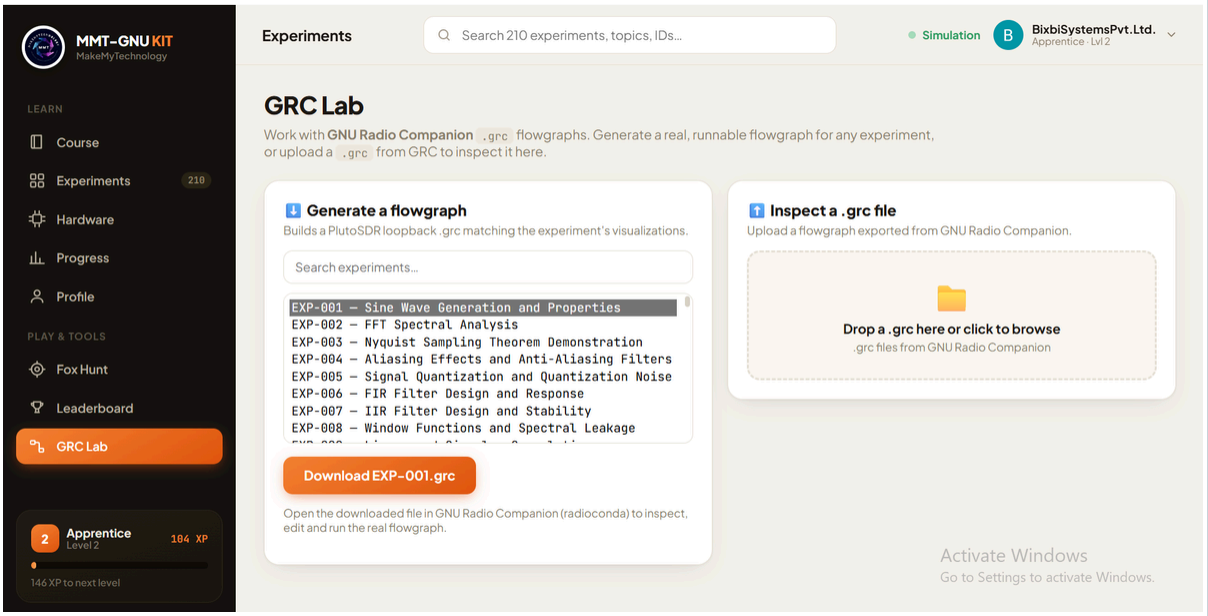
A progress dashboard keeps the whole journey in one view: your overall completion, level and XP, the badges you have collected, and a category-by-category breakdown of how far you have come. You always know what is done and what is next.



The progress dashboard — overall completion, level and XP up top, then a bar for every one of the 15 categories.

Open it in real GNU Radio (GRC Lab)

When you want to go beyond the guided lab, the GRC Lab lets you generate a real, runnable PlutoSDR loopback flowgraph (a `.grc` file) for any experiment and download it to open, edit and run in GNU Radio Companion. You can also upload your own `.grc` exported from GNU Radio Companion to inspect it inside the app. It is the bridge from learning the concept to building the real thing.



The GRC Lab — generate a runnable .grc flowgraph for any experiment, or drop in your own to inspect it.

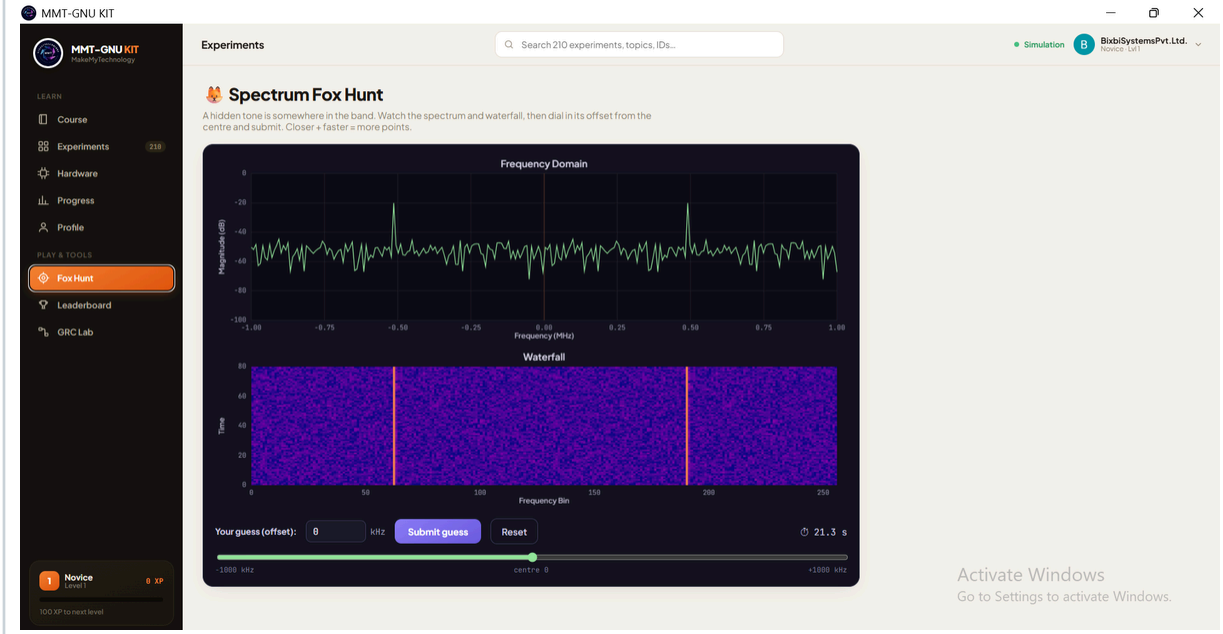
> **Why it is graded on real results**
> Because the challenges score the actual signal you achieve — not a checkbox — your XP, badges and leaderboard rank all mean something. They are proof you can do the work, which is exactly what makes them worth putting on a résumé (see Section 7).

6. The Spectrum Fox Hunt

The Spectrum Fox Hunt is MMT-GNU's most hands-on game. A hidden tone is transmitting somewhere in the band, and your job is to track it down — sharpening the real-world instinct of reading a spectrum and hunting for a transmitter, the same skill engineers use to chase down interference.

How it plays

- **Find the hidden tone.** Watch the live frequency-domain plot and the waterfall, work out where the tone is, then dial in its offset from the centre frequency and submit your guess.
- **Closer + faster = more points.** You are scored on both accuracy and speed, with a running timer — so a quick, precise guess beats a slow one.
- **A class competition.** Scores feed the shared leaderboard, so the whole class competes to be the best fox hunter.
- **Multiple levels.** Difficulty builds through several stages, keeping the hunt challenging as your spectrum-reading sharpens.



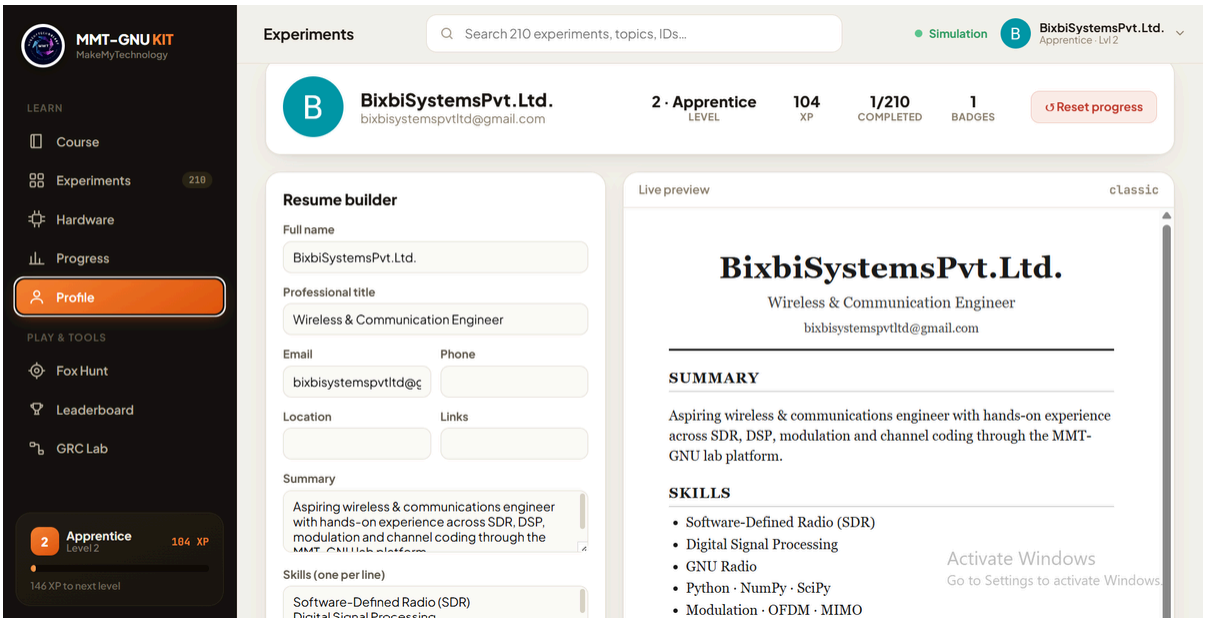
The Spectrum Fox Hunt — read the spectrum and waterfall, dial in the hidden tone's offset, and submit before the clock runs down.

7. Build Your Résumé as You Learn

All that hands-on work is worth showing off. The web app includes a built-in résumé builder that helps you turn what you have done in MMT-GNU into something an employer can read.

What it does

- **Pulls in your progress.** It can auto-fill from your MMT-GNU achievements — the experiments you have completed, the badges you have earned and your challenge scores — so your résumé reflects real, measured RF skills rather than vague claims.
- **Works as a full résumé tool too.** It is also a general-purpose résumé builder, so you can add your education, projects and experience alongside your platform achievements.
- **Multiple templates, live preview.** Choose from several templates and watch your résumé update live as you edit — no guesswork about how the final page will look.
- **Export and send.** Download the finished résumé as a PDF or a Word document, ready to hand to an employer.



The résumé builder — fill in your details on the left and watch a polished résumé, pre-seeded with your MMT-GNU skills, build live on the right.

> **The payoff**
> Because your challenge scores are based on signals you actually produced, a résumé built from them is evidence, not just a list of topics. You finish the course with a portfolio that proves you can do the work.

8. Getting Started

Pick the path that matches what you have. If you are new, start with the simulator — it works on any computer and needs no radio.

Path A — Simulation mode (any computer, fastest start)

Runs on plain Python with no radio and no special stack. From the project folder:

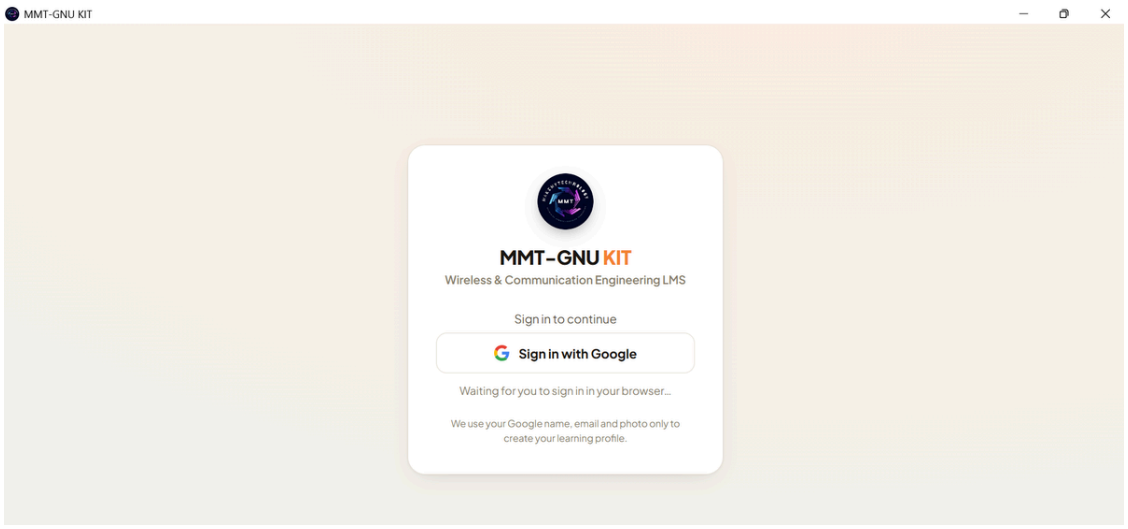
That launches the desktop app in simulation mode. Every experiment is available; results come from the builtin channel simulator.

Path B — Hardware mode (with a real SDR)

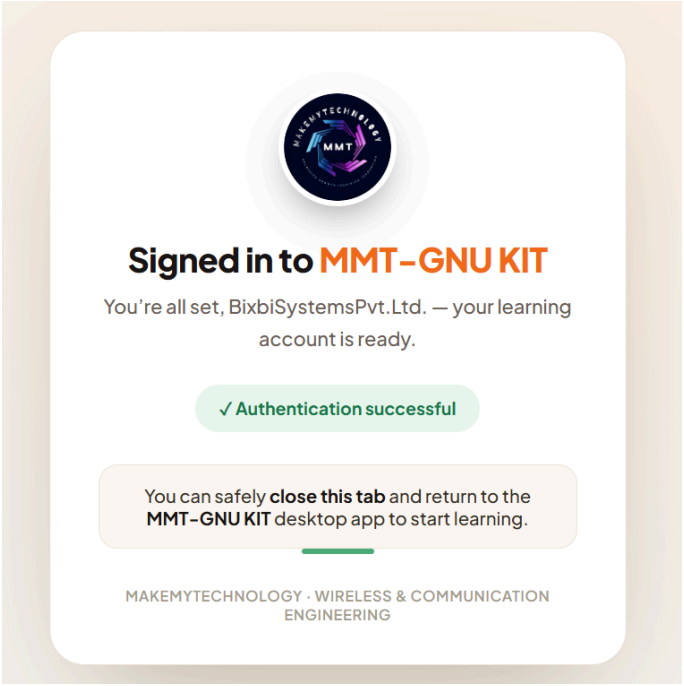
Real-radio support needs the GNU Radio stack (with `gr-iio` / `SoapySDR`), which is installed through conda — not pip. The installer sets up a conda environment with the full SDR stack and a launcher for you.

9. Setting Up Google Sign-In (Optional)

When you open the app you are greeted with a sign-in screen. Tap “Sign in with Google,” approve in your browser, and you are returned to the app with your learning profile ready — your Google name, email and photo are used only to set that up.



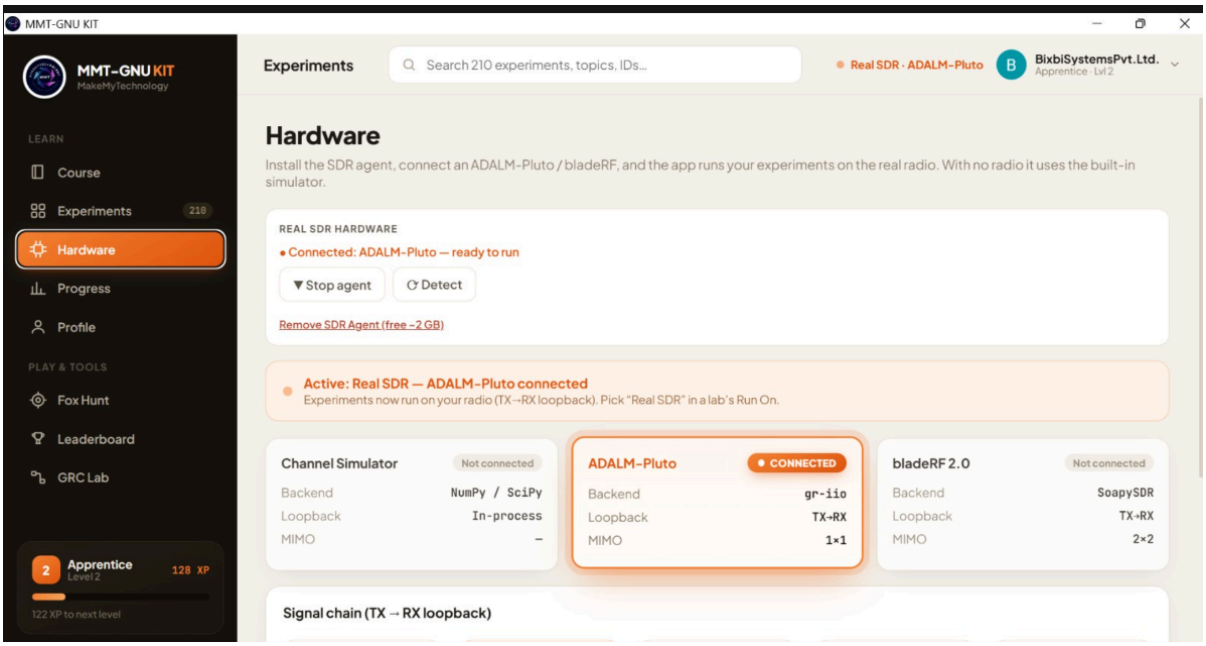
The sign-in screen — one tap with Google, then it waits while you approve in your browser.



Signed in — you can close the browser tab and head back to the app to start learning.

10. Supported SDR Hardware

If you want to run experiments on a real radio, these SDRs are supported out of the box. Install the SDR agent, connect the radio, and the Hardware page shows it live — from there your labs run on the real device (just pick “Real SDR” in a lab’s Run On).



The Hardware page — an ADALM-Pluto connected and ready, with the Channel Simulator and bladeRF 2.0 alongside it.

Device	Backend	Loopback	MIMO
Channel Simulator (no radio)	NumPy / SciPy	In-process	—
ADALM-Pluto	gr-iio	Yes — TX → RX	1×1
bladeRF 2.0	SoapySDR	Yes — TX → RX	2×2

Now pick an experiment, run it, and watch the spectrum come alive. Happy transmitting!

MMT-GNU • MakeMyTechnology • Built on GNU Radio