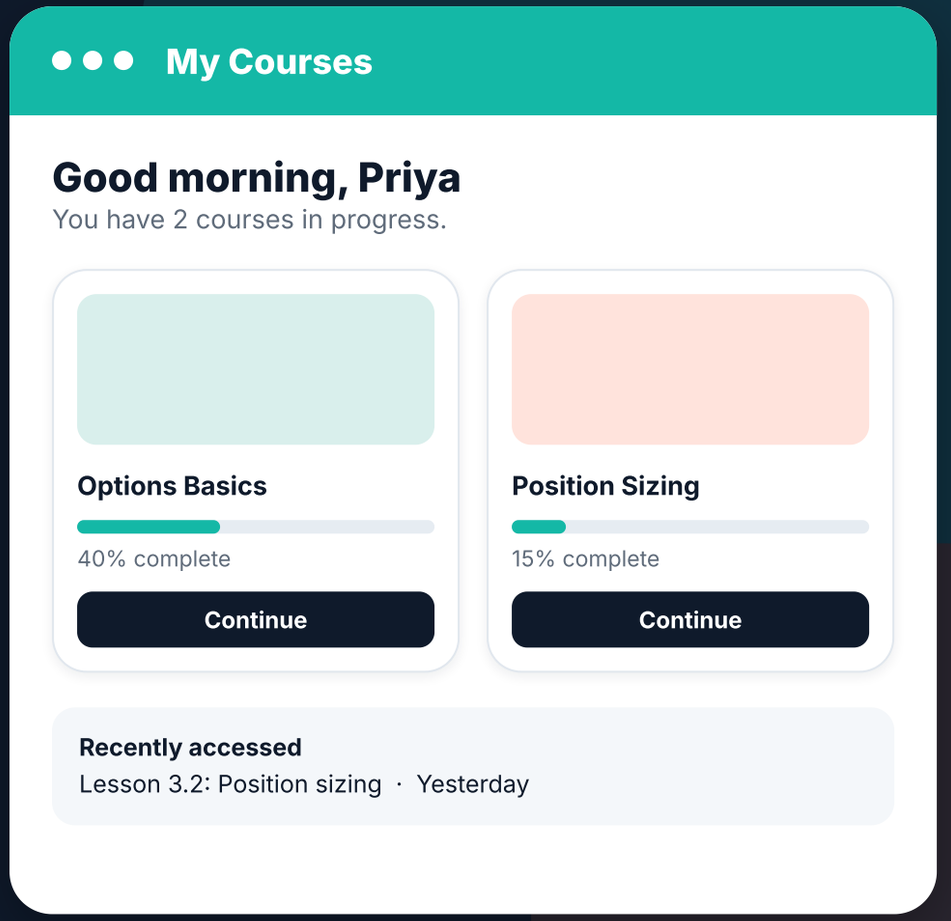
PORTFOLIO CASE STUDY | FULL-STACK WORDPRESS PLUGIN

WP Course Manager: A Course Membership Plugin for WordPress

A WordPress plugin to build courses, enroll students, track progress, and protect paid lessons on any theme.

PHP 8.1 | WordPress | MySQL | REST API | Vanilla JS | Playwright



AI-powered software development | Spec first: ten documents guide an eight-phase build

01 Project Overview

What the platform is, what it is for, and who it helps

PROJECT NAME

WP Course Manager Plugin

A WordPress plugin that lets site owners build courses and deliver them in a clean student area. Students see their courses, track progress, and pick up where they left off. Lessons stay protected for enrolled students only.

MAIN OBJECTIVE

Make three jobs simple: admins build courses, students finish lessons, and everyone sees progress at a glance.

WHO IT HELPS

- **Site owners** — Sell and run courses on their own WordPress site
- **Coaches and educators** — Publish courses and enroll students
- **Students** — Learn on a phone or a desktop
- **Agency developers** — Works with any theme and page builder
- **Course creators** — Admins build now, instructor roles later

3 content types: course, module, lesson

5 custom tables for enrollments and progress

7 REST routes for the player and admin

8 build phases, each tested

02 The Business Pain

The problems with many WordPress course plugins

Dated, crowded pages

Student areas look old and feel busy, so learners lose focus.

Hard to follow

Moving through modules and lessons is confusing.

Progress is hidden

Students often cannot tell how much is left.

Hard to resume

Getting back to the last lesson means searching for it.

Tied to one setup

Many plugins depend on one theme, page builder, or store.

Sound familiar?

When courses are hard to use, students quit and owners lose trust. A calm, theme-proof player with clear progress fixes that.

03 The Proposed Solution

A plugin that guards each lesson and tracks every student

USER ACTION	PLATFORM DOES	RESULT
Admin builds a course	**Saves courses, modules, and lessons** `wcm_course / wcm_module / wcm_lesson`	A clear curriculum with video, text, and audio
Admin enrolls a student	**Creates an enrollment row for an existing user** `POST /wcm/v1/enrollments`	The course shows on the student dashboard
Student opens the dashboard	**Loads enrolled courses with progress bars** `GET /wcm/v1/me/courses`	Course cards and a Continue Learning button
Student opens a lesson	**Checks login, enrollment, and publish status** `AccessService::canAccessLesson()`	The lesson loads, or a clear message shows
Student marks a lesson complete	**Saves one completion record, duplicates blocked** `POST /wcm/v1/progress`	Checkmark and progress bar update
Student comes back later	**Reads the last lesson opened** `wcm_activity`	Back in the right lesson in 2 clicks or fewer

04 Key Features & Capabilities

What the system can actually do

Course builder

Video, text, and audio lessons. YouTube, Vimeo, or self-hosted MP4.

Drag-and-drop curriculum

Reorder modules and lessons, with a keyboard option.

Manual enrollment

Enroll by email, remove, or re-enroll in one row.

Student dashboard

Course cards, progress bars, and recent lessons.

Course player

Sidebar, Previous and Next, and a mobile drawer.

Progress and resume

Percent complete and Continue Learning.

Protected files

Downloads and media check enrollment on every request.

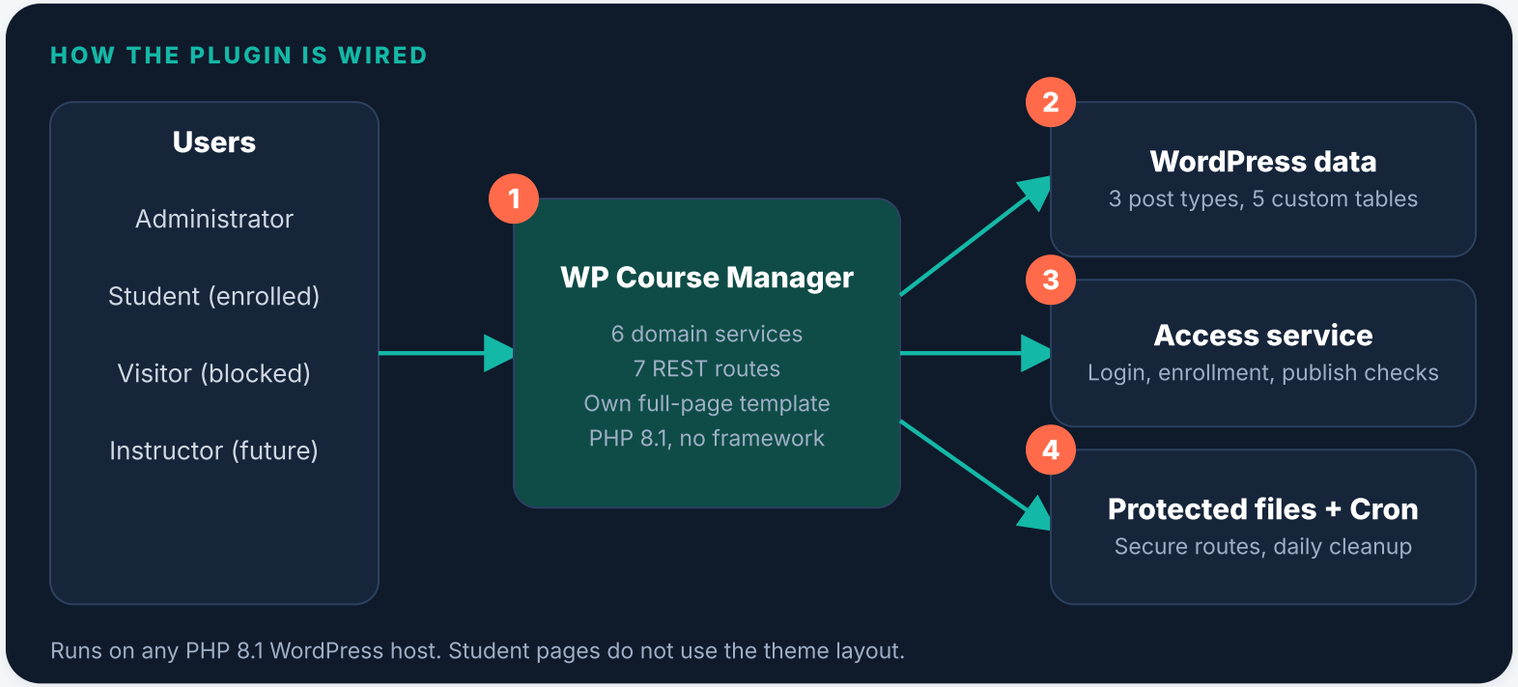
Admin reports

Students per course with percent complete.

Access is checked on the server for every lesson and every download.

05 Workflow Architecture

How the parts of the system connect



Architecture diagram drawn from the Technical Architecture and Database Schema documents.

SIMPLE VIEW

Build course → **Enroll** student → **Access** check → **Learn** lessons → **Save** progress

1 Plugin core

Services for courses, modules, lessons, enrollment, progress, and access. Other code talks to them through interfaces.

2 Data

Courses, modules, and lessons are post types. Enrollments, progress, and activity sit in indexed custom tables.

3 Access service

One method decides who can see a lesson. The player, REST routes, and downloads all use it.

4 Files and jobs

Protected files are served through secure routes. A daily WP-Cron job cleans up old data.

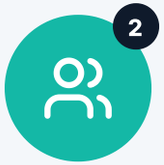
06 How It Works

From a new enrollment to a finished course



Course built

Admin adds modules and lessons.



Student enrolled

Admin adds an existing user.



Access checked

Server confirms enrollment.



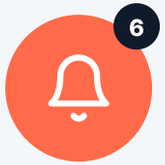
Lesson opens

Video, text, or audio plays.



Mark complete

One record is saved.



Resume or finish

Continue or see the badge.

Example walkthrough (illustrative)

Admin: Enroll Priya in Options Basics for New Traders.

Done. The course now shows on her dashboard.

Priya: Taps Continue Learning on her phone.

Access check passed. Lesson 1.2 opens where she stopped.

Priya: Mark as Complete. The bar now reads 3 of 8 lessons complete.

BEHIND THE SCENES

1. An `active` row is saved in `wcm_enrollments`.
2. `canAccessLesson()` checks login, enrollment, and publish status.
3. `POST /wcm/v1/progress` saves one row. A unique key blocks duplicates.
4. Percent = completed visible lessons / total visible lessons, rounded down.
5. The last lesson opened is saved on each page load.
6. Course completion is saved once and the badge stays.

07 Business Benefits & Expected Outcomes

The value, the time saved, and the operational gains

Easy to follow

Students always see what to learn next.

Progress at a glance

A progress bar and text on every course.

Fast resume

Back in the last lesson in 2 clicks or fewer.

Works on any theme

Student pages use their own scoped styles.

Protected content

Only enrolled students can open lessons and files.

Clear owner view

The Students tab shows percent complete per person.

MVP success targets

10 min	to publish a course
2 clicks	or fewer to resume a course
85+	Lighthouse score on the player
2 sec	player page load target
0	known unauthorized lesson access

Targets from the product requirements, not measured results.

08 Technology Stack & Implementation

The tools, what you need, and how it can grow

TECHNOLOGY STACK

PHP 8.1+	Enums and readonly values, tested up to PHP 8.4
WordPress 6.4+ APIs	Post types, REST, Settings, roles, rewrite, cron
MySQL / MariaDB	Custom tables with prepared $wpdb queries
Vanilla JS + scoped CSS	No React or jQuery for students, no build step
Server-rendered templates	Theme can override files in a wcm/ folder
jQuery UI Sortable	Drag and drop in admin, with keyboard fallback
PHPUnit + Playwright	Unit, integration, and browser tests
PHPCS + PHPStan	WordPress coding standards and static analysis

WHAT YOU NEED

- WordPress 6.4+ site on PHP 8.1+
- MySQL 5.7 or MariaDB 10.4+
- 128 MB PHP memory (256 MB better)
- HTTPS and a working WP-Cron
- Existing WordPress user accounts

CUSTOMIZATION IDEAS

- Quizzes and certificates
- Drip content and course bundles
- WooCommerce payments
- Email notifications
- Instructor roles and CSV export

How it gets built

1 Plan

Ten spec docs: product, flow, data, security, design, and tests.

2 Foundation

Phases 1 and 2: plugin base, tables, and the course builder.

3 Access and learning

Access first, then dashboard, player, and progress.

4 Test and ship

Basic reports, security and speed checks, then a ZIP release.

Have a WordPress project to build?

Start with clear written specs, build in small phases, and test each phase before moving on. That keeps AI-assisted development fast and safe.

Write the spec

Build in phases

Test and launch

Built with PHP, WordPress, MySQL, the REST API, and vanilla JavaScript.