

Domain-Based Student Project Platform – End-to-End Product Plan

1. Problem Statement (Realistic & Market-Driven)

1.1 Current Reality for Students

Students today face **three structural problems** when trying to become job-ready:

1. Project Confusion

2. Students know *domains* (AI, Web Dev, Cybersecurity, etc.) but do not know *what to build*.
3. They rely on random YouTube tutorials, GitHub repos, or copied mini-projects.

4. Lack of Industry-Grade Structure

5. Existing projects rarely include:

- Real-world problem framing
- Case studies
- Clear deliverables
- Time expectations
- Difficulty calibration

6. No Progressive Skill Path

7. Projects are not ordered from *easy* → *medium* → *hard*.
8. Students jump into advanced topics without prerequisites.

Result:

- Weak portfolios
 - Low confidence
 - Poor interview performance
 - Recruiters see repetitive, low-signal projects
-

1.2 Gap in the Market

There is **no single platform** that:

- Starts from **domain selection**
 - Provides **hundreds of structured, difficulty-segregated projects**
 - Treats projects as **mini real-world engagements**, not tutorials
-

2. Solution Overview (What We Are Building)

A **domain-first, project-centric learning platform** where students:

1. Select a **domain**
2. Browse **hundreds of projects** in that domain
3. Choose projects categorized by:
4. Difficulty (Easy / Medium / Hard)
5. Time commitment
6. Skill focus
7. Work on projects that feel like **real industry assignments**

This is **not a course platform**. This is a **project execution and portfolio-building platform**.

3. Supported Domains (Initial Scope)

- AI
- Machine Learning
- Data Science
- Web Development
- Cybersecurity
- Cloud Computing
- IoT
- Embedded Systems
- Finance
- Stock Market & Crypto Trading
- Digital Marketing

Each domain is treated as an **independent vertical** with its own project taxonomy.

4. Core Platform Philosophy

Principle	Meaning
Domain-first	Users start with <i>what they want to become</i>

Principle	Meaning
Project-driven	No passive learning, only execution
Difficulty-aware	Clear separation of Easy / Medium / Hard
Realistic	Projects mirror real industry problems
Outcome-focused	Deliverables > lectures

5. Project Structure (Critical Design)

Each project is a **structured project dossier**, not a blog post.

5.1 Mandatory Sections per Project

1. **Project Title**
2. **Domain & Sub-domain**
3. **Difficulty Level** (Easy / Medium / Hard)
4. **Industry Context / Case Study**
5. Realistic business or technical background
6. **Problem Statement**
7. Clear, bounded, realistic problem
8. **What You Need to Build / Do**
9. Explicit scope
10. Clear success criteria
11. **Prerequisites**
12. Skills, tools, concepts required
13. **Estimated Time Commitment**
14. Minimum time
15. Maximum time (deadline window)
16. **Deliverables**
17. Code
18. Documentation
19. Reports
20. Dashboards
21. Deployments
22. **Optional Extensions (For Advanced Students)**
23. **Evaluation Criteria** (Optional but powerful)

6. User Roles

6.1 Student

- Select domain

- Browse projects
- Filter by difficulty, time, skills
- View full project details

6.2 Admin / Content Team (Internal)

- Create and manage domains
- Add/edit projects
- Tag prerequisites and skills
- Control visibility and difficulty calibration

(Future: Mentors, Recruiters – not in MVP)

7. User Journey (UX Flow)

7.1 Entry Point

- Landing page with:
- Value proposition
- Domain cards

7.2 Domain Selection

- User selects a domain (e.g., AI)
- Lands on **Domain Overview Page**

7.3 Domain Overview Page

Contains:

- Domain description
- Skill map (optional visual)
- Project count by difficulty

7.4 Project Browsing

- Filters:
- Difficulty
- Time commitment
- Sub-domain
- Tools/Tech
- Project cards grid

7.5 Project Detail Page

- Full project dossier
- Clear CTA: *Start Project*

8. UX & UI Design (High-Level)

8.1 Design Principles

- Clean
- Minimal
- Focused on readability
- No clutter

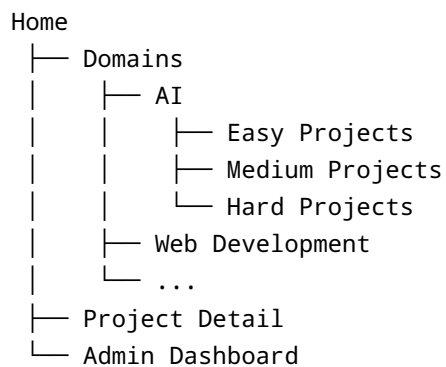
8.2 Key UI Components

- Domain Cards
- Project Cards
- Difficulty Badges
- Time Range Indicators
- Prerequisite Tags
- Structured Content Sections

8.3 Layout

- Left: Filters
 - Center: Project Grid
 - Full-width: Project Detail Pages
-

9. Information Architecture



10. Tech Stack (Recommended)

10.1 Frontend

- Framework: **React / Next.js**

- Styling: **Tailwind CSS**
- State Management: **Zustand / Redux Toolkit**

10.2 Backend

- Runtime: **Node.js**
- Framework: **Express / NestJS**
- API: **REST (initially)**

10.3 Database

- Primary DB: **PostgreSQL**
- ORM: **Prisma**

10.4 Authentication (Optional for MVP)

- Clerk / Auth0 / Firebase Auth

10.5 Hosting

- Frontend: Vercel
 - Backend: Railway / Render
 - Database: Supabase / Neon
-

11. Data Models (Simplified)

Domain

- id
- name
- description

Project

- id
 - domain_id
 - title
 - difficulty
 - case_study
 - problem_statement
 - requirements
 - prerequisites
 - min_time
 - max_time
 - deliverables
-

12. MVP Scope (Strict)

Include

- Domain selection
- Project browsing
- Project detail pages
- Admin project creation

Exclude

- Progress tracking
 - Submissions
 - Certificates
 - Community features
-

13. Future Expansion (Roadmap)

- User accounts & saved projects
 - Project submissions
 - Mentor feedback
 - Recruiter access
 - AI-generated project suggestions
 - Skill gap analysis
-

14. Why This Platform Will Work

- Solves a **real, painful problem**
 - Not another course marketplace
 - High signal for recruiters
 - Strong portfolio alignment
 - Scales across domains
-

15. Final Positioning Statement

A domain-first project platform that turns learners into builders by giving them industry-grade problems, not tutorials.