💻 Project Requirement Document: Student Project & Learning Platform

I. Project Overview & Goal

The goal is to build a high-utility platform that acts as both a curated repository of real-world projects and a personalized workstation for students to manage their learning journey. The platform must foster a community environment where students share, critique, and collaborate on projects.

Inspiration: The platform's overall aesthetic and functionality should draw inspiration from the **AntIntern UI** (for a clean, minimal workspace look) and **Peerlist Launchpad** (for the public project listing and community interaction model).

II. Design Philosophy & UI/UX Requirements

The UI/UX must be the central focus, designed for maximum efficiency and clarity.

Category	Requirement Detail
Aesthetics	**Extremely Professional, Clean, and Minimal.** Avoid excessive styling, shadows, or visual noise. Focus on high information density without clutter.
Color Scheme	**Default: Light Mode.** Must include a clearly accessible **Dark Mode Toggle** (located in the header or profile section).
Workspace Layout	A clear **Workstation** feel: Left Sidebar (Navigation), Center Panel (Content/Inventory), Right Sidebar (Quick Stats/Top Projects).
Typography	Use a clear, highly legible sans-serif font (e.g., Inter, Poppins, or Roboto) optimized for long-form technical content.
Responsiveness	Fully responsive design for desktop, tablet, and mobile viewing.

III. Technical Specification (Suggested Stack)

The freelancer should propose a modern, scalable stack, but the following are recommended:

- **Frontend:** React/Next.js (for performance and structure) with Tailwind CSS (for rapid, minimal UI development).
- **Backend:** Node.js/Express (or Next.js API Routes).
- **Database:** MongoDB or PostgreSQL (Postgres preferred for relational data like users, projects, comments, and complex reporting).

- **Authentication:** NextAuth.js or custom OAuth implementation.
- **File Storage:** AWS S3 or Cloudinary for project screenshots and user profile images.

IV. Detailed Feature Requirements

A. Authentication & User Profile

Feature	Description
Google OAuth Login	Mandatory, primary sign-in method. Quick and secure sign-up/login.
Profile Section	A clean, dedicated page showing: clear Profile Image, User Name, Short Bio, and comprehensive statistics.
Profile Stats	Must display the user's activity metrics: Total Projects Uploaded, Projects Saved, Projects Started (in progress), Projects Completed, Total Likes Received, Total Upvotes Received (on comments and projects).

B. The Project Launchpad (Public Facing)

This is the public browsing area where students discover projects.

Feature	Description
Browsing/Filtering	Primary navigation should be by **Domain** (e.g., Web Development, Data Science, AI/ML, Design, etc.). Must show hundreds of projects.
Project Card	**Minimalist and Informative.** Must include: Project Title, Front Page Screenshot, Like Count, Comment Count.
Project Detail Page	**HIGHLY INFORMATIVE & DENSE.** The core value of the platform. Must be a clean, scrollable view containing the following sections:
	1. Case Study: High-level summary of the context.
	2. Problem Statement: Clear definition of the real-world issue addressed.
	3. Proposed Solution: Detailed explanation of the technical approach.
	4. Prerequisites: Detailed list of required technical skills, software, and libraries.
	5. Deliverables: Clear list of the final outcomes (e.g., fully functional web app, trained model, detailed report).
	6. Initialization Guide (Crucial): A step-by-step, detailed guide on how to clone, set up, configure, and run the project locally. Must be accurate and easily copy-pasteable (e.g., using code blocks).

C. Student Workspace / Dashboard (Private)

This section adopts the **"Workstation"** aesthetic (clean, organized, minimal).

Feature	Description
Sidebar Navigation	Left Sidebar containing links: Dashboard, Launchpad, My Projects (Uploaded), Inventory (My Chosen Projects), Completed Projects, Profile.
Project Inventory	The central area displaying all projects the user has added (like an inventory or cart) from the Launchpad.
Start/Tracking Button	Each project in the Inventory must have a "Start Project" button.
Time Tracking	Upon clicking "Start," the project shows a progress bar. The freelancer needs to implement simple **Elapsed Time Tracking** (not just a countdown). This tracks the time spent on the project (e.g., using a simple session timer that can be paused/resumed or logged). *Initial deadline logic can be simplified to a user-editable "Target Completion Date."*
Completion Status	A "Mark as Done" button. Upon clicking, the project is moved from the Inventory to the **Completed Projects** section.
Right Bar	A small, persistent right sidebar to display dynamic information, such as **Top 3 Trending Projects** from the Launchpad or quick user stats.

D. Project Submission & Community Features

Feature	Description
Project Upload Form	A multi-step form allowing authenticated users to upload their completed projects. The form fields must align with the required content for the Project Detail Page (Title, Screenshot, Case Study, Problem Statement, Solution, Prerequisites, Deliverables, and Initialization Guide).
Launchpad Integration	All uploaded projects must, upon approval (or automatically, depending on scope), become visible in the main **Launchpad** for public browsing.
Liking/Upvoting	Projects and comments must have a clear, functional **Upvote/Like button**. Users can only like/upvote once per item.
Comment Section	Full-fledged comment section on every Project Detail Page, allowing threaded replies and upvoting on individual comments.
Notifications	**Critical:** Any new comment or like/upvote on a project owned by the user must trigger a non-intrusive **Tooltip Notification** in the top-right corner of the screen, similar to modern social media platforms.

V. Deliverables and Project Milestones

The freelancer is expected to provide the following:

1. **Repository Access:** Full access to the codebase (GitHub/GitLab).
2. **Deployment Setup:** Instructions or a working continuous deployment pipeline (e.g., Vercel, Netlify, or self-hosted).
3. **Technical Documentation:** Clear documentation outlining the file structure, component library, and API endpoints.

Suggested Milestone Structure:

Milestone	Deliverable Focus
M1: Foundation	Initial UI/UX scaffolding, full Google OAuth integration, basic Profile Page structure, Light/Dark Mode toggle implemented.
M2: Launchpad Core	Fully functional Launchpad (browsing/filtering), Project Card component, and the complete Project Detail Page (static data acceptable for initial review).
M3: Workspace & Tracking	Full Workspace UI (Sidebar, Center Inventory, Right Bar). Inventory functionality, Start/Pause/Done state management, and basic time tracking implementation.
M4: Community & Submission	Project Upload Form integration, all Community Features (Likes, Comments, Upvotes), and the essential Top-Right Notification System.
M5: Final Polish	Comprehensive testing, bug fixes, final professional UI/UX review (ensuring minimal and clean aesthetic is maintained), and all documentation handover.

VI. Conclusion & Next Steps

The platform requires a developer with a strong eye for **minimalist, professional design** and robust **full-stack engineering** skills. The focus must be on creating a dense, highly functional information portal.

Action for Freelancer: Please review this document and provide:

1. Your proposed technology stack NextJS.
2. An estimated timeline 1 Week.