🎓 Student Practical Work Management System (SPWMS)

🧭 Project Overview

The **Student Practical Work Management System (SPWMS)** is a modern web platform that enables instructors to create, manage, and grade student practical works for software development and graphic design courses.
It provides students with a space to view tasks, submit work (individually or in groups), and receive feedback — all in one intuitive environment.

The platform aims to streamline the process of creating assignments, collecting submissions, grading, and tracking performance through an attractive and efficient digital interface.

🎨 Brand Identity & Color Palette

Primary Colors

Name	HEX	RGB	Description
Academic Blue	`#4AA6E8`	`74, 166, 232`	Represents knowledge, trust, and clarity — used for primary buttons, headers, and highlights.
Energy Red	`#D34B34`	`211, 75, 52`	Symbolizes creativity, energy, and action — used for alerts, key icons, and accent highlights.

Neutral Colors

Name	HEX	RGB	Description
Light Gray	`#F5F6FA`	`245, 246, 250`	For background surfaces and cards.
Dark Charcoal	`#2E2E2E`	`46, 46, 46`	For text and strong contrast.

Typography

- **Primary Font:** Inter (Sans-serif, clean and modern)
- **Secondary Font:** Poppins (for titles and UI labels)

🧱 Technical Stack

Layer	Technology	Description
Frontend	React + TypeScript	Modern, component-based UI with responsive design
Backend	Node.js + Express + TypeScript	RESTful API with clean architecture
Database	MySQL	Relational database for structured academic data
ORM	Prisma	Type-safe database interaction
Authentication	JWT + bcrypt	Secure login and access control
Storage	AWS S3 / Firebase	Cloud-based submission file storage
Styling Framework	TailwindCSS + ShadCN/UI	Modern, accessible, and consistent interface

---

⚙️ Core Features

👩‍🏫 Instructor Module

- Create and publish tasks or projects.
- Assign work to individuals or groups.
- Manage student groups.
- Review submissions and provide grades/comments.

- View class analytics (submission rates, performance trends).

👨‍🎓 Student Module

- View assigned tasks and deadlines.
- Submit files, links, or projects.
- Receive instructor feedback and grades.
- Track personal or group performance history.

🧩 Admin Module (Optional)

- Manage instructors, students, and course data.
- Oversee storage and user activity logs.
- Generate performance reports.

📊 Database Design (Overview)

Table	Description
Users	Manages login credentials and user roles (student, instructor, admin).
Courses	Contains course details and assigned instructors.
Assignments	Stores assignment details, deadlines, and type.
Groups	Handles group membership and related assignments.
Submissions	Tracks submitted files, links, or projects.
Grades	Records instructor evaluations, scores, and feedback.

🧩 Architecture Overview

Frontend (React + TS)
↓ REST API
Backend (Node.js + Express + Prisma)
↓
Database (MySQL)
↓
Cloud Storage (AWS/Firebase)

🧰 Development Roadmap

Phase 1 — Setup

- Configure Node.js + Express + Prisma.
- Set up React + Tailwind + ShadCN UI.
- Implement authentication system (JWT).

Phase 2 — Core Modules

- Instructor task management.
- Student submission interface.
- File upload integration.

Phase 3 — Evaluation & Feedback

- Grade submission workflows.
- Student feedback display.
- Analytics dashboard.

Phase 4 — Enhancements

- Group collaboration module.
- Notifications via email or in-app alerts.
- Role-based permissions.

Phase 5 — Deployment & Maintenance

- Testing (unit, integration, and UX).
- Deployment to Vercel (frontend) and Render/EC2 (backend).
- Continuous integration & backup setup.

🎨 UI Design Guidelines

- **Primary Color Usage:** Blue for main interface and call-to-action buttons.
- **Accent Color Usage:** Red for alerts, notifications, and highlights.
- **Consistent Spacing:** Use Tailwind utilities for padding/margin consistency.
- **Component Library:** ShadCN/UI components with custom color tokens based on palette.
- **Accessibility:** High-contrast color scheme for readability.

📈 Future Enhancements

- AI-assisted grading for text or code submissions.
- Offline submission syncing.
- Chat or feedback threads within each assignment.
- Integration with Moodle, Google Classroom, or similar LMS platforms.