

# AI Prompts for Collaborative Web Platform (Fullstack)

## 1. Project Boilerplate Setup

Create a full-stack MERN boilerplate with React frontend and Node.js + Express backend, connected to MongoDB. Include folder structure for scalable development. Add CORS, dotenv, and proxy setup.

## 2. Auth + Team System

Implement user authentication using JWT in MERN stack. Support Google OAuth. Users should be able to register, login, create/join teams, and assign a team leader. Store user and team data in MongoDB.

## 3. Live Coding Battle (Monaco Editor + Socket.io)

Build a real-time collaborative coding interface using Monaco Editor and Socket.io. Two teams should be able to join a battle room, edit code collaboratively, and see live updates.

### Timer Integration

Add a synchronized countdown timer (e.g., 15 min) to the coding battle room using Socket.io. The timer should start when both teams are ready.

## 4. Code Submission + Evaluation

Integrate code evaluation using Judge0 API. After teams submit their code, send it to Judge0, get output, and compare it with expected result. Save score and accuracy in database.

## 5. Video/Audio Chat with WebRTC

Add video and audio chat functionality between team members in a room using WebRTC and Socket.io for signaling.

pop

### Using SDK

Integrate 100ms/Agora video calling in a React app for real-time team communication inside a battle

room. Each room should have its own session.

## **6. Leaderboard (Room/Global)**

Create a leaderboard showing top teams based on win count, average accuracy, and fastest completion time. Support both global and room-specific leaderboards using MongoDB aggregation.

## **7. AI Assistant - Prompt Battles & Code Quizzes**

Use Gemini/OpenAI API to generate code-based quizzes with the following types:

1. Predict output
2. Fix the bug
3. MCQs on theory
4. Fill in the blanks

Return difficulty level, correct answer, explanation, and feedback.

## **Evaluation + Scoring**

Evaluate a user's quiz answers (MCQ, fix bug, output prediction) and return a score based on time and accuracy. Provide detailed AI-generated feedback.

## **8. AI Interactive Learning**

Design an AI-based learning module that recommends coding topics to students based on their past performance and difficulty level. Support chat-like tutoring on selected topics (e.g., recursion, async JS).

## **9. Folder Structure**

Suggest a professional MERN project folder structure with support for:

- Modular routes/controllers
- Socket.io integration
- AI helper functions
- Separate folder for battles, quizzes, leaderboard, and auth.

## 10. Deployment

Write deployment guide for:

- React frontend to Vercel
- Node.js/Express backend to Render
- MongoDB to Atlas

Also include environment variable setup and CORS configuration.

### **Bonus: AI-Driven Feedback Tool**

Implement a feature where AI gives feedback on user's submitted code: correctness, performance, and suggested improvements. Use Gemini/OpenAI API for analysis.