Code Buddy

Overview

This document outlines the requirements for a platform that allows users to:

1. Run single-file code and analyze outputs, including error detection and AI-generated solutions.
2. Store and manage code with descriptive metadata for easy search and retrieval.
3. Convert image-based code to text for execution.
4. Generate AI-driven personalized learning schedules.

The software will include both free and premium features, with a comprehensive admin dashboard for management.

Functional Requirements

1. Code Execution and Error Analysis

User Features:

- **Input Code:** Users can write single-file code in supported languages.
- **Run Code:** Execute the written code.
- **Error Detection:** If the output contains errors, AI integration (OpenAI) will:
    - Analyze the error.
    - Provide potential causes.
    - Suggest solutions.

Backend Requirements:

- API to handle code execution.
- Integration with OpenAI for error analysis.

2. Code Storage and Management

User Features:

- **Store Code:** Save code snippets with:
    - Title.
    - Description.
    - Tags.
- **Search and Retrieve:** Search for stored code using keywords or tags.

Backend Requirements:

- MongoDB database schema to store code with metadata.
- APIs to:
    - Save code.
    - Retrieve code.
    - Update/delete code entries.

3. Image to Code Conversion (Premium Feature)

User Features:

- **Upload Image:** Accept images containing code blocks.
- **Convert to Text:** Use AI to:
    - Extract text from the image.
    - Format the extracted text as code.
- **Run Extracted Code:** Execute the generated code directly.

Backend Requirements:

- Integration with OCR tools for text extraction.
- APIs to:
    - Process images.
    - Convert text to code.
    - Execute the code.

4. AI-Generated Learning Schedules (Premium Feature)

User Features:

- **Input Learning Goals:** Users specify:
    - Topics to learn.
    - Desired timeframe.
- **Generate Schedule:** AI provides:
    - A learning path.
    - A calendar with daily tasks.

Backend Requirements:

- Integration with OpenAI for generating schedules.
- APIs to:
    - Accept user inputs.
    - Generate and return the learning schedule.

5. Admin Dashboard

Admin Features:

- **User Management:**
  - View, edit, or delete user accounts.
  - Monitor user activity.
- **Content Management:**
  - Manage stored code entries.
- **System Operations:**
  - Handle premium subscriptions.

Non-Functional Requirements

1. Scalability

- Support multiple simultaneous code executions and API calls.

2. Security

- User authentication and authorization using JWT.
- Secure storage of user data and premium subscription details.

3. Performance

- Fast response times for code execution and AI-based analysis.
- Efficient handling of large code storage and image processing.

4. Usability

- Intuitive user interface designed with React.
- Clear documentation for users and admins.

Technology Stack

Backend:

- Node.js with Express.js for API development.
- Mongoose for MongoDB database integration.
- OpenAI APIs for AI functionalities.

Frontend:

- React for the user interface.

Database:

- MongoDB for data storage.

Premium Features:

- AamarPay payment integration for subscription management.

Business Model

Premium Features:

1. **Image to Code Conversion.**
2. **Learning Schedule Generation.**

Revenue Streams:

- Subscription plans for accessing premium features.

Conclusion

This platform aims to simplify coding, learning, and problem-solving processes using cutting-edge AI technologies while offering a seamless user experience and scalable architecture.