Sustainable Smart City Assistant Using IBM Granite LLM

Documentation format

1. Introduction

- **Project Title: EDUTUTOR AI**
- **Team Members: :** K.NAVEEN

  GOLLA NAVVEN

  C SIDDU

  E POOVENDRAN

- 2. Project Overview

- **Purpose:**. The purpose of EduTutor AI, or any AI-powered intelligent tutoring system, is to enhance and personalize the learning experience by providing individualized support and guidance to students. It acts as a virtual tutor, adapting to each student's pace, learning style, and knowledge level to make learning more effective and engaging.
- **Features: Adaptive Learning:**
  EduTutor AI analyzes student performance data (e.g., quiz scores, time spent on tasks) to adjust the difficulty level of lessons in real-time, ensuring each student learns at their optimal pace.

  - **Personalized Learning Paths:**

- AI algorithms create customized learning plans based on individual learning styles and needs, focusing on areas where students struggle.

  - **Content Recommendations:**

- The system recommends relevant learning materials and resources based on student performance and learning preferences.

- 2. Enhanced Learning Experience:

  - **Immediate Feedback:**

- Students receive instant feedback on their work, allowing them to identify and correct mistakes quickly.

3. Architecture

- **Frontend:** Describe the frontend architecture using React.
- **Backend:** Outline the backend architecture using Node.js and Express.js.
- **Database:** Detail the database schema and interactions with MongoDB.

4. Setup Instructions

- **Prerequisites:** List software dependencies (e.g., Node.js, MongoDB).
- **Installation:** Step-by-step guide to clone, install dependencies, and set up the environment variables.

5. Folder Structure

- **Client:** Describe the structure of the React frontend.
- **Server:** Explain the organization of the Node.js backend.

6. Running the Application

- Provide commands to start the frontend and backend servers locally.
    - **Frontend:** `npm start` in the client directory. ○
      Backend: `npm start` in the server directory.

7. API Documentation

- Document all endpoints exposed by the backend.
- Include request methods, parameters, and example responses.

8. Authentication

- Explain how authentication and authorization are handled in the project.
- Include details about tokens, sessions, or any other methods used.

9. User Interface

- Provide screenshots or GIFs showcasing different UI features.

10. Testing

- Describe the testing strategy and tools used.

11. Screenshots or Demo

- Provide screenshots or a link to a demo to showcase the application.

12. Known Issues

- Document any known bugs or issues that users or developers should be aware of.

13. Future Enhancements

- Outline potential future features or improvements that could be made to the project.