Project Overview

This project aims to analyze the quiz performance of students and provide them with personalized recommendations to improve their preparation for the NEET exam. The solution leverages Python to process current and historical quiz data, analyze performance, and generate actionable insights.

Project Structure

This project is divided into the following steps:

1. **Step 1: Data Setup**
   This step involves loading the current and historical quiz data from JSON files for further analysis.

2. **Step 2: Data Analysis**
   In this step, the performance data is analyzed to identify patterns related to topics, difficulty levels, and accuracy.

3. **Step 3: Recommendations**
   Based on the analysis, personalized recommendations are generated to help students improve their weak areas.

4. **Step 4: Data Visualization**
   The analysis results are visualized to help students understand their performance and improvement trends better.

5. **Step 5: Student Persona and Insights**
   This step generates a student persona based on the performance patterns, along with insights into their strengths and weaknesses.

Setup Instructions

1. **Clone the repository**
   Clone the repository to your local machine using the following command:
   bash
   CopyEdit

   ```
   git clone <repository_url>
   ```

2.

3. **Install dependencies**
   Ensure you have Python 3.x installed. Install the required dependencies:
   bash
   CopyEdit

```
pip install -r requirements.txt
```

4.

5. **Run the main script**
   After installing the dependencies, run the main script:
   bash
   CopyEdit

   ```
   python main.py
   ```

6.

Approach

1. **Data Loading (Step 1)**:
   The `load_data()` function loads the current and historical quiz data. This data is then used for analysis.

2. **Data Analysis (Step 2)**:
   The `analyze_data()` function processes the data and identifies trends and patterns in student performance.

3. **Recommendation Generation (Step 3)**:
   Based on the analysis, the `generate_recommendations()` function provides personalized recommendations to the student.

4. **Visualization (Step 4)**:
   The `visualize_data()` function generates visualizations (like charts or graphs) to help the student understand their performance.

5. **Persona and Insights (Step 5)**:
   The `generate_persona()` function creates a student persona by analyzing their strengths and weaknesses based on historical data.

Example Output

Once the script is executed, it will output the following:

1. **Student Persona and Insights**:
   A detailed description of the student's performance trends, including strengths and weaknesses.

2. **Recommendations**:
   Actionable recommendations such as suggested topics, difficulty levels, or question types to focus on for improvement.