**Technical Report: MONTHLY REFLECTION PROJECT**

**Capstone Project**

**Author: Odion Joel Okonofua  
ID: FE/23/88847286**

**Overview**

The code combines two major functionalities:

1. **Student Progress Tracker**:
   * Allows users to track their progress in a course.
   * Provides insights into completed modules, strengths, weaknesses, and performance evaluation.
   * Visualizes the time spent on each module compared to expected time using a bar chart.
2. **Job Interview Preparation Assistant**:
   * Offers course-specific interview questions.
   * Evaluates user responses against key concepts and provides feedback.
   * Supplies additional interview tips for better preparation.

**1. Student Progress Tracker**

**Modules and Functions**

**1.1. Collect User Input**

**Function:** get\_student\_data()

* **Purpose**: Collects information about a student's course progress, including total modules, completed modules, and time spent per module.
* **Steps**:
  1. Accepts the student's name, course name, total modules, and completed modules.
  2. Gathers time spent on each completed module.
  3. Returns all inputs for further processing.
* **Input/Output**:
  1. Input: User inputs via input() prompts.
  2. Output: A tuple containing student details and module data.

**1.2. Calculate Insights**

**Function:** calculate\_insights(total\_modules, completed\_modules, time\_spent, avg\_time\_per\_module=5)

* **Purpose**: Analyzes the data to compute:
  1. Progress percentage.
  2. Average time spent per module.
  3. Lists of strengths (modules with adequate time spent) and weaknesses (modules with inadequate time).
  4. Determines if the student passes the course requirements.
* **Steps**:
  1. Compute progress as a percentage of total modules.
  2. Calculate average time spent per completed module.
  3. Compare time spent per module against the expected average (avg\_time\_per\_module).
  4. Use criteria to determine if the student has passed.

**1.3. Display Insights**

**Function:** display\_insights(student\_name, course\_name, progress\_percentage, avg\_time\_actual, strengths, weaknesses, passed)

* **Purpose**: Provides a detailed textual summary of:
  1. Course progress.
  2. Strengths and weaknesses.
  3. Pass/fail status.
* **Output**:
  1. Prints the summary report to the console.

**1.4. Visualize Insights**

**Function:** visualize\_insights(time\_spent, avg\_time\_per\_module)

* **Purpose**: Creates a bar chart comparing actual vs. expected time spent on modules.
* **Steps**:
  1. Use matplotlib to plot actual time spent and expected time.
  2. Add annotations, gridlines, and styling for readability.
  3. Display the chart.

**1.5. Main Function**

**Function:** main()

* **Purpose**: Coordinates all the above functions for a seamless user experience.
* **Steps**:
  1. Collect user input.
  2. Analyze and calculate insights.
  3. Display the summary report.
  4. Visualize the progress.

**2. Job Interview Preparation Assistant**

**Modules and Functions**

**2.1. Interview Questions**

**Dictionary:** course\_interviews

* **Purpose**: Maps various courses to a list of sample interview questions.
* **Content**: Courses such as AI/ML, Animation, Cloud Computing, etc., each with 3 predefined questions.

**2.2. Evaluate Answer**

**Function:** evaluate\_answer(answer, question)

* **Purpose**: Evaluates user-provided answers based on relevant keywords for the selected course.
* **Steps**:
  1. Match keywords from a predefined list with the user's answer.
  2. Count the number of keyword matches to determine the quality of the response.
  3. Return feedback: "Great job", "Decent", or "Needs improvement".

**2.3. Interview Preparation**

**Function:** interview\_preparation()

* **Purpose**: Guides the user through a mock interview process.
* **Steps**:
  1. Display a list of courses for selection.
  2. Randomly select a question from the chosen course.
  3. Collect the user's answer and evaluate it.
  4. Provide feedback and additional interview tips.

**2.4. Main Function**

**Function:** \_\_main\_\_

* **Purpose**: Ensures that the correct module (Student Progress Tracker or Interview Assistant) runs independently.

**Key Features**

1. **Student Progress Tracker**:
   * User-friendly prompts to gather data.
   * Detailed analysis and insights on module performance.
   * Visual representation of progress for easy understanding.
2. **Interview Preparation Assistant**:
   * Course-specific questions ensure relevance.
   * Automated feedback for improvement.
   * Useful tips to ace interviews.

**Dependencies**

* **Python Libraries**:
  1. matplotlib: For visualizing progress in the tracker.
  2. numpy: For handling data-related tasks.
  3. random: For random question selection in the interview assistant.

**Execution**

To run either module:

1. Ensure the script is saved as a .py file.
2. Use the terminal or an IDE to execute it.
   * For Student Progress Tracker: Ensure to follow all prompts and input accurate data.
   * For Interview Assistant: Select a course, answer questions, and review feedback.

**Potential Improvements**

1. **Enhanced Error Handling**:
   * Validate numeric inputs and handle invalid data gracefully.
2. **Expanded Visualization**:
   * Add line graphs or pie charts for deeper insights.
3. **Customization**:
   * Allow users to define their average time per module or add new courses dynamically.