Overview:

Your task is to design and build an API for a **Health Tracker Application**. The application will allow users to track and analyze their health data, including **Physical Activity**, **Sleep Activity**, and **Blood Tests** (e.g., steps, sleep duration, and glucose levels). Your goal is to implement a database schema, build a RESTful API, and create an aggregation endpoint to calculate a user's health score.

This assignment is divided into two parts: **Practical Tasks** and **Theoretical Questions**.

Practical Tasks:

1. Design the Database Schema

- Design a **relational database schema** for the health tracker use case.
- Include tables for storing user information and health data for **Physical Activity**, **Sleep Activity**, and **Blood Tests**.

Build a RESTful API

- Use **Flask** or **FastAPI** for API development.
- Use **SQLAlchemy** for database interaction.
- Implement the following functionalities:
  - Basic **CRUD operations** (Create, Read, Update, Delete) for users and health data.
  - An API endpoint, `/get_health_score`, that:
    - Aggregates health data for a user.
    - Calculates a **health score** based on comparisons to other users in the system. You can define your own formula for the health score based on available data.

Example: The health score might combine factors like steps, sleep duration, and glucose levels, comparing them to average values across the system.

Theoretical Questions:

1. Deployment on AWS

- Explain how you would deploy the above application on AWS.

2. Scaling & Troubleshooting

Imagine the health tracker application has become wildly popular, gaining thousands of new users every day. However, users start reporting the following issues:

- Health scores are inaccurate.
- API responses are delayed.
- The application occasionally crashes under load.

Questions:

- How would you approach diagnosing and solving this problem?
- How would you design a long-term plan to make the system resilient to future scalability challenges?