Smart Health Monitoring System

# Description:

The Smart Health Monitoring System is a Java-based project designed to help users maintain a healthy lifestyle by tracking their daily activities, vital signs, and medical records. The project uses a database to store user data, allowing for the analysis and visualization of personal health trends over time. The Monitoring System will help users keep track of their health data and stay on top of their wellness goals while providing healthcare professionals with valuable insights into their patients' overall health.

# Instructions:

For this project, the skeleton is given to you. Please complete the implementation of all the necessary classes and methods. You may need to restructure some methods and classes and import some external libraries. Please feel free to edit the code any how you want. We want you to be creative in your ideas so add more functionality to make this a complete app.

# Features:

* User registration and login: Users can create accounts and securely log in to the system.
* Health data input: Users can input their daily activities, vital signs, and medical records.
* Data storage: It is your choice whether you want to store the data in a database or in files to disk, I’ve already setup the basic connection for postgres with an example UserDaoExample class to show how to make the queries.
* Personalized recommendations: Based on the user's health data, the system generates personalized health recommendations to help users maintain a healthy lifestyle.
* Medicine reminder: Users can set reminders for taking medications, and the system will notify them when it's time to take their medicine.
* Doctor portal: A separate portal for doctors to access their patients' data, enabling them to monitor patients' health and provide better care.
* Every compile time the code must connect to database, read and load the record from the tables. Finally, must write\update new data into the tables.

# Database Layout:

* User information: Tables for storing user data, including personal details, login credentials, and access permissions.
* Health data: Tables for storing daily activities, vital signs, and medical records.
* Recommendations: Tables for storing personalized health recommendations generated by the system.
* Medicine reminders: Tables for storing medication details and reminder schedules.
* Doctor-patient relationship: Tables for managing the relationships between doctors and their patients.

# Deliverables: [100 Marks]

## Code files: [50 marks]

Design and implement the app using all the requirements mentioned above. A complete running code(java classes) files for everything is required.

## Documentation: [ 20 marks]

You need to document your java project above.

1. User Documentation This include a document stating what the application is about, explanation of all the classes and their working, and how to start it/access it. Also include the class diagram with the associations between them. [10]
2. Development Documentation This includes at least the Javadocs, a description of the source code directory structure, the build process (i.e., how to compile the project), compiler time dependencies, development standards, how to set up a database for development, and how to get the source code from the repository. [5]
3. Deployment Documentation This is basically the installation manual of the application, describing any steps needed to make it run. [5]

## To demonstrate the project [30 marks]

1. Complete HealthMonitoringApp.java class to test the health monitoring system app and all its features. Add appropriate data records (database) to demonstrate the working. (Add minimum of 10 records). [20]
2. Record a video to demonstrate\explain\show\present the working of your health monitoring system app. Video must explain all the classes, code, functionality and finally the working (output) of the app. [10] min 4 minutes - maximum 8 minutes.

Note: Similar code logic (code with the same variable names/ layout), documents, or Chatgpt generated code will be graded as 0. Moreover, everyone’s database must be unique i.e (record inserted into the database must be different).

**Good Luck**