Personal Health Tracker Application – Documentation

Overview  
The Personal Health Tracker Application allows users to log their health metrics (e.g., weight, steps, calories), set health-related goals, and receive notifications on goal achievements or reminders. The application is built using Java 17, Spring Boot, PostgreSQL, and is containerized using Docker.  
  
Key Features:  
- User Authentication: Secure login and registration using Spring Security.  
- Health Metrics Logging: Users can log, view, and edit health metrics.  
- Goal Setting: Users can set health goals and track their progress.  
- Progress Reports: Weekly and monthly reports on health progress.  
- Notifications: Users receive alerts when they achieve goals or reminders if they fall behind.  
  
Project Structure  
  
The project is structured as follows:  
src/main/java/

* com.example.Personal.Health.Tracker/
* Config/ # Configuration files
* Controllers/ # API Controllers
* Dto/ # Data Transfer Objects
* Entitiy/ # JPA Entity classes
* Enum/ #Enums
* Exception/ #Exception Class
* Repository/ # Data repositories
* Services/ # Service classes for business logic
* Utils/ #Utilities Class

PersonalHealthTrackerApplication.java # Main application entry point  
  
src/main/resources/  
 application.properties # Configuration properties

Setup Instructions  
  
Prerequisites  
- Java 17or later  
- Maven (for managing dependencies)  
- Docker(for containerization)  
- PostgreSQL (as the database)  
  
1. Clone the Repository  
Clone the project from the version control system (e.g., GitHub):  
  
git clone <https://github.com/Madhumal-Thushan/Personal-Health-Tracker>

cd health-tracker  
  
2. Database Setup  
  
**Option A:** Using Docker for PostgreSQL  
If you prefer to run PostgreSQL in a Docker container, use the following command:

docker-compose up --build  
  
3. Configuration  
  
Configure the database and email service by updating the `application.properties` file located in `src/main/resources/`.

Here is an example configuration for PostgreSQL and Gmail SMTP.  
  
Database configuration  
spring.datasource.url=jdbc:postgresql://localhost:5432/personal\_health\_tracker\_db  
spring.datasource.username=<your\_user\_name>  
spring.datasource.password=<your\_user\_password>  
spring.datasource.driver-class-name=org.postgresql.Driver  
  
JPA/Hibernate configurations  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect  
  
  
#Email Configuration  
spring.mail.host=sandbox.smtp.mailtrap.io  
spring.mail.port=587  
spring.mail.username=452dd6721db5df   
spring.mail.password=fcc763c7210940  
spring.mail.properties.mail.smtp.auth=PLAIN  
spring.mail.properties.mail.smtp.starttls.enable=STARTTLS

4. Build and Run the Application

**Option A**: Running without Docker  
1. Build the project with Maven:  
mvn clean install  
  
2. Run the Spring Boot application:  
mvn spring-boot:run  
  
**Option B**: Running with Docker  
1. Build the Docker image:  
docker build -t personal-health-tracker-app .

2.Build Database Imsage :

docker-compose up –build

5. Access the Application

Once the application is running, you can access it at:  
`http://localhost:8080`  
  
  
API Documentation  
used Swagger in this application. Once Application starts you can access Swagger API Doc at :

<http://localhost:8080/swagger-ui/index.html#/>

Notification Service  
The system sends two types of notifications:

1. Goal Achievement Notification:

When the user reaches the target value for a set goal.

2. Goal Reminder:

When the user is behind on their progress based on the timeline of the goal.  
  
Configuration for Email Notifications  
- The application uses Spring Boot’s `JavaMailSender` for sending emails.  
- To test the notification service, use tools like Mailtrap

#Email Configuration  
spring.mail.host=sandbox.smtp.mailtrap.io  
spring.mail.port=587  
spring.mail.username=<YourUserName>  
spring.mail.password=<YourPassword>  
spring.mail.properties.mail.smtp.auth=PLAIN  
spring.mail.properties.mail.smtp.starttls.enable=STARTTLS

Testing  
  
Unit Tests  
Run unit tests with Maven:  
mvn test  
  
Conclusion  
  
This Personal Health Tracker Application allows users to set and track their health goals, log health metrics, and receive notifications. The system is containerized for ease of deployment and can be scaled as needed with a relational database like PostgreSQL.