**Wellness Tracker Application**

**Assumption**

1. Background

A state-of-the-art wearable device has been developed and deployed on users to capture key health metrics, including daily hydration levels, sleep duration, physical activity, and meditation time. To fully leverage this rich stream of health data, a companion application has been designed to empower data scientists and health professionals with tools to record, manage, and analyze user data effectively.

The application supports both **manual data entry** for precise individual tracking and **bulk data import** from the wearable device to ensure seamless synchronization. This dual-mode input system ensures flexibility and scalability in data management.

Beyond data collection, the app enables users to **query historical health records** within a defined date range and to **generate daily or weekly aggregated summaries**, offering valuable insights into behavioral trends and overall well-being. These features provide the analytical foundation for delivering personalized health assessments and actionable wellness recommendations.

This solution bridges wearable technology and data science, supporting informed decision-making for improved user health outcomes.

1. Wearable Health Data Integration
   1. Data Collection

* Users wear the device.
* The devices automatically track:

1. Water Intake (hydration)
2. Sleep Duration
3. Exercise Minutes
4. Meditation Time
   1. Data Transfer to APP

* Option A: Manual input by data scientists via app interface
* Option B: Automatic syncing from wearable via API (“/sync-device”)
  1. Data Storage
* All data is stored in the centralized server-side database that is managed by using MySQL.
* Each record includes: “user\_id”, “date”, “hydration\_liters”, “sleep\_hours”, “exercise\_minutes”, “mediation\_minutes”
  1. Data Access and Visualization
* Querying historical data by date range
* Summarizing data into

1. Daily summaries
2. Weekly summaries

* Generating reports to assess user health trends
  1. Health Insights and Actions
* Use summaries to

1. Evaluate user wellness
2. Recommend personalized improvements
3. Identify patterns and areas of concern

**Use Guideline**

The interface of the app is shown in the figure below,

图形用户界面

AI 生成的内容可能不正确。

1. Input the record into database manually

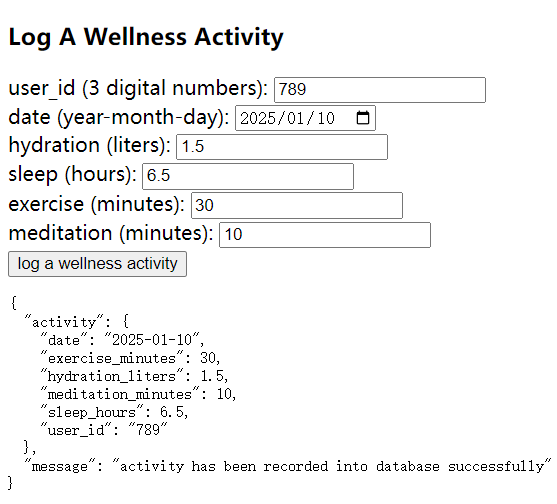
For example, you should input:

{user id: “789”, “date”: “2025-01-10”, “hydration\_liters”: 1.5, “sleep\_hours”: 6.5, “exercise\_minutes”: 30, “meditation\_minutes”: 10}.

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

When finishing data input, click the button “log a wellness activity”, we can see the



1. Sync the wearable device data into database

Assume that we have the following records:

[

{"user\_id": "478", "date": "2025-1-10", "hydration\_liters": 1.5, "sleep\_hours": 8, "exercise\_minutes": 20,"meditation\_minutes": 10},

{"user\_id": "479", "date": "2025-1-11", "hydration\_liters": 2.5, "sleep\_hours": 7, "exercise\_minutes": 30,"meditation\_minutes": 20},

{"user\_id": "479", "date": "2025-2-11", "hydration\_liters": 2.4, "sleep\_hours": 6, "exercise\_minutes": 40,"meditation\_minutes": 30}

]

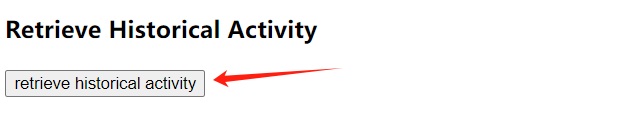
Assume that these records are written on a document ending with “.py” in “WearableDeviceData” folder, by clicking the button “sync wearable device data”, we can see the following results,

图形用户界面, 文本

AI 生成的内容可能不正确。

1. Retrieve historical activity

We assume that A user (id 479) is interested in retrieving and analyzing their logged wellness activities for the period from January 1st to February 28th, 2025, to better understand their health behavior during that timeframe. We should click the button “retrieve historical activity”, as shown in the following picture.



After clicking, we can see the new page , as shown in the following picture,

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

Input the above the information, as shown in the following,

文本

AI 生成的内容可能不正确。

Finally, click the button “retrieval”, we can see the result like this,

图形用户界面, 应用程序, 表格

AI 生成的内容可能不正确。

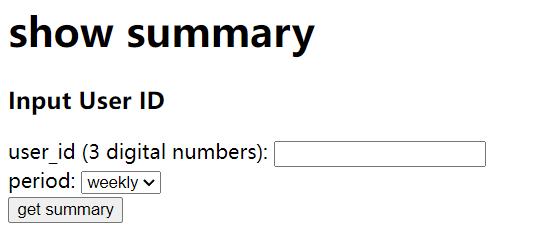
1. Generate the daily or weekly summary.

Since adopting the wearable device, the user (id 479) seeks to evaluate their average daily and weekly wellness metrics to better understand and track their overall health patterns. To this end, we can click the button “show summary”( beforehand we should return the previous home page by using the browser return function)

图形用户界面, 应用程序

AI 生成的内容可能不正确。

After that, a new page is shown as following picture,



Input “479” into the box of “user id” and choose “daily” in period box, then click the button “get summary”,

文本, 信件

AI 生成的内容可能不正确。

Choose “weekly” in period box then click the button “get summary”,

文本

AI 生成的内容可能不正确。

**Explanation and Discussion**

The wellness tracker application is developed by using Python and the backend framework Flask. It consists of a front end (html documents), backend (py documents), and database (MySQL). The structural architecture of the system is illustrated in the diagram below (the arrows denote data flows),

图示, 示意图

AI 生成的内容可能不正确。

The user sends the request to the backend server via frontend. Once the backend receives the request, it will extract the corresponding data from the database, forming the response that will be sent to the front end. Or it will send the collected data to the database and return a message to the front end.

The current system is functional, but it is not without limitations. For instance, repeatedly clicking the button to sync wearable device data can result in a large volume of duplicate records being stored in the database, leading to inefficient use of storage resources. Additionally, there is no input validation in place — users can enter unrealistic values such as 1000 liters of daily water intake or 48 hours of sleep in a single day, which may compromise the reliability of the collected data.