IMPLEMENTATION OF PERSONAL FITNESS TRACKER USING PYTHON

Abstract:

The project is focused on creating a personal fitness tracker using Python, with Streamlit as the front-end framework and Python as the back-end. The application offers users a platform to monitor their daily fitness activities, establish goals, and evaluate progress through visual insights.

Introduction:

With growing recognition of fitness and well-being, a personal fitness tracker assists users in tracking their activities effectively. The project employs Streamlit for an interactive web interface and Python for data processing and analytics.

Objectives:

- Create an easy-to-use interface using Streamlit.
- Enable users to log exercises, monitor calories, and track progress.
- Include data visualization and analytics.
- Offer customized recommendations based on user data.

Technology Stack:

Front-end:

- Streamlit: A lightweight web framework to create interactive applications.

Back-end:

- Python: For business logic and data processing.
- SQLite: Database to store user activity logs.
- Matplotlib/Plotly: To display user progress.

Implementation Details:

1. User Authentication: Secure registration and login.

2. Activity Logging: Users can log daily workouts, steps, calories burned, and water consumption.

3. Goal Tracking: Users can create fitness goals and track progress.

4. Data Visualization: Interactive charts for fitness activity insights.

5. Recommendations: AI-driven insights for improved health outcomes.

Code:

Install Dependencies

pip install streamlit pandas sqlite3 matplotlib

Database Setup(SQLite)

Create a database to store user fitness data.

import sqlite3

database.py

```
# Create a database connection
conn = sqlite3.connect("fitness_tracker.db")
cursor = conn. cursor ()

# Create users table
cursor.execute('''
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT UNIQUE NOT NULL,
    password TEXT NOT NULL
)
''')

# Create fitness logs table
cursor.execute('''
```

```
CREATE TABLE IF NOT EXISTS fitness_logs (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL,
    date TEXT NOT NULL,
    steps INTEGER,
    calories_burned INTEGER,
    water_intake REAL,
   exercise_time INTEGER
)
''')

conn.commit()
conn.close()
```

Streamlit Front-end

app.py

```
import streamlit as st
import sqlite3
import pandas as pd
import hashlib
import matplotlib.pyplot as plt

# Function to hash passwords
def hash_password(password):
    return hashlib.sha256(password.encode()).hexdigest()

# Function to verify user login
def verify_user (username, password):
    conn = sqlite3.connect("fitness_tracker.db")
```

```
    cursor = conn.cursor()
    cursor.execute("SELECT password FROM users WHERE username=?", (username,))
    result = cursor.fetchone()
    conn.close()
    if result:
        return result[0] == hash_password(password)
    return False

# Function to register a new user
def register_user(username, password):
    conn = sqlite3.connect("fitness_tracker.db")
    cursor = conn.cursor()
    try:
        cursor.execute("INSERT INTO users (username, password) VALUES (?, ?)",
                       (username, hash_password(password)))
        conn.commit()
        return True
    except sqlite3.IntegrityError:
        return False
    finally:
        conn.close()

# Function to log fitness data
def log_fitness_data (username, steps, calories, water, exercise_time, date):
    conn = sqlite3.connect("fitness_tracker.db")
    cursor = conn.cursor()
    cursor.execute("INSERT INTO fitness_logs (username, date, steps, calories_burned,
water_intake, exercise_time) VALUES (?, ?, ?, ?, ?, ?)",
                   (username, date, steps, calories, water, exercise_time))
```

```
    conn.commit()
    conn.close()

# Function to fetch user fitness data
def fetch_fitness_data(username):
    conn = sqlite3.connect("fitness_tracker.db")
    cursor = conn.cursor()
    cursor.execute("SELECT date, steps, calories_burned, water_intake, exercise_time FROM
fitness_logs WHERE username=? ORDER BY date", (username,))
    data = cursor.fetchall()
    conn.close()
    return data

# Streamlit UI
st.title("Personal Fitness Tracker")

menu = ["Login", "Register", "Log Activity", "View Progress"]
choice = st.sidebar.selectbox("Menu", menu)

if choice == "Register":
    st.subheader("Create a New Account")
    new_user = st.text_input("Username")
    new_pass = st.text_input("Password", type="password")

    if st.button("Register"):
        if register_user(new_user, new_pass):
            st.success("Account created successfully! Please login.")
        else:
            st.error("Username already exists.")
```

```
elif choice == "Login":
    st.subheader("User Login")
    username = st.text_input("Username")
    password = st.text_input("Password", type="password")

    if st.button("Login"):
        if verify_user(username, password):
            st.success(f"Welcome, {username}!")
            st.session_state["username"] = username
        else:
            st.error("Invalid credentials. Please try again.")

elif choice == "Log Activity":
    if "username" in st.session_state:
        st.subheader("Log Your Fitness Activity")
        date = st.date_input("Select Date")
        steps = st.number_input("Steps Walked", min_value=0, step=1)
        calories = st.number_input("Calories Burned", min_value=0, step=1)
        water = st.number_input("Water Intake (liters)", min_value=0.0, step=0.1)
        exercise_time = st.number_input("Exercise Time (minutes)", min_value=0, step=1)

        if st.button("Save Activity"):
            log_fitness_data(st.session_state["username"], steps, calories, water, exercise_time,
date)
            st.success("Activity logged successfully!")
    else:
        st.warning("Please login first.")
```

```
elif choice == "View Progress":
    if "username" in st.session_state:
        st.subheader("Your Progress Overview")
        data = fetch_fitness_data(st.session_state["username"])

        if data:
            df = pd.DataFrame(data, columns=["Date", "Steps", "Calories Burned", "Water Intake", "Exercise Time"])
            st.dataframe(df)

            # Plot steps and calories burned
            fig, ax1 = plt.subplots()
            ax1.set_xlabel("Date")
            ax1.set_ylabel("Steps", color="blue")
            ax1.plot(df["Date"], df["Steps"], marker='o', linestyle='-', color="blue", label="Steps")
            ax1.tick_params(axis='y', labelcolor="blue")

            ax2 = ax1.twinx()
            ax2.set_ylabel("Calories Burned", color="red")
            ax2.plot(df["Date"], df["Calories Burned"], marker='o', linestyle='-', color="red", label="Calories Burned")
            ax2.tick_params(axis='y', labelcolor="red")

            fig.autofmt_xdate()
            st.pyplot(fig)
        else:
            st.info("No activity logged yet.")
    else:
        st.warning("Please login first.")
```

Running the project

streamlit run app.py

open the **localhost URL** to use the tracker

Results & Discussion:

The project effectively facilitates users to monitor fitness data in an organized way. The easy-to-use interface facilitates effortless navigation, and visual analytics enhance user motivation and engagement.

Conclusion:

The Personal Fitness Tracker offers an effective and user-friendly solution for monitoring fitness goals. Future enhancements involve adding wearable device support and AI-driven workout recommendations.