Fitness Tracker

1. **User Authentication and Profile Management**:
   * Allow users to register, log in, and manage their profiles.
   * Store user information such as username, password (hashed for security), email, age, weight, height, etc., in the database.
2. **Activity Logging**:
   * Create tables to log various activities such as running, cycling, weightlifting, swimming, etc.
   * Store details like activity type, duration, distance, calories burned, date/time, and any additional relevant metrics.
3. **Workout Planning and Tracking**:
   * Enable users to create, edit, and delete workout plans.
   * Store workout plans in the database with details such as plan name, description, duration, and associated exercises.
4. **Exercise Database**:
   * Maintain a database of exercises with details like exercise name, category (e.g., cardio, strength), muscle groups targeted, difficulty level, etc.
   * Allow users to search and select exercises when creating workout plans or logging activities.
5. **Progress Tracking**:
   * Implement functionality to track users' progress over time.
   * Store historical data for activities and workouts to visualize progress trends.
6. **Data Analysis and Visualization**:
   * Use SQL queries to analyze fitness data stored in the database.
   * Generate charts and graphs to visualize users' progress, activity distribution, calorie consumption, etc.
   * You can use libraries like Matplotlib or Plotly to create visualizations directly in Python.
7. **Goal Setting and Achievement**:
   * Allow users to set fitness goals (e.g., target weight, running distance, strength gains).
   * Track users' progress towards their goals and provide feedback and achievements when goals are reached.
8. **Social Features**:
   * Create a simple system where users can follow other users by storing relationships in text files or simple database tables.
   * Each user can have a list of followers and people they follow. You can manage these relationships by storing user IDs in simple text files or tables.
   * Users can view activities and progress of users they follow by reading their data from the files or tables.
   * Sharing achievements or workout plans can be implemented by allowing users to write to a shared file or table where other users can read from.
9. **Reminders and Notifications**:
   * Implement reminders by storing scheduled events (like workouts) for each user in a file or table.
   * Create a simple function that checks these scheduled events periodically (maybe every time the user logs in) and notifies the user if there are any upcoming events.
   * Notifications can be basic messages printed to the console, as sending emails or SMS might require external libraries or services.
10. **Data Export and Backup**:
    * Allow users to export their fitness data by writing it to a text file or a specific format you define.
    * Users can trigger this export functionality through your application, and the data can be written to a file that they can download.
    * For backup, you can periodically save the entire database to a backup file (again, a simple text file) at regular intervals or provide a manual backup option.

friend\_connection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Username | FullName | Follower | Follow | C\_Followers | C\_Following |
| Mann | Mann Gupta | Step2: Shivam  If yes: delete shivam |  | Step3:If yes  Shivam  Vishal | Step3:If yes  Shivam Vishal |
| Shivam | Shivam Kumar |  | Step 1: Mann,Aditya Remove mann if step 3  Remove Aditya if Step4 | Mann,Vishal | Mann,Vishal |
| Aditya | Aditya Pandey | Step2:Shivam  Step 4:If No  Delete |  |  |  |
| Vishal | Vishal Jaiswal |  |  | Mann,Shivam | Mann,Shivam |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Username** | FullName | **Username** | **Username** | **Username** | **Username** |

Mann 🡪 Shivam Progress: Shivam daily activity