CS411 Final Project Proposal

1. Project title:
   RunTrack: A Running Data Monitoring App

2. Project Summary:
   RunTrack is a web application designed to help users track and monitor their running sessions. The app will allow users to input essential running metrics such as running time, speed, and pace, providing a detailed overview of each session. Users will be able to log and view their progress over time, helping them optimize their workouts and set personalized fitness goals. The app aims to offer both basic functionality for casual runners and advanced features for serious athletes.

3. Description of application:
   The primary goal of RunTrack is to provide a digital platform for tracking running sessions. The problem it aims to solve is the difficulty in keeping accurate and detailed logs of personal running data without using complex or expensive fitness tracking devices. Our web app will allow users to manually enter data such as the distance they ran, their running duration, pace, and speed, and calculate these metrics for each session.

4. Create component:

   CRUD Operations: Users can create their accounts, log in, update their user information, and delete accounts. Also, for events, we also support create, read, update, and delete operations. The log in process should involve typing in passwords.

   Search Feature: In our web app, users can search for events and other users to make friends using the search bar. Search results will be ranked according to the keywords users type. There will be a database to support the searching operations.

   One creative feature we plan to do is integrating an external API, such as a weather API, to automatically log the weather conditions during the user's run, providing context to their performance on specific days.

   Also, we propose implementing a predictive performance analytics feature using machine learning algorithms. By training models on combined datasets—including user running history, sensor data from accelerometers and gyroscopes, environmental conditions, and historical marathon data—we can forecast future performance metrics such as running times, pace, and endurance levels. This feature will provide users with personalized insights, like the probability of achieving specific running goals or suggestions for training adjustments to enhance performance. Integrating machine learning adds a technically challenging and innovative component that significantly improves the app's functionality and offers users data-driven guidance to reach their fitness objectives.

We plan to develop an interactive visualization dashboard that allows users to explore their running data dynamically. Utilizing advanced visualization libraries like D3.js or Plotly, the dashboard will feature interactive charts and graphs displaying performance over time, comparative analyses with historical marathon data, and geospatial visualizations such as route performance heatmaps and 3D elevation profiles. Users can engage with the data through zooming, filtering, and tooltips, gaining deeper insights into their performance patterns and environmental impacts. This component requires substantial engineering effort and provides a sophisticated tool that enriches the user experience, enabling runners to make informed decisions about their training and goals.

5. Usefulness:
   RunTrack is useful for anyone who runs and wants to keep track of their performance without investing in expensive fitness trackers. Users can track their progress over time and improve their running efficiency by analyzing the data.
   There exists running apps like Keep, but the differences our projects trying to make lie in graphical analysis of performance over time and current trend. Moreover, our app would store entries of both users and existing public datasets.

6. realness:

   1）**Run or Walk Sensor Data Dataset**

- **Data Source**: Run or Walk Dataset on Kaggle
- **Format**: CSV file
- **Data Size**:
  - **Cardinality**: 88,588 sensor data samples
  - **Degree**: 9 attributes per sample
- **Information Captured**:
  - **Sensor Data**:
    - **Acceleration**: `acceleration_x`, `acceleration_y`, `acceleration_z`
    - **Gyroscope**: `gyro_x`, `gyro_y`, `gyro_z`
  - **Activity Type** (`activity`):
    - `"0"`: Walking
    - `"1"`: Running
  - **Wrist Position** (`wrist`):
    - `"0"`: Left wrist
    - `"1"`: Right wrist
  - **Timestamp and User Information**:
    - `date`, `time`, `username`

Application in RunTrack:

We plan to utilize this dataset to develop an activity recognition model based on accelerometer and gyroscope data. By training machine learning algorithms on this data, our app can automatically detect whether a user is running or walking based on sensor inputs from their smartphone or wearable device. This feature enhances the user experience by providing automatic activity logging, especially useful when users forget to manually record their sessions.

2）Berlin Marathon Historical Data

- **Data Source**: Berlin Marathons Data on Kaggle
- **Format**: CSV files
- **Data Size**:
  - **Cardinality**:884,944 records of runners' finishing times from 1974 to 2019
  - **Degree**: 5 attributes per sample
- **Information Captured**:
  - **Finish Time**: Runner's finishing time in HH:MM
  - **Sex**
  - **Age**
  - **Country**
  - **Event Year**: Year of the marathon

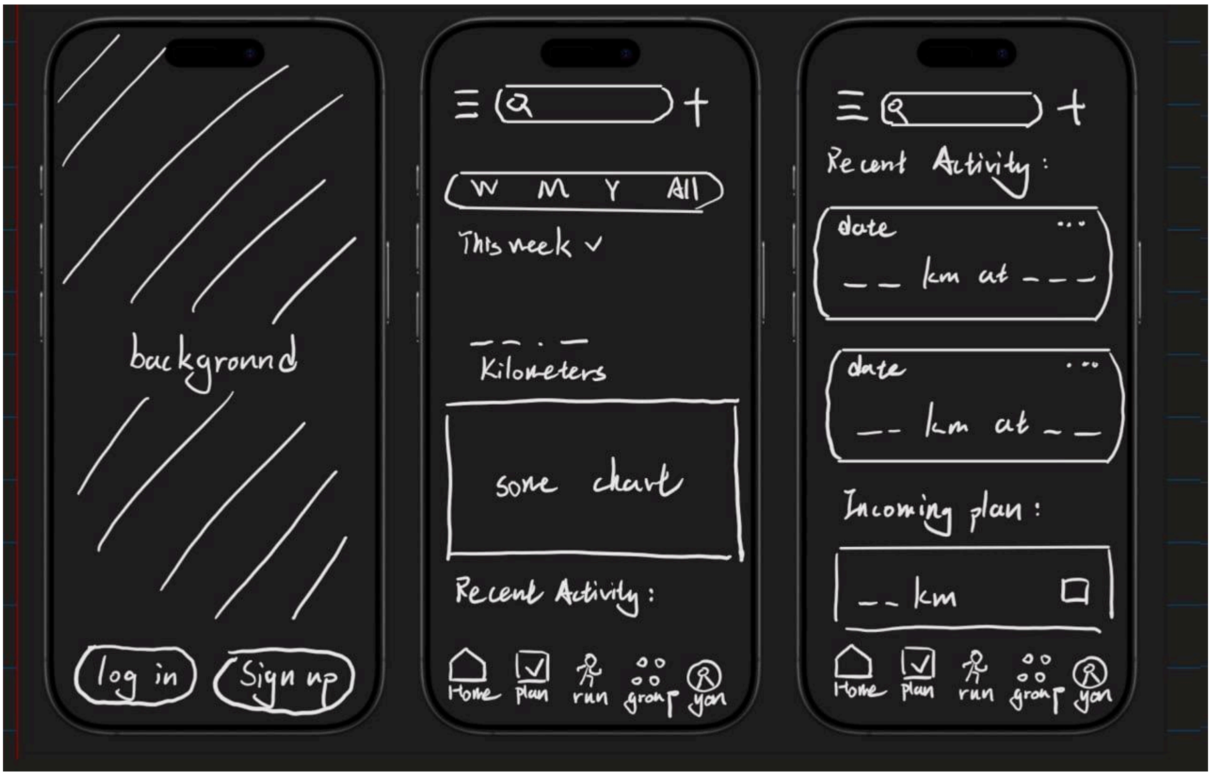
Application in RunTrack:

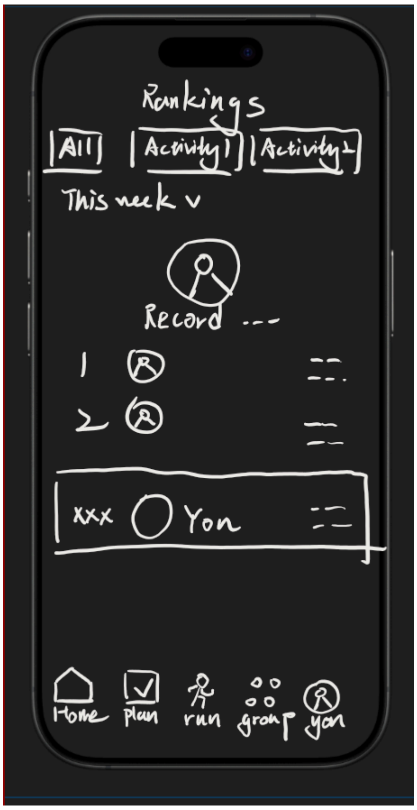
By integrating the Berlin Marathon dataset, we aim to provide users with benchmarking tools to compare their performance against a large pool of marathon runners over several decades. Specifically, our application will:

- **Performance Percentiles**: Calculate and display the percentile ranking of a user's personal best time in comparison to historical marathon finishing times. For example, a user can see that their time places them in the top 30% of all recorded marathon finishers.
- **Demographic Comparisons**: Allow users to compare their performance against runners of similar age and sex, providing more personalized insights.
- **Motivational Insights**: Provide historical context and inspiration by highlighting notable performances or trends from the Berlin Marathon history.

7. functionality:

(1): A low-fidelity UI mockup:





(2): Project work distribution:
Xiaohan Mu: database design and setup, define constraints design schema

Boyu Liu: weather API integration, logic for storing, retrieving, and modifying

Ziheng Qi: unit and integration testing, cloud deployment, and docker setup

Xiaoyang Chen: backend logic for user management, UI/UX design, input data, and frontend API integration