Backend Coding Task

Overview

As part of this coding task, we will design and implement a **ride matching service** for a mobility platform. The goal of this service is to **allocate available drivers to incoming ride requests** efficiently and safely under concurrent load.

Time to Complete

This task should take around **3 hours** to complete.

Requirements

1. Register driver availability.
    a. The service should allow drivers to **register and update their availability** along with their **current location**.
2. Request a ride.
    a. The service should allow riders to **request a ride** by providing their **pickup location**. The destination is not important.
    b. The service should find and allocate the **nearest available driver** closest to the pickup location.
    c. The driver should be **marked as unavailable** for subsequent ride requests.
    d. The rider should receive the **ride and driver details**.
3. Complete the ride.
    a. The service should allow the rider to **mark the ride as completed**.
    b. Upon completion, the driver of the ride should **become available again** for new ride requests.
4. Get list of currently available drivers.
    a. The service should be able to **receive a location** and return the **X nearest drivers**, in ascending order of distance.

Technical Requirements

1. The service should be implemented as a **Java backend.**
2. Data should be stored **in-memory**, no database is required.
3. **Straight line Euclidean** should be used for distance calculations.
4. Best efforts should be made to ensure **thread safety** and appropriate behaviours under concurrent requests.
5. **Unit tests** should be implemented to cover the above scenarios.

Deliverables

1. A public repository link (GitHub, Bitbucket etc) along with a README containing setup and run instructions.