**Dating Suggestions App**

**Design Document**

**Coding Assignment Solution**

**-- Developed by Dinesh Patil**

**November 2024**

**Context**

In recent times, some niche online dating apps have seen massive success. At the heart of these apps is the algorithm to

recommend best match profiles. In order to ensure that users do not end up browsing large amounts of irrelevant results, the

algorithm must return the results based on ranking the matches using weightage of individual matching requirements.

**Problem Statement**

Write code for a dating recommendation engine which would find the closest match for any user. Each registered user is

expected to have provided:  
• Name  
• Gender  
• Age  
• Interests

Recommendation engine applies following rules in given order while identifying the closest matches:

• Gender Rule: Opposite gender is given a preference.  
• Age rule: Closest match in terms of age is given a preference.  
• Interest rule: Closest match in terms of interests is given a preference

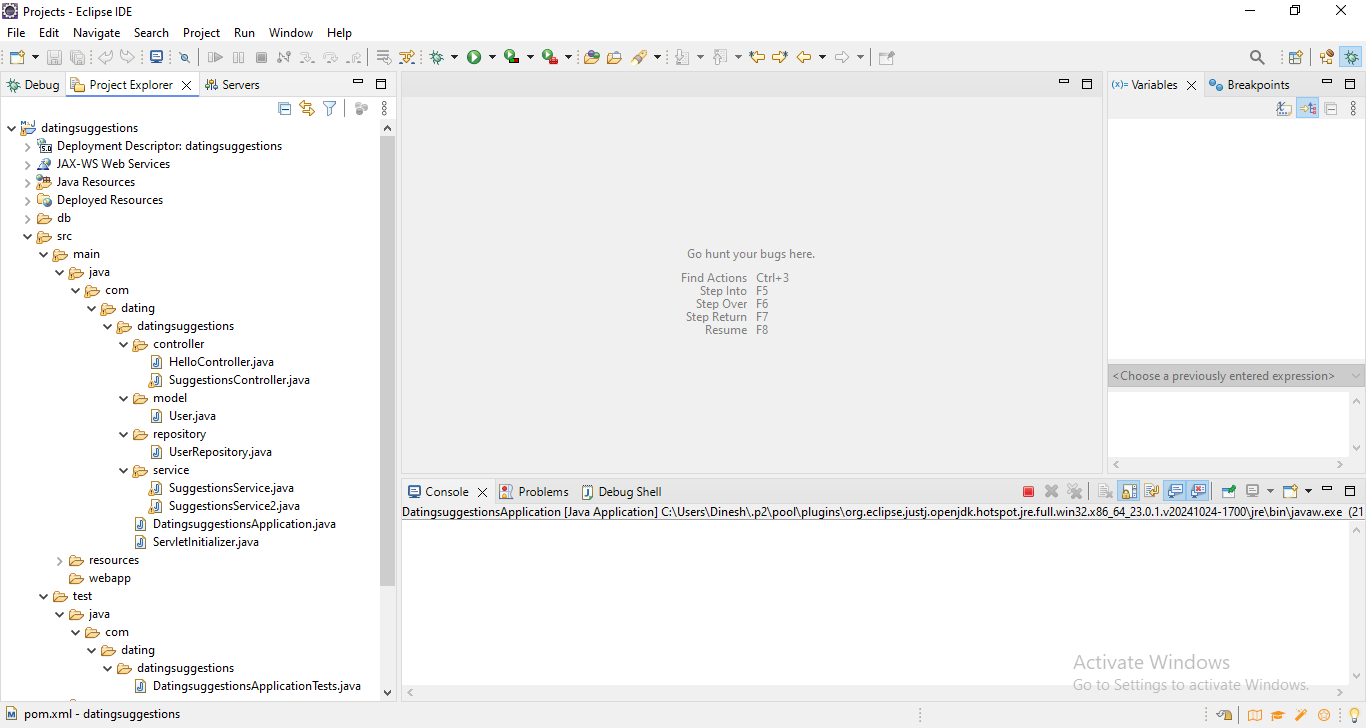
**Solution:**

Tools & Technologies Used:

* Spring boot
* Java 21
* H2 Database – In memory Database
* Junit Tests cases

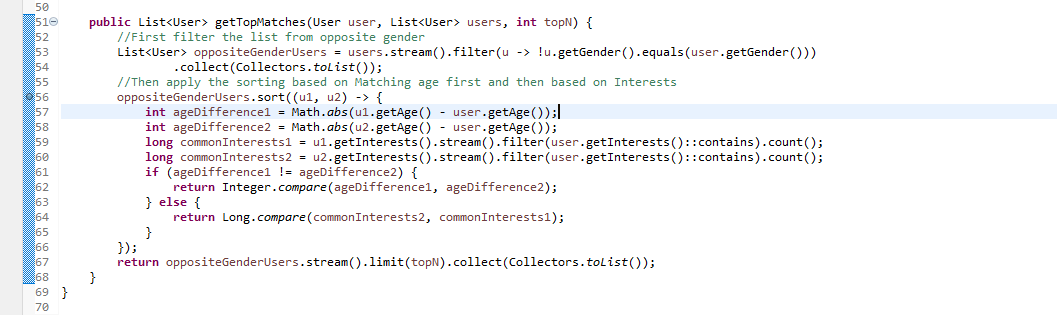
**Project Structure:**

Model Class, Controller, Service Layer, Repository Layer, Unit Tests

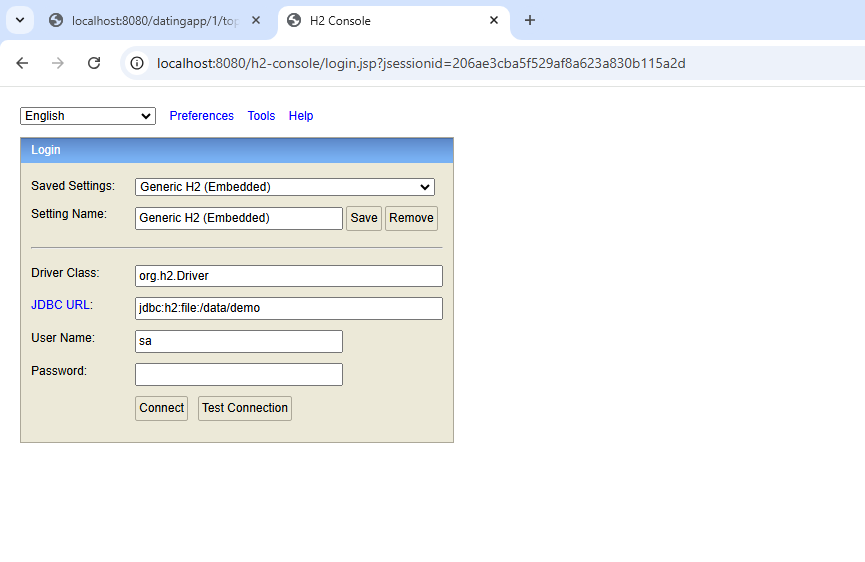


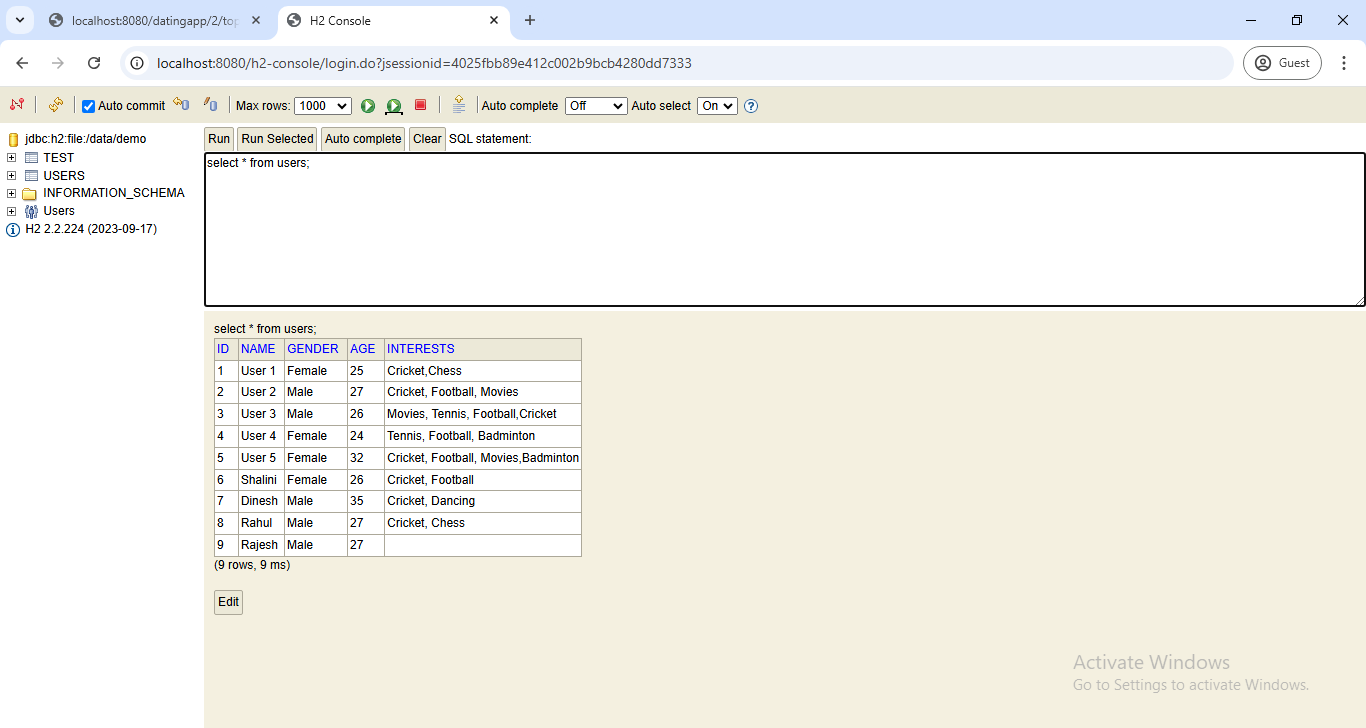
**Main Logic:**

Get all users from the DB. Filter the list using opposite gender first. Then Sort the list on the basis of matching Age and then sort on the basis of Matching interests.



**H2 Database Console:**

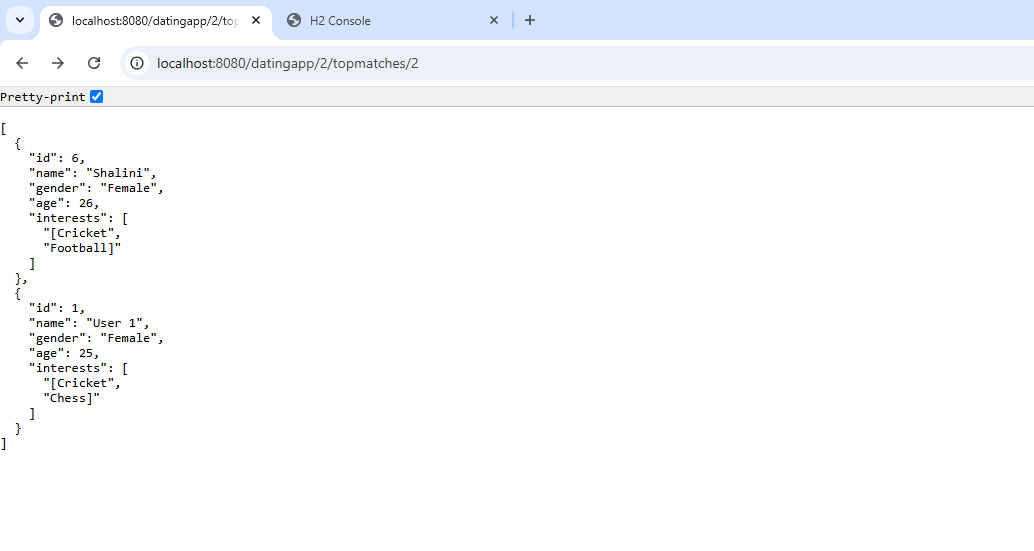




**Running the REST API in the browser:**

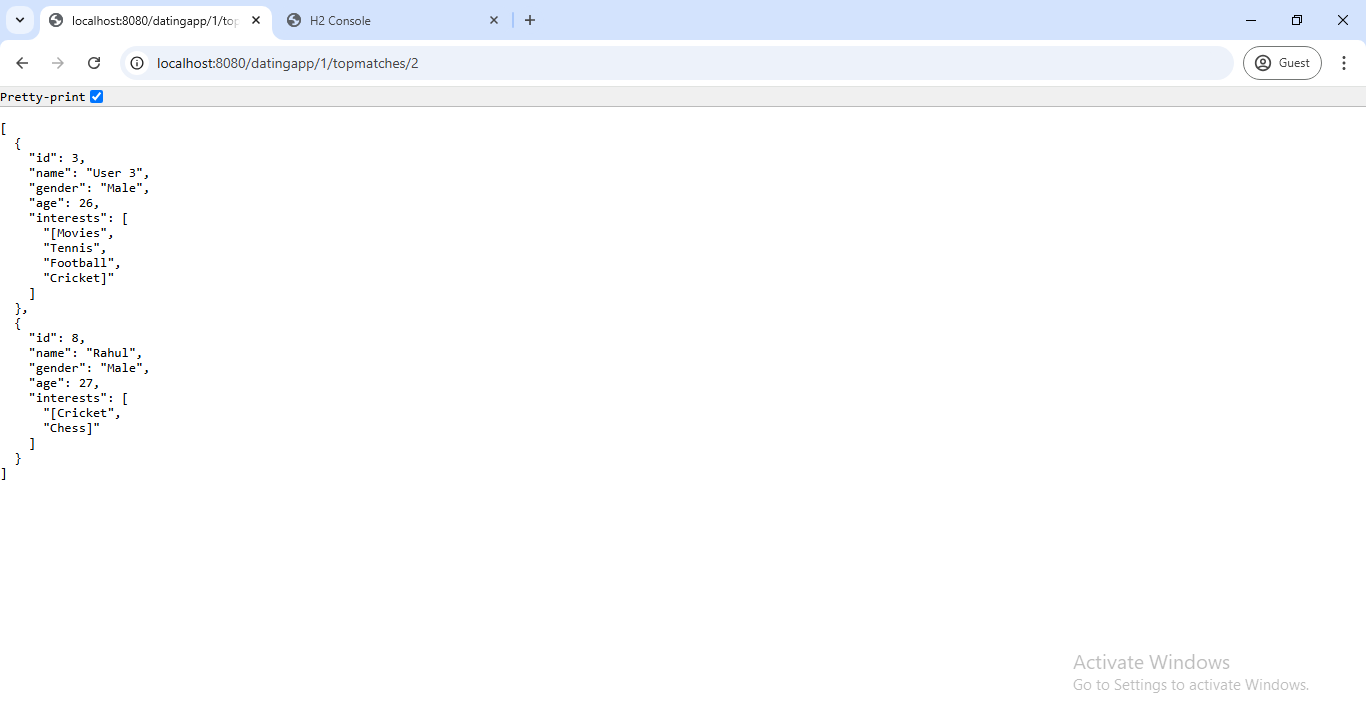
Output for User 2:

<http://localhost:8080/datingapp/2/topmatches/2>

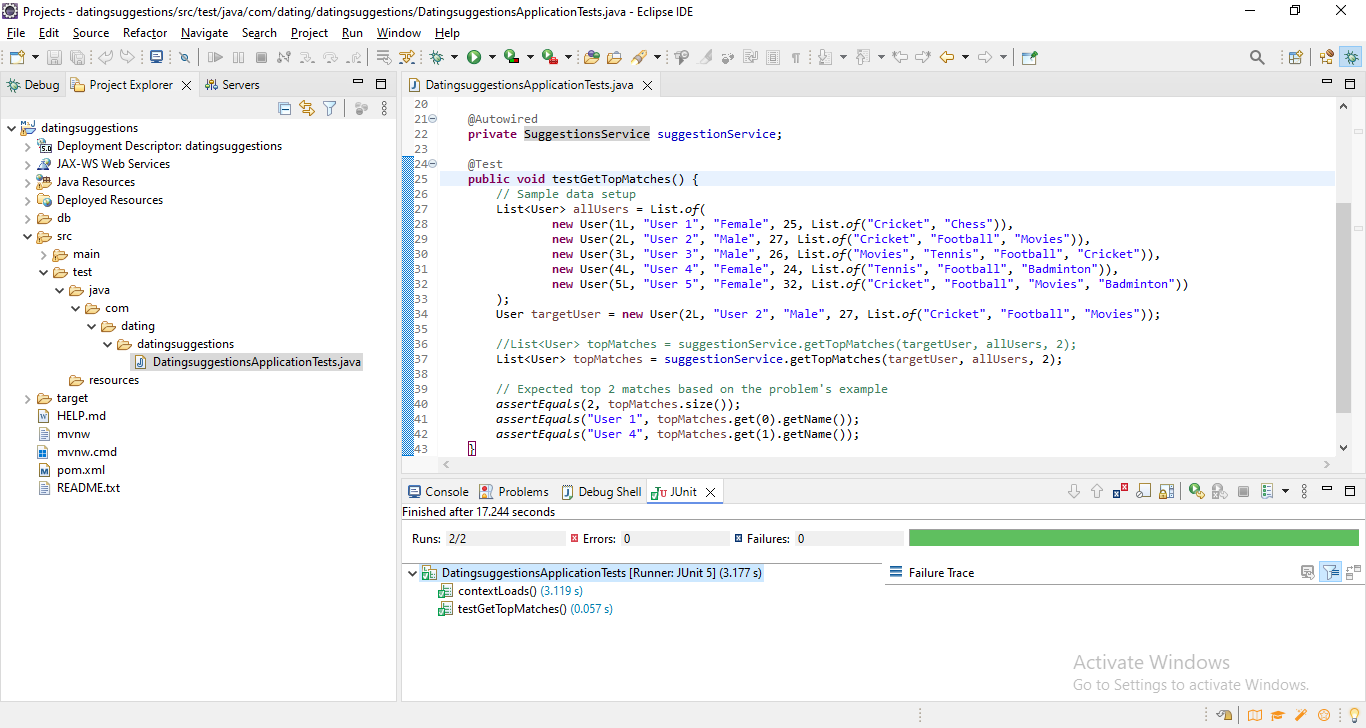


Output for User 1:

<http://localhost:8080/datingapp/1/topmatches/2>



**Unit Tests:**



**Additional future scopes and Enhancements:**

1. Implement UI using Angular and provide User Login and matching result
2. Provide Add user page to add new users
3. Write REST API for Adding/Registering new users
4. Implement caching using Radis cache
5. Other parameters for users can be added like Location, Profile picture etc.

**How to run the project:**

Pre-requisite:  
Java 21:

Download the project source folder datingsuggestions from below Github repository

<https://github.com/dineshpt12/datingapp.git>

Extract the folder in C:\Working\Projects\datingsuggestions

Copy C:\Working\Projects\datingsuggestions\db\data folder to C: drive.

Make sure JDK 21 is installed on the machine if not download JDK 21 from the below link:

<https://www.oracle.com/in/java/technologies/downloads/#jdk21-windows>

Extract the zip jdk-21.0.5 and copy the fonder in C drive:  
Go to command prompt and go to project folder with command:

cd C:\Working\Projects\datingsuggestions

C:\Working\Projects\datingsuggestions>set JAVA\_HOME=C:\MyPrograms\jdk-21.0.5  
C:\Working\Projects\datingsuggestions>set PATH=%PATH%;%JAVA\_HOME%\bin

C:\Working\Projects\datingsuggestions>java -version

java version "21.0.5" 2024-10-15 LTS  
Java(TM) SE Runtime Environment (build 21.0.5+9-LTS-239)  
Java HotSpot(TM) 64-Bit Server VM (build 21.0.5+9-LTS-239, mixed mode, sharing)

Type below command to run the project:

mvnw spring-boot:run

Once server is up, open the browser and run below URL:

<http://localhost:8080/datingapp/2/topmatches/2>

**H2 Database Console:**<http://localhost:8080/h2-console>  
Username: sa  
Password: password

Proposed System Design Architecture diagram for the project:

Application can be deployed using below Architecture to provide Scalability and Resiliency using Cloud, Docker and Kubernetes platform:



