TC101\_Sep24\_ Introduction to Computers project draft #1

https://github.com/angrbrd/friend-finder

Title: CampusLink

App logo:



Idea:

It's an app for college students that aims to connect them with others who have personalities and interests by having them take a personality test upon signing up for the service.

The application utilises an algorithm influenced by networking sites to constantly enhance these connections through user engagement activities that promote academic partnerships and social relationships. The app offers groups based on personalities. Suggests custom events while also displaying a feed, with timely updates and posts. By integrating personality insights with advanced social media-style algorithms,By integrating personality insights and location-based matching it enhances the university experience, making it easier for students to connect, collaborate, and engage.

Google doc question

1-How do you currently meet new people at your university?

Social media

Clubs

Classes

2-Would you be interested in an app that connects you with others based on personality and interests?;

yes / no

Did you ever use personality tests?

Yes./no

3-Do you think using personality matching would help you meet people more easily?

(Yes/No)

Which of these features would be useful for you?

(Check all that apply)

* Groups based on personality
* Suggested study partners
* Event recommendations
* Updates and posts from other students
* Location-based matching

How easy is it for you to find study partners at your uni? (Very easy, Easy, Hard, Very hard)

Would an app for both academic and social connections be helpful? (Yes/No)

suggestions do you have about this app?

Interview question:

### 1. How do you currently meet new people on campus? What tools or platforms do you use most?

### 2. Do you find it easy to connect with people who share your academic or social interests? Why or why not?

### 3. What challenges do you face when trying to form study groups or collaborate with classmates?

### 4. How important is it to you to find people with similar personalities or interests when building new connections?

### 5. Would you be interested in using an app that helps you connect with people based on personality traits and interests? Why or why not?

### 6. How do you think personality matching could improve your social or academic life at university?

### 7. What kind of features would you expect from an app designed to connect students?

### 8. How important is it for you to have personalized event recommendations or group suggestions?

4o

The code to help us build the app

Create a User Model:

public class User {

private int id;

private String name;

private String email;

private String password;

private String personalityType;

private String location;

// Constructor

public User(int id, String name, String email, String password, String personalityType, String location) {

this.id = id;

this.name = name;

this.email = email;

this.password = password;

this.personalityType = personalityType;

this.location = location;

}

// Getters and Setters

// ...

}

Database Connection (using JDBC):

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class Database {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/studentapp";

private static final String USER = "root";

private static final String PASS = "password";

public static Connection getConnection() {

Connection connection = null;

try {

connection = DriverManager.getConnection(DB\_URL, USER, PASS);

} catch (SQLException e) {

e.printStackTrace();

}

return connection;

}

}

User Registration and Login Logic:

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

public class UserService {

// Register user

public boolean registerUser(User user) {

try (Connection connection = Database.getConnection()) {

String sql = "INSERT INTO users (name, email, password, personalityType, location) VALUES (?, ?, ?, ?, ?)";

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, user.getName());

statement.setString(2, user.getEmail());

statement.setString(3, user.getPassword());

statement.setString(4, user.getPersonalityType());

statement.setString(5, user.getLocation());

int rowsInserted = statement.executeUpdate();

return rowsInserted > 0;

} catch (Exception e) {

e.printStackTrace();

}

return false;

}

// User login

public User loginUser(String email, String password) {

try (Connection connection = Database.getConnection()) {

String sql = "SELECT \* FROM users WHERE email = ? AND password = ?";

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, email);

statement.setString(2, password);

ResultSet resultSet = statement.executeQuery();

if (resultSet.next()) {

return new User(

resultSet.getInt("id"),

resultSet.getString("name"),

resultSet.getString("email"),

resultSet.getString("password"),

resultSet.getString("personalityType"),

resultSet.getString("location")

);

}

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

}

Matching Users Based on Personality:

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

public class MatchService {

public List<User> findMatches(User user) {

List<User> matches = new ArrayList<>();

try (Connection connection = Database.getConnection()) {

String sql = "SELECT \* FROM users WHERE personalityType = ? AND location = ?";

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, user.getPersonalityType());

statement.setString(2, user.getLocation());

ResultSet resultSet = statement.executeQuery();

while (resultSet.next()) {

matches.add(new User(

resultSet.getInt("id"),

resultSet.getString("name"),

resultSet.getString("email"),

resultSet.getString("password"),

resultSet.getString("personalityType"),

resultSet.getString("location")

));

}

} catch (Exception e) {

e.printStackTrace();

}

return matches;

}

}

Event Suggestions and Feed (simplified):

public class EventService {

public List<Event> getSuggestedEvents(User user) {

// Logic for suggesting events based on personality and location

}

public List<Post> getFeed(User user) {

// Logic to retrieve a feed of posts (e.g., user updates, news, events)

}

}

### **Database Structure (MySQL Example):**

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255),

email VARCHAR(255),

password VARCHAR(255),

personalityType VARCHAR(50),

location VARCHAR(255)

);

CREATE TABLE events (

id INT AUTO\_INCREMENT PRIMARY KEY,

eventName VARCHAR(255),

eventDate DATE,

location VARCHAR(255),

personalityType VARCHAR(50)

);

CREATE TABLE posts (

id INT AUTO\_INCREMENT PRIMARY KEY,

userId INT,

postContent TEXT,

postDate TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);