|  |  |  |
| --- | --- | --- |
| **Assessment of Open-Ended Lab** | | |
| **Mapped GAs** | **Bloom Taxonomy** | **Marks** |
| GA 2 | P3 | 10 |
| GA 5 | C3 | 5 |
| GA 10 | A3 | 5 |

A blue sign with white letters

AI-generated content may be incorrect.**Open-Ended Lab**

### **Made By: Nimra Qurban(68405)**

***Mini Facebook – A Social Network Simulation***

**Objective:**

Design and implement a simplified social networking system that lets users:

* Add new members
* Create friend connections
* View a user’s friends
* Get friend suggestions (based on mutual friends)

This project demonstrates practical use of **Data Structures** (HashMap, ArrayList,

HashSet) together with a **JavaFX GUI**.

**Motivation:**

Modern social platforms (e.g., Facebook) are essentially **graphs** of users (nodes) and friendships (edges). This project offers a hands-on, student-friendly model to connect **theoretical DSA concepts** with a working GUI app, showing how lists, maps, and sets

power real features like connections and recommendations.

**Concept:**

* Model the network as a **graph**:
  + **Node:** User
  + **Edge:** Friendship **(**undirected**)**
* **Friend suggestions** are generated using **mutual friends** logic (friends of friends

who aren’t already connected).

* **JavaFX** provides a clean UI: ComboBoxes, ListViews, Buttons, TextFields.

**Problem Statement:**

Build a simplified social network where:

* Users can be added dynamically
* Friend connections can be created between existing users
* Friends and suggestions are displayed immediately
* Interaction is **easy** and **visual** with JavaFX

**Design / Ways & Means:**

* **Language**: Java (JDK 8 or later)
* **GUI Framework**: JavaFX
* **Core Data Structures**:
  + **HashMap<String, List<String>>** → adjacency list (user → friends)
  + **ArrayList<String>** → ordered list of friends
  + **HashSet** **<String>** → unique friend suggestions (no duplicates)

**Background / Theory:**

* Graph Theory: **Users = nodes, Friendships = undirected edges**
* Collections Framework:
  + **HashMap** for O(1) average insert/lookup of users/friend lists
  + **ArrayList** for simple, ordered storage of friends
  + **HashSet** to avoid duplicate suggestions

**Procedure / Methodology:**

**1.** **Launch JavaFX** and show a **Welcome Screen** (“MiniFacebook” + Continue).

**2.** On Continue, show the **main UI**.

**3.** Preload demo users (Alice, Bob, Charlie, David, Eve) and a couple of friendships.

**4.** Provide controls to:

* Add users (TextField + Button)
* Add friendships (two ComboBoxes + Button)
* Select a user (ComboBox) → show **Friends** (ListView) & **Suggestions** (ListView)

**5.** Update UI (refresh lists/combos) after every change.

**Flowchart / Block Diagram:**

**┌─────────────────────┐**

**│ Start App │**

**└─────────┬───────────┘**

**│**

**┌─────────▼──────────┐**

**│ Show Welcome UI │**

**│ (MiniFacebook) │**

**└─────────┬──────────┘**

**│ Continue**

**┌─────────▼──────────┐**

**│ Load Main UI │**

**│ + Demo Data │**

**└─────────┬──────────┘**

**│**

**┌─────────▼──────────┐**

**│ User Actions: │**

**│ Add User / Add │**

**│ Friend / Select │**

**│ User │**

**└─────────┬──────────┘**

**│**

**┌─────────▼──────────┐**

**│ Update Data Using │**

**│ HashMap/List/Set │**

**└─────────┬──────────┘**

**│**

**┌─────────▼──────────┐**

**│ Refresh GUI: │**

**│ Friends + │**

**│ Suggestions │**

**└─────────┬──────────┘**

**│**

**┌─────────▼──────────┐**

**│ Continue / Exit │**

**└─────────────────────┘**

**Analysis:**

* **Add User:** O(1) average (HashMap putIfAbsent)
* **Add Friend:** O(1) average to check/add in ArrayList (amortized)
* **View Friends:** O(1) to fetch list from map
* **Suggest Friends:** O(F × FF) where F = # of user’s friends, FF = average # of each friend’s friends

**Results:**

* Users can be added dynamically.
* Friendships update both users (bidirectional).
* Friends and suggestions display instantly.
* Clean, styled UI (gradient background, styled buttons/panels).

**Discussion:**

* Accurately models a **graph** using adjacency lists (HashMap of lists).
* Demonstrates **mutual-friend recommendation** with a HashSet to ensure uniqueness.
* Teaches students how DSA integrates with GUI programming.
* **Limitations:** No persistent storage (data resets on restart); no authentication/profile details; no shortest-path/degree-of-separation yet.

**Conclusion**

MiniFacebook is a concise yet powerful Data Structures project in Java. It shows how HashMap, ArrayList, and HashSet implement a social graph with interactive features via JavaFX. This bridgestheory (graphs, adjacency lists, set uniqueness) with a friendly app —

a solid base for adding BFS/DFS, persistence, and richer features.

**Future Enhancements**

1. Persistence (MySQL/SQLite) to save users and friendships
2. Login/Profiles (per-user information)
3. BFS shortest path (“How many degrees between A and B?”)
4. Weighted recommendations (score by number of mutual friends)
5. Search bar & remove friend feature
6. Import/Export network to JSON/CSV

**References:**

* Oracle JavaFX Documentation
* Robert Lafore — *Data Structures and Algorithms in Java*
* GeeksforGeeks — Graphs & Collections Framework
* Oracle Java SE Docs — Collections (Map, List, Set)

**Code:**

**1) Main.java — Welcome / Splash Screen**

**package** application;

**import** javafx.application.Application;

**import** javafx.geometry.Pos;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Button;

**import** javafx.scene.control.Label;

**import** javafx.scene.layout.VBox;

**import** javafx.stage.Stage;

**public** **class** Main **extends** Application {

Stage window;

@Override

**public** **void** start(Stage stage) {

window = stage;

// Welcome Screen

Label title = **new** Label("Mini FaceBook");

title.setStyle("-fx-font-size: 36px; -fx-font-weight: bold; -fx-text-fill: #1565c0;");

Button continueBtn = **new** Button("Continue →");

continueBtn.setStyle("-fx-background-color: linear-gradient(to right, #4facfe, #00f2fe);" +

"-fx-text-fill: white; -fx-font-weight: bold;" +

"-fx-background-radius: 8; -fx-padding: 10 20; -fx-font-size: 14px;");

VBox welcomeLayout = **new** VBox(20, title, continueBtn);

welcomeLayout.setAlignment(Pos.***CENTER***);

welcomeLayout.setStyle("-fx-background-color: linear-gradient(to bottom, #e3f2fd, #bbdefb);");

Scene welcomeScene = **new** Scene(welcomeLayout, 600, 400);

// Continue → Main Facebook App

continueBtn.setOnAction(e -> window.setScene(FacebookUI.*createMainScene*(window)));

// Show window

window.setScene(welcomeScene);

window.setTitle("🌐 Mini FaceBook");

window.show();

}

**public** **static** **void** main(String[] args) {

*launch*();

}

}

**2) FacebookUI.java — Main Application UI + Data Structure Logic**

**package** application;

**import** javafx.geometry.Insets;

**import** javafx.scene.Scene;

**import** javafx.scene.control.\*;

**import** javafx.scene.layout.\*;

**import** javafx.stage.Stage;

**import** java.util.\*;

**public** **class** FacebookUI {

**static** HashMap<String, List<String>> *network* = **new** HashMap<>();

// Add user

**public** **static** **void** addUser(String name) {

*network*.putIfAbsent(name, **new** ArrayList<>());

}

// Add friend connection

**public** **static** **void** addFriend(String u1, String u2) {

**if** (!*network*.containsKey(u1) || !*network*.containsKey(u2)) **return**;

**if** (!*network*.get(u1).contains(u2)) *network*.get(u1).add(u2);

**if** (!*network*.get(u2).contains(u1)) *network*.get(u2).add(u1);

}

// Get friends

**public** **static** List<String> getFriends(String user) {

**return** *network*.getOrDefault(user, **new** ArrayList<>());

}

// Suggest friends

**public** **static** Set<String> suggestFriends(String user) {

Set<String> suggestions = **new** HashSet<>();

**if** (!*network*.containsKey(user)) **return** suggestions;

**for** (String f : *getFriends*(user)) {

**for** (String ff : *getFriends*(f)) {

**if** (!ff.equals(user) && !*getFriends*(user).contains(ff)) {

suggestions.add(ff);

}

}

}

**return** suggestions;

}

// Build Main UI Scene

**public** **static** Scene createMainScene(Stage stage) {

// Demo users

*addUser*("Alice");

*addUser*("Bob");

*addUser*("Charlie");

*addUser*("David");

*addUser*("Eve");

*addFriend*("Alice", "Bob");

*addFriend*("Bob", "Charlie");

// GUI elements

ComboBox<String> userDropdown = **new** ComboBox<>();

ListView<String> friendList = **new** ListView<>();

ListView<String> suggestList = **new** ListView<>();

TextField newUserField = **new** TextField();

Button addUserBtn = **new** Button("Add User");

ComboBox<String> user1 = **new** ComboBox<>();

ComboBox<String> user2 = **new** ComboBox<>();

Button addFriendBtn = **new** Button("Add Friend");

// Refresh users

Runnable refresh = () -> {

userDropdown.getItems().setAll(*network*.keySet());

user1.getItems().setAll(*network*.keySet());

user2.getItems().setAll(*network*.keySet());

};

refresh.run();

// Show friends & suggestions

userDropdown.setOnAction(e -> {

String u = userDropdown.getValue();

**if** (u != **null**) {

friendList.getItems().setAll(*getFriends*(u));

suggestList.getItems().setAll(*suggestFriends*(u));

}

});

// Add user

addUserBtn.setOnAction(e -> {

String name = newUserField.getText().trim();

**if** (!name.isEmpty()) {

*addUser*(name);

newUserField.clear();

refresh.run();

}

});

// Add friend

addFriendBtn.setOnAction(e -> {

String u1 = user1.getValue();

String u2 = user2.getValue();

**if** (u1 != **null** && u2 != **null** && !u1.equals(u2)) {

*addFriend*(u1, u2);

refresh.run();

}

});

// Styling

String buttonStyle = "-fx-background-color: linear-gradient(to right, #4facfe, #00f2fe);" +

"-fx-text-fill: white; -fx-font-weight: bold;" +

"-fx-background-radius: 8; -fx-padding: 5 15;";

String panelStyle = "-fx-background-color: #f0f9ff; -fx-border-color: #90caf9; -fx-border-radius: 10; -fx-background-radius: 10;";

String labelStyle = "-fx-font-size: 14px; -fx-font-weight: bold; -fx-text-fill: #1565c0;";

addUserBtn.setStyle(buttonStyle);

addFriendBtn.setStyle(buttonStyle);

// Left section

VBox left = **new** VBox(10,

**new** Label("Select User:"),

userDropdown,

**new** Label("Friends:"),

friendList,

**new** Label("Suggestions:"),

suggestList

);

left.setPadding(**new** Insets(15));

left.setStyle(panelStyle);

((Label) left.getChildren().get(0)).setStyle(labelStyle);

((Label) left.getChildren().get(2)).setStyle(labelStyle);

((Label) left.getChildren().get(4)).setStyle(labelStyle);

// Right section

VBox right = **new** VBox(15,

**new** Label("Add New User:"),

**new** HBox(10, newUserField, addUserBtn),

**new** Label("Add Friend:"),

**new** HBox(10, user1, user2, addFriendBtn)

);

right.setPadding(**new** Insets(15));

right.setStyle(panelStyle);

((Label) right.getChildren().get(0)).setStyle(labelStyle);

((Label) right.getChildren().get(2)).setStyle(labelStyle);

// Root layout

HBox root = **new** HBox(20, left, right);

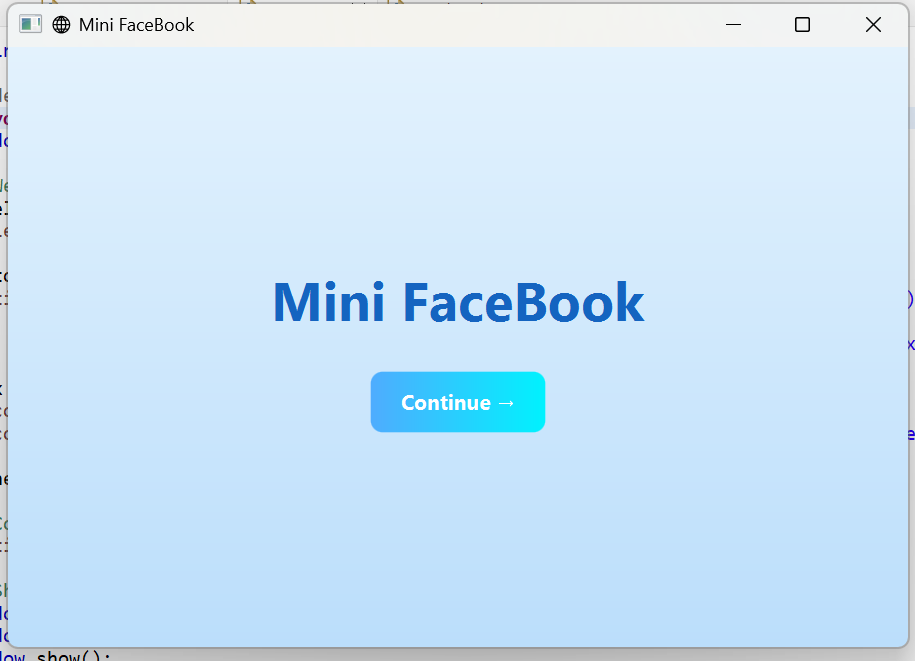
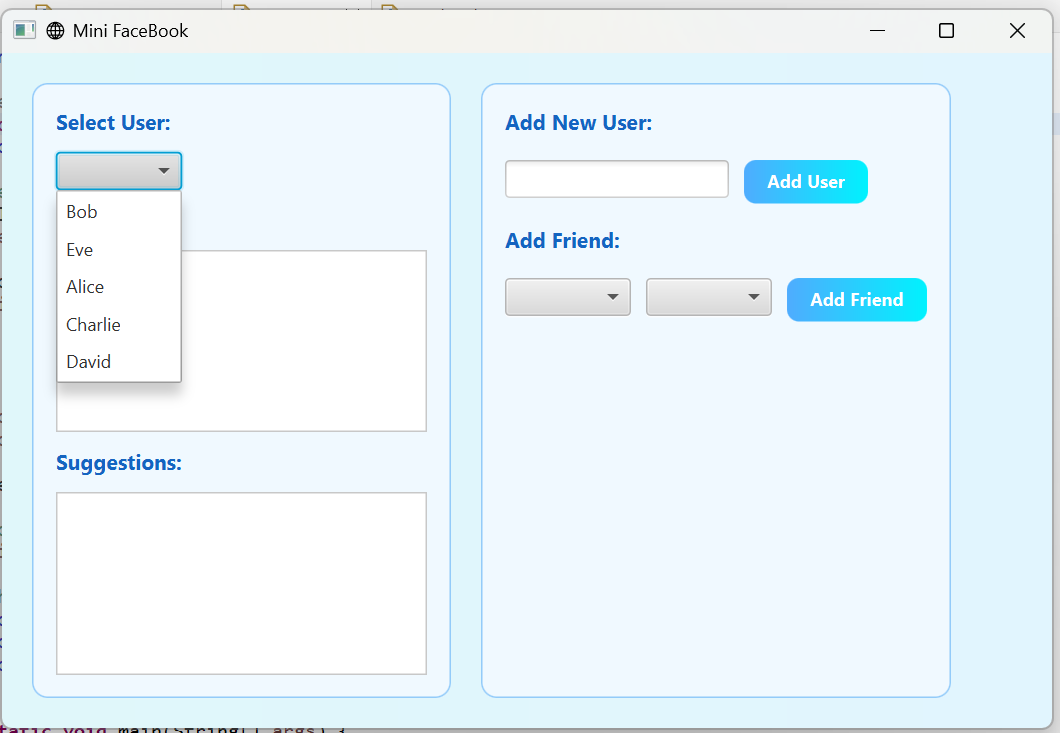
root.setPadding(**new** Insets(20));

root.setStyle("-fx-background-color: linear-gradient(to right, #e0f7fa, #e1f5fe);");

**return** **new** Scene(root, 700, 450);

}

}

**Outputs:**

### 

### 