HERE IS THE CODE:

package gam;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Scanner;

import java.sql.Date;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int choice;

do {

System.out.println("Menu:");

System.out.println("1. Customer");

System.out.println("2. Advocate");

System.out.println("3. Appointment");

// System.out.println("4. Service");

System.out.println("0. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

switch (choice) {

case 1:

handleCustomerMenu(scanner);

break;

case 2:

handleAdvocateMenu(scanner);

break;

case 3:

handleAppointmentMenu(scanner);

break;

// case 4:

// handleServiceMenu(scanner);

// break;

case 0:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice. Please try again.");

break;

}

} while (choice != 0);

scanner.close();

}

private static void handleCustomerMenu(Scanner scanner) {

int choice;

do {

System.out.println("Customer Menu:");

System.out.println("1. Register Customer");

System.out.println("2. Modify Customer Details");

System.out.println("3. Delete Customer Record");

System.out.println("4. View Single Record");

System.out.println("5. View All Records");

System.out.println("0. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

switch (choice) {

case 1:

// Register Customer logic

scanner.nextLine();

System.out.println("Enter customer name: ");

String name = scanner.nextLine();

System.out.println("Enter customer address: ");

String address = scanner.nextLine();

// scanner.next();

System.out.println("Enter customer contact information: ");

String number = scanner.nextLine();

// scanner.next();

System.out.println(name+" "+address+" "+number);

try {

// Establish the database connection

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant");

// Prepare the SQL insert statement

String insertQuery = "INSERT INTO customer (name, address, number) VALUES (?, ?, ?)";

PreparedStatement preparedStatement = connection.prepareStatement(insertQuery);

preparedStatement.setString(1, name);

preparedStatement.setString(2, address);

preparedStatement.setString(3, number);

// Execute the insert statement

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

System.out.println("Customer registered successfully!");

} else {

System.out.println("Failed to register customer.");

}

// Close the database connection and resources

preparedStatement.close();

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 2:

// Modify Customer Details logic

scanner.nextLine();

try (Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant")) {

// Step 2: Get customer ID input from user

System.out.print("Enter customer ID to modify: ");

int customerId = scanner.nextInt();

// Step 3: Check if customer ID exists in the database

String checkQuery = "SELECT \* FROM customer WHERE id = ?";

try (PreparedStatement checkStatement = connection.prepareStatement(checkQuery)) {

checkStatement.setInt(1, customerId);

ResultSet resultSet = checkStatement.executeQuery();

if (!resultSet.next()) {

System.out.println("Customer ID does not exist in the database.");

break;

}

}

// Get the field to modify from the user

System.out.println("Select the field to modify:");

System.out.println("1. Name");

System.out.println("2. Address");

System.out.println("3. Phone Number");

int fieldChoice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

// Get the new value for the selected field

String newValue = "";

switch (fieldChoice) {

case 1:

System.out.print("Enter new name: ");

newValue = scanner.nextLine();

break;

case 2:

System.out.print("Enter new address: ");

newValue = scanner.nextLine();

break;

case 3:

System.out.print("Enter new phone number: ");

newValue = scanner.nextLine();

break;

default:

System.out.println("Invalid field choice.");

return;

}

// Execute the update query

String updateQuery = "";

switch (fieldChoice) {

case 1:

updateQuery = "UPDATE customer SET name = ? WHERE id = ?";

break;

case 2:

updateQuery = "UPDATE customer SET address = ? WHERE id = ?";

break;

case 3:

updateQuery = "UPDATE customer SET number = ? WHERE id = ?";

break;

}

try (PreparedStatement statement = connection.prepareStatement(updateQuery)) {

statement.setString(1, newValue);

statement.setInt(2, customerId);

//Execute the query

int rowsAffected = statement.executeUpdate();

System.out.println(rowsAffected + " row(s) updated.");

}

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 3:

// Delete Customer Record logic

scanner.nextLine();

System.out.println("Enter the ADMIN CODE: ");

Integer code=scanner.nextInt();

if(code!=6949) {

System.out.println("ACCESS DENIED ");

break;

}

try (Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant")) {

// Step 2: Get customer ID input from user

System.out.println("Enter customer ID to delete: ");

int customerId = scanner.nextInt();

scanner.nextLine();

// Step 3: Check if customer ID exists in the database

String checkQuery = "SELECT \* FROM customer WHERE id = ?";

String delQueryString="DELETE FROM employees WHERE id = ?";

try (PreparedStatement checkStatement = connection.prepareStatement(checkQuery)) {

checkStatement.setInt(1, customerId);

ResultSet resultSet = checkStatement.executeQuery();

if (!resultSet.next()) {

System.out.println("Customer ID does not exist in the database.");

break;

}else {

int id1 = resultSet.getInt("id");

String name1 = resultSet.getString("name");

String address1 = resultSet.getString("address");

String number1 = resultSet.getString("number");

//

System.out.println("ID: " + id1);

System.out.println("Name: " + name1);

System.out.println("Adress: " + address1);

System.out.println("Number: " + number1);

System.out.println("======================");

}

System.out.println("Do you want to delete: Yes for 1 OR No for 0");

Integer res=scanner.nextInt();

if(res==1) {

break;

}else {

// Create a SQL statement

String sql = "DELETE FROM customer WHERE id = ?";

PreparedStatement statement = connection.prepareStatement(sql);

// Set the ID parameter

// Replace with your custom ID

statement.setInt(1, customerId);

// Execute the SQL query

int rowsAffected = statement.executeUpdate();

// Check the number of rows affected

if (rowsAffected > 0) {

System.out.println("Data deleted successfully.");

} else {

System.out.println("No data found to delete.");

}

// Close the statement and connection

statement.close();

}

connection.close();

}

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 4:

// View Single Record logic

System.out.println("Enter customer ID to show: ");

int customerId = scanner.nextInt();

scanner.nextLine();

try(Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant")){

// Step 3: Check if customer ID exists in the database

String checkQuery = "SELECT \* FROM customer WHERE id = ?";

try (PreparedStatement checkStatement = connection.prepareStatement(checkQuery)) {

checkStatement.setInt(1, customerId);

ResultSet resultSet = checkStatement.executeQuery();

if (!resultSet.next()) {

System.out.println("Customer ID does not exist in the database.");

break;

}else {

int id1 = resultSet.getInt("id");

String name1 = resultSet.getString("name");

String address1 = resultSet.getString("address");

String number1 = resultSet.getString("number");

//

System.out.println("ID: " + id1);

System.out.println("Name: " + name1);

System.out.println("Adress: " + address1);

System.out.println("Number: " + number1);

System.out.println("======================");

}} catch (SQLException e) {

e.printStackTrace();

}

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 5:

// View All Records logic

String jdbcUrl = "jdbc:mysql://localhost:3306/demo";

String username = "root";

String password = "jayant";

try {

// Establish the database connection

Connection connection = DriverManager.getConnection(jdbcUrl, username, password);

String query = "SELECT \* FROM customer";

// Create a statement object

try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {

// Execute the query

try (ResultSet resultSet = preparedStatement.executeQuery()) {

// Process the result set

while (resultSet.next()) {

int id1 = resultSet.getInt("id");

String name1 = resultSet.getString("name");

String address1 = resultSet.getString("address");

String number1 = resultSet.getString("number");

// String email = resultSet.getString("email");

// ... Retrieve other columns

System.out.println("ID: " + id1);

System.out.println("Name: " + name1);

System.out.println("Adress: " + address1);

System.out.println("Number: " + number1);

System.out.println("======================");

}

}

} // PreparedStatement and ResultSet are automatically closed here

// Close the connection

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 0:

System.out.println("Exiting Customer Menu...");

break;

default:

System.out.println("Invalid choice. Please try again.");

break;

}

} while (choice != 0);

}

private static void handleAdvocateMenu(Scanner scanner) {

int choice;

do {

System.out.println("Advocate Menu:");

System.out.println("1. Register Advocate");

System.out.println("2. Modify Advocate Details");

System.out.println("3. Delete Advocate Record");

System.out.println("4. View Single Record");

System.out.println("5. View All Records");

System.out.println("6. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

switch (choice) {

case 1:

// register advocate

scanner.nextLine();

System.out.println("Enter Advocate name: ");

String name = scanner.nextLine();

System.out.println("Enter Advocate speciality: ");

String speciality = scanner.nextLine();

// scanner.next();

// scanner.next();

try {

// Establish the database connection

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant");

// Prepare the SQL insert statement

String insertQuery = "INSERT INTO advocate (name, speciality) VALUES (?, ?)";

PreparedStatement preparedStatement = connection.prepareStatement(insertQuery);

preparedStatement.setString(1, name);

preparedStatement.setString(2, speciality);

// Execute the insert statement

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

System.out.println("Customer registered successfully!");

} else {

System.out.println("Failed to register customer.");

}

// Close the database connection and resources

preparedStatement.close();

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 2:

// Modify Advocate Details logic

scanner.nextLine();

try (Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant")) {

// Step 2: Get Advocate ID input from user

System.out.print("Enter Adovate ID to modify: ");

int customerId = scanner.nextInt();

// Step 3: Check if customer ID exists in the database

String checkQuery = "SELECT \* FROM advocate WHERE id = ?";

try (PreparedStatement checkStatement = connection.prepareStatement(checkQuery)) {

checkStatement.setInt(1, customerId);

ResultSet resultSet = checkStatement.executeQuery();

if (!resultSet.next()) {

System.out.println("Advocate ID does not exist in the database.");

break;

}

}

// Get the field to modify from the user

System.out.println("Select the field to modify:");

System.out.println("1. Name");

System.out.println("2. Speciality");

int fieldChoice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

// Get the new value for the selected field

String newValue = "";

switch (fieldChoice) {

case 1:

System.out.print("Enter new name: ");

newValue = scanner.nextLine();

break;

case 2:

System.out.print("Enter new speciality: ");

newValue = scanner.nextLine();

break;

default:

System.out.println("Invalid field choice.");

return;

}

// Execute the update query

String updateQuery = "";

switch (fieldChoice) {

case 1:

updateQuery = "UPDATE advocate SET name = ? WHERE id = ?";

break;

case 2:

updateQuery = "UPDATE advocate SET speciality = ? WHERE id = ?";

break;

}

try (PreparedStatement statement = connection.prepareStatement(updateQuery)) {

statement.setString(1, newValue);

statement.setInt(2, customerId);

//Execute the query

int rowsAffected = statement.executeUpdate();

System.out.println(rowsAffected + " row(s) updated.");

}

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 3:

// Delete Advocate Logic

scanner.nextLine();

System.out.println("Enter the ADMIN CODE: ");

Integer code=scanner.nextInt();

if(code!=6949) {

System.out.println("ACCESS DENIED ");

break;

}

try (Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant")) {

// Step 2: Get customer ID input from user

System.out.println("Enter Advocate ID to delete: ");

int customerId = scanner.nextInt();

scanner.nextLine();

// Step 3: Check if customer ID exists in the database

String checkQuery = "SELECT \* FROM advocate WHERE id = ?";

String delQueryString="DELETE FROM advocate WHERE id = ?";

try (PreparedStatement checkStatement = connection.prepareStatement(checkQuery)) {

checkStatement.setInt(1, customerId);

ResultSet resultSet = checkStatement.executeQuery();

if (!resultSet.next()) {

System.out.println("Customer ID does not exist in the database.");

break;

}else {

int id1 = resultSet.getInt("id");

String name1 = resultSet.getString("name");

String address1 = resultSet.getString("speciality");

//

System.out.println("ID: " + id1);

System.out.println("Name: " + name1);

System.out.println("Speciality: " + address1);

System.out.println("======================");

}

System.out.println("Do you want to delete: Yes for 1 OR No for 0");

Integer res=scanner.nextInt();

if(res==0) {

break;

}else {

// Create a SQL statement

String sql = "DELETE FROM advocate WHERE id = ?";

PreparedStatement statement = connection.prepareStatement(sql);

// Set the ID parameter

// Replace with your custom ID

statement.setInt(1, customerId);

// Execute the SQL query

int rowsAffected = statement.executeUpdate();

// Check the number of rows affected

if (rowsAffected > 0) {

System.out.println("Data deleted successfully.");

} else {

System.out.println("No data found to delete.");

}

// Close the statement and connection

statement.close();

}

connection.close();

}

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 4:

// View Single Record logic

System.out.println("Enter advocate ID to show: ");

int customerId = scanner.nextInt();

scanner.nextLine();

try(Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant")){

// Step 3: Check if customer ID exists in the database

String checkQuery = "SELECT \* FROM advocate WHERE id = ?";

try (PreparedStatement checkStatement = connection.prepareStatement(checkQuery)) {

checkStatement.setInt(1, customerId);

ResultSet resultSet = checkStatement.executeQuery();

if (!resultSet.next()) {

System.out.println("Advocate ID does not exist in the database.");

break;

}else {

int id1 = resultSet.getInt("id");

String name1 = resultSet.getString("name");

String speciality1 = resultSet.getString("speciality");

//

System.out.println("ID: " + id1);

System.out.println("Name: " + name1);

System.out.println("Speciality: " + speciality1);

System.out.println("======================");

}} catch (SQLException e) {

e.printStackTrace();

}

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 5:

// View All Records logic

String jdbcUrl = "jdbc:mysql://localhost:3306/demo";

String username = "root";

String password = "jayant";

try {

// Establish the database connection

Connection connection = DriverManager.getConnection(jdbcUrl, username, password);

String query = "SELECT \* FROM advocate";

// Create a statement object

try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {

// Execute the query

try (ResultSet resultSet = preparedStatement.executeQuery()) {

// Process the result set

while (resultSet.next()) {

int id1 = resultSet.getInt("id");

String name1 = resultSet.getString("name");

String address1 = resultSet.getString("speciality");

// String email = resultSet.getString("email");

// ... Retrieve other columns

System.out.println("ID: " + id1);

System.out.println("Name: " + name1);

System.out.println("Speciality: " + address1);

System.out.println("======================");

}

}

} // PreparedStatement and ResultSet are automatically closed here

// Close the connection

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 6:

System.out.println("Exiting Advocate Menu...");

choice=0;

break;

default:

System.out.println("Invalid choice. Please try again.");

break;

}

} while (choice != 0);

}

private static void handleAppointmentMenu(Scanner scanner) {

// Implement the logic for handling the Appointment menu

int choice;

do {

System.out.println("Appointment Menu:");

System.out.println("0. Book an Appointment");

System.out.println("1. Modify Appointment Details");

System.out.println("2. Delete an Appointment");

System.out.println("3. View Single Record");

System.out.println("4. View All Records");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

switch (choice) {

case 0:

// Book an Appointment logic

System.out.println("All Advocate");

// View All Advocate Records logic

String jdbcUrl = "jdbc:mysql://localhost:3306/demo";

String username = "root";

String password = "jayant";

try {

// Establish the database connection

Connection connection = DriverManager.getConnection(jdbcUrl, username, password);

String query = "SELECT \* FROM advocate";

// Create a statement object

try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {

// Execute the query

try (ResultSet resultSet = preparedStatement.executeQuery()) {

// Process the result set

while (resultSet.next()) {

int id1 = resultSet.getInt("id");

String name1 = resultSet.getString("name");

String address1 = resultSet.getString("speciality");

// String email = resultSet.getString("email");

// ... Retrieve other columns

System.out.println("ID: " + id1);

System.out.println("Name: " + name1);

System.out.println("Speciality: " + address1);

System.out.println("======================");

}

}

} // PreparedStatement and ResultSet are automatically closed here

// Close the connection

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

// View All customer Records logic

System.out.println("All Customer");

try {

// Establish the database connection

Connection connection = DriverManager.getConnection(jdbcUrl, username, password);

String query = "SELECT \* FROM customer";

// Create a statement object

try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {

// Execute the query

try (ResultSet resultSet = preparedStatement.executeQuery()) {

// Process the result set

while (resultSet.next()) {

int id1 = resultSet.getInt("id");

String name1 = resultSet.getString("name");

String address1 = resultSet.getString("address");

String number1 = resultSet.getString("number");

// String email = resultSet.getString("email");

// ... Retrieve other columns

System.out.println("ID: " + id1);

System.out.println("Name: " + name1);

System.out.println("Adress: " + address1);

System.out.println("Number: " + number1);

System.out.println("======================");

}

}

} // PreparedStatement and ResultSet are automatically closed here

// Close the connection

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

//Appointment

scanner.nextLine();

System.out.println("Enter Customer ID: ");

Integer customerId=scanner.nextInt();

System.out.println("Enter Advocate ID: ");

Integer advocateId= scanner.nextInt();

scanner.nextLine();

System.out.println("Enter date for Appointment ");

String date=scanner.nextLine();

Date myDate = Date.valueOf(date);

System.out.println(myDate);

try {

// Establish the database connection

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant");

// Prepare the SQL insert statement

String insertQuery = "INSERT INTO appointment (custid, advid,appdate) VALUES (?, ?,?)";

PreparedStatement preparedStatement = connection.prepareStatement(insertQuery);

preparedStatement.setInt(1, customerId);

preparedStatement.setInt(2, advocateId);

preparedStatement.setDate(3, myDate);

// Execute the insert statement

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

System.out.println("Appointment registered successfully!");

} else {

System.out.println("Failed to set Apponitment.");

}

// Close the database connection and resources

preparedStatement.close();

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 1:

// Modify Appointment Details logic

// Modify Advocate Details logic

scanner.nextLine();

Date myDate1 = null;

try (Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant")) {

// Step 2: Get Advocate ID input from user

System.out.print("Enter Adovate ID to modify: ");

int customerId1 = scanner.nextInt();

// Step 3: Check if customer ID exists in the database

String checkQuery = "SELECT \* FROM appointment WHERE appid = ?";

try (PreparedStatement checkStatement = connection.prepareStatement(checkQuery)) {

checkStatement.setInt(1, customerId1);

ResultSet resultSet = checkStatement.executeQuery();

if (!resultSet.next()) {

System.out.println("Appointment does not exist in the database.");

break;

}

}

// Get the field to modify from the user

System.out.println("Select the field to modify:");

System.out.println("1.Customer ID");

System.out.println("2.Advocate ID");

System.out.println("3 Date");

int fieldChoice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

// Get the new value for the selected field

String newValue = "";

switch (fieldChoice) {

case 1:

System.out.print("Enter new Customer ID for Different Customer: ");

newValue = scanner.nextLine();

break;

case 2:

System.out.print("Enter new Advocate ID for Different Advocate: ");

newValue = scanner.nextLine();

break;

case 3:

System.out.print("Enter new date: ");

newValue = scanner.nextLine();

myDate1= Date.valueOf(newValue);

break;

default:

System.out.println("Invalid field choice.");

return;

}

System.out.println(myDate1);

// Execute the update query

String updateQuery = "";

switch (fieldChoice) {

case 1:

updateQuery = "UPDATE appointment SET custid = ? WHERE appid = ?";

break;

case 2:

updateQuery = "UPDATE appointment SET advid = ? WHERE appid = ?";

break;

case 3:

updateQuery = "UPDATE appointment SET date = ? WHERE appid = ?";

break;

}

try (PreparedStatement statement = connection.prepareStatement(updateQuery)) {

if(fieldChoice!=3){

statement.setString(1, newValue);

}else {

statement.setDate(1, myDate1);

}

statement.setInt(2, customerId1);

//Execute the query

int rowsAffected = statement.executeUpdate();

System.out.println(rowsAffected + " row(s) updated.");

}

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 2:

// Delete an Appointment logic

scanner.nextLine();

System.out.println("Enter the ADMIN CODE: ");

Integer code=scanner.nextInt();

if(code!=6949) {

System.out.println("ACCESS DENIED ");

break;

}

try (Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant")) {

// Step 2: Get customer ID input from user

System.out.println("Enter Appointment ID to delete: ");

int customerId1 = scanner.nextInt();

scanner.nextLine();

// Step 3: Check if customer ID exists in the database

String checkQuery = "SELECT \* FROM appointment WHERE appid = ?";

String delQueryString="DELETE FROM advocate WHERE id = ?";

try (PreparedStatement checkStatement = connection.prepareStatement(checkQuery)) {

checkStatement.setInt(1, customerId1);

ResultSet resultSet = checkStatement.executeQuery();

if (!resultSet.next()) {

System.out.println("Appointment does not exist in the database.");

break;

}else {

int id1 = resultSet.getInt("appid");

String name1 = resultSet.getString("custid");

String address1 = resultSet.getString("advid");

Date myDate2=resultSet.getDate("appdate");

// String email = resultSet.getString("email");

// ... Retrieve other columns

System.out.println("ID: " + id1);

System.out.println("Customer ID: " + name1);

System.out.println("Advocate ID: " + address1);

System.out.println("Date:" + myDate2);

System.out.println("======================");

}

System.out.println("Do you want to delete: Yes for 1 OR No for 0");

Integer res=scanner.nextInt();

if(res==1) {

break;

}else {

// Create a SQL statement

String sql = "DELETE FROM appointment WHERE appid = ?";

PreparedStatement statement = connection.prepareStatement(sql);

// Set the ID parameter

// Replace with your custom ID

statement.setInt(1, customerId1);

// Execute the SQL query

int rowsAffected = statement.executeUpdate();

// Check the number of rows affected

if (rowsAffected > 0) {

System.out.println("Data deleted successfully.");

} else {

System.out.println("No data found to delete.");

}

// Close the statement and connection

statement.close();

}

connection.close();

}

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 3:

// View Single Record logic

System.out.println("Enter Appointment ID to show: ");

int customerId1 = scanner.nextInt();

scanner.nextLine();

try(Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo", "root", "jayant")){

// Step 3: Check if customer ID exists in the database

String checkQuery = "SELECT \* FROM appointment WHERE appid = ?";

try (PreparedStatement checkStatement = connection.prepareStatement(checkQuery)) {

checkStatement.setInt(1, customerId1);

ResultSet resultSet = checkStatement.executeQuery();

if (!resultSet.next()) {

System.out.println("Appointment does not exist in the database.");

break;

}else {

int id1 = resultSet.getInt("appid");

String name1 = resultSet.getString("custid");

String address1 = resultSet.getString("advid");

Date myDate2=resultSet.getDate("appdate");

// String email = resultSet.getString("email");

// ... Retrieve other columns

System.out.println("ID: " + id1);

System.out.println("Customer ID: " + name1);

System.out.println("Advocate ID: " + address1);

System.out.println("Date:" + myDate2);

System.out.println("======================");

}} catch (SQLException e) {

e.printStackTrace();

}

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 4:

// View All Records logic

String jdbcUrl1 = "jdbc:mysql://localhost:3306/demo";

String username1 = "root";

String password1 = "jayant";

try {

// Establish the database connection

Connection connection = DriverManager.getConnection(jdbcUrl1, username1, password1);

String query = "SELECT \* FROM appointment";

// Create a statement object

try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {

// Execute the query

try (ResultSet resultSet = preparedStatement.executeQuery()) {

// Process the result set

while (resultSet.next()) {

int id1 = resultSet.getInt("appid");

String name1 = resultSet.getString("custid");

String address1 = resultSet.getString("advid");

Date myDate2=resultSet.getDate("appdate");

// String email = resultSet.getString("email");

// ... Retrieve other columns

System.out.println("ID: " + id1);

System.out.println("Customer ID: " + name1);

System.out.println("Advocate ID: " + address1);

System.out.println("Date:" + myDate2);

System.out.println("======================");

}

}

} // PreparedStatement and ResultSet are automatically closed here

// Close the connection

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

break;

case 5:

System.out.println("Exiting Advocate Menu...");

choice=0;

break;

default:

System.out.println("Invalid choice. Please try again.");

break;

}

} while (choice != 0);

}

private static void handleServiceMenu(Scanner scanner) {

// Implement the logic for handling the Service menu

}

}

TEST SCREENSHOT:



