**EXPENSE MANAGEMENT SYSTEM**

**Mini-Project Report of**

**Object Oriented Programming through Java**

***Submitted by***

**Aesha Parikh-B014**

***Under the Guidance Of***

**PROF. SAHIL MEHTA**

***In partial fulfillment for the award of the degree of***

**B Tech.**

**COMPUTER ENGINEERING**

**At**



**MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING**

NMIMS (Deemed –to-be University)

JVPD Scheme Bhaktivedanta Swami Marg,

Ville Parle (W), Mumbai-400 056.

**2023-2024**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Page No.** |
|  |  |  |
| 1 | Problem Statement and Scope |  |
| 2 | Project Synopsis |  |
| 3 | Implementation (Source Code) |  |
| 4 | Results |  |
| 5 | Conclusion |  |
|  | References |  |

*PROBLEM STATEMENT AND SCOPE*

Problem Statement: You have developed an Expense Tracker System that allows users to register, login, create trips, add members to trips, add expenses to trips, and calculate the total expenses for each trip. The system also allows users to display the members of a specific trip.

Scope: The scope of the Expense Tracker System includes:

User Management:

* Registration: Users can register with a username and password.
* Login: Registered users can log in using their credentials.

Trip Management:

* Create Trip: Users can create a new trip by specifying a trip name.
* Add Members to Trip: Users can add members to an existing trip by providing the trip name and the number of members to add.
* Display Members: Users can view the members of a specific trip by entering the trip name.

Expense Management:

* Add Expense: Users can add expenses to a trip by specifying the trip name and the amount spent.
* Calculate Total Expense: Users can calculate the total expenses for a specific trip.

Database Management:

* Connection to Database: The system connects to a MySQL database to store user information, trip details, and expenses.
* Data Integrity: The system ensures data integrity by properly managing database transactions and handling exceptions.

Graphical User Interface (GUI):

* The system provides a user-friendly GUI using Java Swing components for interaction with users.
* GUI elements include buttons, text fields, labels, and frames for registration, login, and various functionalities.

Error Handling:

* The system handles errors gracefully by displaying appropriate error messages to users.
* Error scenarios include database connection failures, SQL errors, invalid input, and trip not found.

Security:

* User passwords are stored securely using Java's built-in security features.
* Access to trip details and expenses is restricted based on user authentication.

Extensibility:

* The system is designed to be easily extendable to incorporate additional features such as editing/deleting trips, modifying expenses, and generating reports.

Overall, the Expense Tracker System provides users with a convenient way to manage their expenses during trips, ensuring accountability and transparency among trip members.

*PROJECT SYNOPSIS*

Introduction:

Expense Manager is a comprehensive software solution designed to simplify and streamline expense management for individuals and groups, particularly during trips or group outings. Whether it's a vacation with friends or a business trip with colleagues, Expense Manager provides the necessary tools to track expenses efficiently, ensuring transparency and accuracy throughout the process.

Key Features:

Expense Manager offers a wide range of features to streamline expense management for individuals and groups alike. It includes a secure user authentication system with password encryption for safeguarding credentials. Users can create new trips, add members to trips, and easily view trip members for better organization. The system facilitates expense tracking by allowing users to record expenses incurred during trips, add expenses for specific trips, and calculate total expenses for each trip. Expense Manager integrates seamlessly with a relational database for data storage, ensuring data integrity and consistency. Its intuitive graphical user interface (GUI) provides easy navigation and includes well-designed forms and dialogs for inputting data. Comprehensive error handling and validation mechanisms guide users and prevent data corruption, while reporting and analytics functionalities generate detailed expense reports and provide insights into spending patterns and trends. With robust security measures in place, Expense Manager protects user data and implements access control mechanisms to restrict unauthorized access.

Project Scope:

Expense Manager aims to address the diverse needs and requirements of users involved in planning trips or group outings requiring expense tracking. It caters to individuals seeking a simple and efficient tool to manage expenses collaboratively, as well as organizations or teams looking for a cost-effective solution to track business trip expenses. Additionally, Expense Manager serves travel enthusiasts desiring transparency and accountability in expense management. The system provides a centralized platform for users to organize and track expenses seamlessly, fostering collaboration and accountability among trip members. With its comprehensive features and user-friendly interface, Expense Manager streamlines the expense management process, saving time and effort for users.

Conclusion:

Expense Manager offers a comprehensive solution for efficient expense management during trips or group outings. By combining user-friendly design, robust functionality, and secure data management, Expense Manager empowers users to take control of their expenses with confidence and ease. Whether it's planning a vacation or managing business travel, Expense Manager serves as the ultimate tool for hassle-free expense tracking, ensuring transparency, accuracy, and accountability throughout the process.

*IMPLEMENTATION (SOURCE CODE)*

package project;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.\*;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

class User {

private String username;

private String password;

public User(String username, String password) {

this.username = username;

this.password = password;

}

public String getUsername() {

return username;

}

public String getPassword() {

return password;

}

}

class Trip {

private String tripName;

private List<User> members;

private Map<User, Double> expenses;

public Trip(String tripName) {

this.tripName = tripName;

members = new ArrayList<>();

expenses = new HashMap<>();

}

public String getTripName() {

return tripName;

}

public List<User> getMembers() {

return members;

}

public void addMember(User user) {

members.add(user);

expenses.put(user, 0.0);

}

public void addExpense(User user, double amount) {

if (expenses.containsKey(user)) {

double currentExpense = expenses.get(user);

expenses.put(user, currentExpense + amount);

} else {

System.out.println("User is not a member of this trip.");

}

}

public double getTotalExpense() {

double total = 0.0;

for (double expense : expenses.values()) {

total += expense;

}

return total;

}

public double getExpensePerMember() {

return getTotalExpense() / members.size();

}

}

public class Main extends JFrame {

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

private static final String DB\_USER = "root";

private static final String DB\_PASSWORD = "4@Komalparag";

private static Connection connection = null;

private static PreparedStatement preparedStatement = null;

private static ResultSet resultSet = null;

private static User currentUser;

private JPanel mainPanel;

private JButton registerButton;

private JButton loginButton;

private JButton exitButton;

private JLabel systemLabel;

private JLabel titleLabel;

private JLabel developerLabel;

public Main() {

setTitle("\uD83D\uDCB8 Expense Tracker System \uD83D\uDCB8");

setSize(800, 600);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setLayout(new BorderLayout());

Font titleFont = new Font("Arial", Font.BOLD | Font.ITALIC, 28);

Color titleColor = new Color(0, 0, 0);

Color backgroundColor = new Color(135, 206, 250);

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(5, 1));

panel.setBackground(backgroundColor);

systemLabel = new JLabel("Expense Tracker System");

systemLabel.setHorizontalAlignment(SwingConstants.CENTER);

systemLabel.setFont(titleFont);

systemLabel.setForeground(titleColor);

panel.add(systemLabel);

Font buttonFont = new Font("Arial", Font.BOLD, 18);

loginButton = new JButton("Login");

loginButton.setFont(buttonFont);

loginButton.setPreferredSize(new Dimension(200, 50));

loginButton.setBackground(Color.LIGHT\_GRAY);

loginButton.setForeground(titleColor);

loginButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

login();

}

});

panel.add(loginButton);

registerButton = new JButton("Register");

registerButton.setFont(buttonFont);

registerButton.setPreferredSize(new Dimension(200, 50));

registerButton.setBackground(Color.LIGHT\_GRAY);

registerButton.setForeground(titleColor);

registerButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

register();

}

});

panel.add(registerButton);

exitButton = new JButton("Exit");

exitButton.setFont(buttonFont);

exitButton.setPreferredSize(new Dimension(200, 50));

exitButton.setBackground(Color.LIGHT\_GRAY);

exitButton.setForeground(titleColor);

exitButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

closeDatabaseResources();

System.exit(0);

}

});

panel.add(exitButton);

add(panel, BorderLayout.CENTER);

developerLabel = new JLabel("Developed by Aesha,Aakash,Siddhesh");

developerLabel.setFont(new Font("Arial",Font.ITALIC,20));

developerLabel.setHorizontalAlignment(SwingConstants.CENTER);

add(developerLabel, BorderLayout.SOUTH);

connectToDatabase();

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

new Main().setVisible(true);

}

});

}

private static void connectToDatabase() {

try {

connection = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

} catch (SQLException e) {

System.out.println("Error connecting to database: " + e.getMessage());

System.exit(1);

}

}

private static void closeDatabaseResources() {

try {

if (resultSet != null) resultSet.close();

if (preparedStatement != null) preparedStatement.close();

if (connection != null) connection.close();

} catch (SQLException e) {

System.out.println("Error closing database resources: " + e.getMessage());

}

}

private void register() {

JFrame registerFrame = new JFrame("Register");

registerFrame.setSize(800, 600);

registerFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

registerFrame.setLocationRelativeTo(null);

registerFrame.setLayout(new BorderLayout());

registerFrame.getContentPane().setBackground(new Color(135, 206, 250));

JPanel registerPanel = new JPanel();

registerPanel.setLayout(new GridLayout(5, 1));

registerPanel.setBackground(new Color(135, 206, 250));

JLabel headingLabel = new JLabel("Register");

headingLabel.setHorizontalAlignment(SwingConstants.CENTER);

headingLabel.setFont(new Font("Verdana", Font.ITALIC, 27));

registerPanel.add(headingLabel);

JPanel usernamePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

usernamePanel.setBackground(new Color(135, 206, 250));

JLabel usernameLabel = new JLabel("Username:");

usernameLabel.setFont(new Font("Arial",Font.ITALIC,21));

JTextField usernameField = new JTextField(15);

usernameField.setBackground(Color.LIGHT\_GRAY);

usernamePanel.add(usernameLabel);

usernamePanel.add(usernameField);

registerPanel.add(usernamePanel);

JPanel passwordPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

passwordPanel.setBackground(new Color(135, 206, 250));

JLabel passwordLabel = new JLabel("Password:");

passwordLabel.setFont(new Font("Arial",Font.ITALIC,21));

JPasswordField passwordField = new JPasswordField(15);

passwordField.setBackground(Color.LIGHT\_GRAY);

passwordPanel.add(passwordLabel);

passwordPanel.add(passwordField);

registerPanel.add(passwordPanel);

JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

buttonPanel.setBackground(new Color(135, 206, 250));

JButton registerButton = new JButton("Register");

registerButton.setBackground(Color.WHITE);

registerButton.setPreferredSize(new Dimension(100, 40));

buttonPanel.add(registerButton);

registerPanel.add(buttonPanel);

registerFrame.add(registerPanel, BorderLayout.CENTER);

registerButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String username = usernameField.getText().trim();

String password = new String(passwordField.getPassword());

try {

String query = "INSERT INTO users (username, password) VALUES (?, ?)";

preparedStatement = connection.prepareStatement(query);

preparedStatement.setString(1, username);

preparedStatement.setString(2, password);

preparedStatement.executeUpdate();

JOptionPane.showMessageDialog(null, "Registration successful!");

registerFrame.dispose();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "Error registering user: " + ex.getMessage());

}

}

});

registerFrame.setVisible(true);

}

private void login() {

JFrame loginFrame = new JFrame("Login");

loginFrame.setSize(800, 600);

loginFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

loginFrame.setLocationRelativeTo(null);

loginFrame.setLayout(new BorderLayout());

loginFrame.getContentPane().setBackground(new Color(135, 206, 250));

JPanel loginPanel = new JPanel();

loginPanel.setLayout(new GridLayout(5, 1));

loginPanel.setBackground(new Color(135, 206, 250));

JLabel headingLabel = new JLabel("Login");

headingLabel.setHorizontalAlignment(SwingConstants.CENTER);

headingLabel.setFont(new Font("Verdana", Font.ITALIC, 27));

loginPanel.add(headingLabel);

JPanel usernamePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

usernamePanel.setBackground(new Color(135, 206, 250));

JLabel usernameLabel = new JLabel("Username:");

usernameLabel.setFont(new Font("Arial",Font.ITALIC,21));

JTextField usernameField = new JTextField(15);

usernameField.setBackground(Color.LIGHT\_GRAY);

usernamePanel.add(usernameLabel);

usernamePanel.add(usernameField);

loginPanel.add(usernamePanel);

JPanel passwordPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

passwordPanel.setBackground(new Color(135, 206, 250));

JLabel passwordLabel = new JLabel("Password:");

passwordLabel.setFont(new Font("Arial",Font.ITALIC,21));

JPasswordField passwordField = new JPasswordField(15);

passwordField.setBackground(Color.LIGHT\_GRAY);

passwordPanel.add(passwordLabel);

passwordPanel.add(passwordField);

loginPanel.add(passwordPanel);

JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

buttonPanel.setBackground(new Color(135, 206, 250));

JButton loginButton = new JButton("Login");

loginButton.setBackground(Color.WHITE);

loginButton.setPreferredSize(new Dimension(100, 40));

buttonPanel.add(loginButton);

loginPanel.add(buttonPanel);

loginFrame.add(loginPanel, BorderLayout.CENTER);

loginButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String username = usernameField.getText().trim();

String password = new String(passwordField.getPassword());

try {

String query = "SELECT \* FROM users WHERE username = ? AND password = ?";

preparedStatement = connection.prepareStatement(query);

preparedStatement.setString(1, username);

preparedStatement.setString(2, password);

resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

currentUser = new User(username, password);

JOptionPane.showMessageDialog(null, "Login successful!");

loggedInMenu();

loginFrame.dispose();

} else {

JOptionPane.showMessageDialog(null, "Invalid username or password. Please try again.");

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "Error logging in: " + ex.getMessage());

}

}

});

loginFrame.setVisible(true);

}

private void loggedInMenu() {

JFrame loggedInFrame = new JFrame("Logged In Menu");

loggedInFrame.setSize(800, 600);

loggedInFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

loggedInFrame.setLocationRelativeTo(null);

loggedInFrame.setLayout(new BorderLayout());

loggedInFrame.getContentPane().setBackground(new Color(135, 206, 250));

JPanel loggedInPanel = new JPanel();

loggedInPanel.setLayout(new GridLayout(8, 1));

loggedInPanel.setBackground(new Color(135, 206, 250));

loggedInPanel.add(Box.createVerticalGlue());

JLabel headingLabel = new JLabel("Logged In Menu");

headingLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

headingLabel.setAlignmentY(Component.CENTER\_ALIGNMENT);

headingLabel.setFont(new Font("Verdana", Font.ITALIC, 24));

headingLabel.setBorder(BorderFactory.createEmptyBorder(0, 300, 50, 0));

loggedInPanel.add(headingLabel);

JButton createTripButton = new JButton("Create Trip");

JButton addMembersButton = new JButton("Add Members to Trip");

JButton addExpenseButton = new JButton("Add Expense");

JButton calculateExpenseButton = new JButton("Calculate Total Expense");

JButton displayMembersButton = new JButton("Display Members");

JButton logoutButton = new JButton("Logout");

createTripButton.setFont(new Font("Arial", Font.BOLD, 16));

addMembersButton.setFont(new Font("Arial", Font.BOLD, 16));

addExpenseButton.setFont(new Font("Arial", Font.BOLD, 16));

calculateExpenseButton.setFont(new Font("Arial", Font.BOLD, 16));

displayMembersButton.setFont(new Font("Arial", Font.BOLD, 16));

logoutButton.setFont(new Font("Arial", Font.BOLD, 16));

Color buttonColor = Color.LIGHT\_GRAY;

createTripButton.setBackground(buttonColor);

addMembersButton.setBackground(buttonColor);

addExpenseButton.setBackground(buttonColor);

calculateExpenseButton.setBackground(buttonColor);

displayMembersButton.setBackground(buttonColor);

logoutButton.setBackground(buttonColor);

createTripButton.setForeground(Color.BLACK);

addMembersButton.setForeground(Color.BLACK);

addExpenseButton.setForeground(Color.BLACK);

calculateExpenseButton.setForeground(Color.BLACK);

displayMembersButton.setForeground(Color.BLACK);

logoutButton.setForeground(Color.BLACK);

loggedInPanel.add(createTripButton);

loggedInPanel.add(addMembersButton);

loggedInPanel.add(addExpenseButton);

loggedInPanel.add(calculateExpenseButton);

loggedInPanel.add(displayMembersButton);

loggedInPanel.add(logoutButton);

loggedInFrame.add(loggedInPanel, BorderLayout.CENTER);

createTripButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

createTrip();

}

});

addMembersButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

addMembersToTrip();

}

});

addExpenseButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

addExpense();

}

});

calculateExpenseButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

calculateTotalExpense();

}

});

displayMembersButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String tripName = JOptionPane.showInputDialog(loggedInFrame, "Enter the trip name:");

if (tripName != null && !tripName.isEmpty()) {

displayTripMembers(tripName);

}

}

});

logoutButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

currentUser = null;

loggedInFrame.dispose();

}

});

loggedInFrame.setVisible(true);

}

private void displayTripMembers(String tripName) {

JFrame displayMembersFrame = new JFrame("Trip Members");

displayMembersFrame.setSize(800, 600);

displayMembersFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

displayMembersFrame.setLocationRelativeTo(null);

JPanel displayMembersPanel = new JPanel(new BorderLayout());

displayMembersPanel.setBackground(new Color(135, 206, 250));

JLabel titleLabel = new JLabel("Members of Trip: " + tripName);

titleLabel.setFont(new Font("Verdana", Font.BOLD, 20));

titleLabel.setHorizontalAlignment(SwingConstants.CENTER);

displayMembersPanel.add(titleLabel, BorderLayout.NORTH);

JTextArea membersTextArea = new JTextArea();

membersTextArea.setEditable(false);

membersTextArea.setFont(new Font("Arial", Font.PLAIN, 20));

membersTextArea.setBackground(Color.LIGHT\_GRAY);

membersTextArea.setMargin(new Insets(10, 10, 10, 10));

try {

int tripId = getTripId(tripName);

if (tripId != -1) {

String query = "SELECT username FROM users INNER JOIN trip\_members ON users.user\_id = trip\_members.user\_id WHERE trip\_id = ?";

preparedStatement = connection.prepareStatement(query);

preparedStatement.setInt(1, tripId);

resultSet = preparedStatement.executeQuery();

StringBuilder membersText = new StringBuilder();

while (resultSet.next()) {

String username = resultSet.getString("username");

membersText.append("->").append(username).append("\n");

}

membersTextArea.setText(membersText.toString());

} else {

JOptionPane.showMessageDialog(null, "Trip not found.");

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Error retrieving trip members: " + e.getMessage());

}

JScrollPane scrollPane = new JScrollPane(membersTextArea);

displayMembersPanel.add(scrollPane, BorderLayout.CENTER);

displayMembersFrame.add(displayMembersPanel);

displayMembersFrame.setVisible(true);

}

private void createTrip() {

JFrame createTripFrame = new JFrame("Create a Trip");

createTripFrame.setSize(800, 600);

createTripFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

createTripFrame.setLocationRelativeTo(null);

createTripFrame.setLayout(new BorderLayout());

createTripFrame.getContentPane().setBackground(new Color(135, 206, 250));

JPanel createTripPanel = new JPanel();

createTripPanel.setLayout(new GridLayout(3, 1));

createTripPanel.setBackground(new Color(135, 206, 250));

JLabel headingLabel = new JLabel("Create a Trip");

headingLabel.setHorizontalAlignment(SwingConstants.CENTER);

headingLabel.setFont(new Font("Verdana", Font.ITALIC, 27));

createTripPanel.add(headingLabel);

JPanel tripNamePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

tripNamePanel.setBackground(new Color(135, 206, 250));

JLabel tripNameLabel = new JLabel("Trip Name:");

tripNameLabel.setFont(new Font("Arial",Font.ITALIC,21));

JTextField tripNameField = new JTextField(20);

tripNameField.setBackground(Color.LIGHT\_GRAY);

tripNamePanel.add(tripNameLabel);

tripNamePanel.add(tripNameField);

createTripPanel.add(tripNamePanel);

JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

buttonPanel.setBackground(new Color(135, 206, 250));

JButton createTripButton = new JButton("Create Trip");

createTripButton.setBackground(Color.WHITE);

createTripButton.setPreferredSize(new Dimension(150, 40));

buttonPanel.add(createTripButton);

createTripPanel.add(buttonPanel);

createTripFrame.add(createTripPanel, BorderLayout.CENTER);

createTripButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String tripName = tripNameField.getText().trim();

try {

String query = "INSERT INTO trips (trip\_name) VALUES (?)";

preparedStatement = connection.prepareStatement(query, Statement.RETURN\_GENERATED\_KEYS);

preparedStatement.setString(1, tripName);

preparedStatement.executeUpdate();

ResultSet rs = preparedStatement.getGeneratedKeys();

int tripId = -1;

if (rs.next()) {

tripId = rs.getInt(1);

}

if (tripId != -1) {

JOptionPane.showMessageDialog(null, "Trip created successfully.");

addMemberToTrip(tripId, currentUser.getUsername());

createTripFrame.dispose();

} else {

JOptionPane.showMessageDialog(null, "Failed to create trip.");

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "Error creating trip: " + ex.getMessage());

}

}

});

createTripFrame.setVisible(true);

}

private void addMembersToTrip() {

JFrame addMembersFrame = new JFrame("Add Members to Trip");

addMembersFrame.setSize(800, 600);

addMembersFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

addMembersFrame.setLocationRelativeTo(null);

addMembersFrame.setLayout(new BorderLayout());

addMembersFrame.getContentPane().setBackground(new Color(135, 206, 250));

JPanel addMembersPanel = new JPanel();

addMembersPanel.setLayout(new GridLayout(4, 1));

addMembersPanel.setBackground(new Color(135, 206, 250));

JLabel headingLabel = new JLabel("Add Members to Trip");

headingLabel.setHorizontalAlignment(SwingConstants.CENTER);

headingLabel.setFont(new Font("Verdana", Font.ITALIC, 27));

addMembersPanel.add(headingLabel);

JPanel inputPanel = new JPanel(new GridLayout(2, 2));

inputPanel.setBackground(new Color(135, 206, 250));

JLabel tripNameLabel = new JLabel("Trip Name:");

tripNameLabel.setFont(new Font("Arial",Font.ITALIC,21));

JTextField tripNameField = new JTextField(15);

JLabel numMembersLabel = new JLabel("Number of Members:");

numMembersLabel.setFont(new Font("Arial",Font.ITALIC,21));

JTextField numMembersField = new JTextField(5);

tripNameField.setBackground(Color.LIGHT\_GRAY);

numMembersField.setBackground(Color.LIGHT\_GRAY);

inputPanel.add(tripNameLabel);

inputPanel.add(tripNameField);

inputPanel.add(numMembersLabel);

inputPanel.add(numMembersField);

addMembersPanel.add(inputPanel);

JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

buttonPanel.setBackground(new Color(135, 206, 250));

JButton addMemberButton = new JButton("Add Members");

addMemberButton.setBackground(Color.WHITE);

addMemberButton.setPreferredSize(new Dimension(150, 40));

buttonPanel.add(addMemberButton);

addMembersPanel.add(buttonPanel);

addMembersFrame.add(addMembersPanel, BorderLayout.CENTER);

addMemberButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String tripName = tripNameField.getText().trim();

int tripId = getTripId(tripName);

if (tripId != -1) {

int numMembers = Integer.parseInt(numMembersField.getText().trim());

for (int i = 0; i < numMembers; i++) {

String memberUsername = JOptionPane.showInputDialog(addMembersFrame, "Enter member " + (i + 1) + " username:");

addMemberToTrip(tripId, memberUsername);

}

addMembersFrame.dispose();

} else {

JOptionPane.showMessageDialog(null, "Trip not found.");

}

}

});

addMembersFrame.setVisible(true);

}

private void addExpense() {

JFrame addExpenseFrame = new JFrame("Add Expense");

addExpenseFrame.setSize(800, 600);

addExpenseFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

addExpenseFrame.setLocationRelativeTo(null);

addExpenseFrame.setLayout(new BorderLayout());

addExpenseFrame.getContentPane().setBackground(new Color(135, 206, 250));

JPanel addExpensePanel = new JPanel();

addExpensePanel.setLayout(new GridLayout(4, 1));

addExpensePanel.setBackground(new Color(135, 206, 250));

JLabel headingLabel = new JLabel("Add Expense");

headingLabel.setHorizontalAlignment(SwingConstants.CENTER);

headingLabel.setFont(new Font("Verdana", Font.ITALIC, 27));

addExpensePanel.add(headingLabel);

JPanel tripNamePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

tripNamePanel.setBackground(new Color(135, 206, 250));

JLabel tripNameLabel = new JLabel("Trip Name:");

tripNameLabel.setFont(new Font("Arial",Font.ITALIC,21));

JTextField tripNameField = new JTextField(20);

tripNameField.setBackground(Color.LIGHT\_GRAY);

tripNamePanel.add(tripNameLabel);

tripNamePanel.add(tripNameField);

addExpensePanel.add(tripNamePanel);

JPanel amountPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

amountPanel.setBackground(new Color(135, 206, 250));

JLabel amountLabel = new JLabel("Amount Spent:");

amountLabel.setFont(new Font("Arial",Font.ITALIC,21));

JTextField amountField = new JTextField(20);

amountField.setBackground(Color.LIGHT\_GRAY);

amountPanel.add(amountLabel);

amountPanel.add(amountField);

addExpensePanel.add(amountPanel);

JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

buttonPanel.setBackground(new Color(135, 206, 250));

JButton addExpenseButton = new JButton("Add Expense");

addExpenseButton.setBackground(Color.WHITE);

addExpenseButton.setPreferredSize(new Dimension(150, 40));

buttonPanel.add(addExpenseButton);

addExpensePanel.add(buttonPanel);

addExpenseFrame.add(addExpensePanel, BorderLayout.CENTER);

addExpenseButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String tripName = tripNameField.getText().trim();

double amount = Double.parseDouble(amountField.getText().trim());

int tripId = getTripId(tripName);

if (tripId != -1) {

try {

String query = "INSERT INTO expenses (trip\_id, amount) VALUES (?, ?)";

preparedStatement = connection.prepareStatement(query);

preparedStatement.setInt(1, tripId);

preparedStatement.setDouble(2, amount);

preparedStatement.executeUpdate();

JOptionPane.showMessageDialog(null, "Expense added successfully.");

addExpenseFrame.dispose();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "Error adding expense: " + ex.getMessage());

}

} else {

JOptionPane.showMessageDialog(null, "Trip not found.");

}

}

});

addExpenseFrame.setVisible(true);

}

private void calculateTotalExpense() {

JFrame calculateExpenseFrame = new JFrame("Calculate Total Expense");

calculateExpenseFrame.setSize(800, 600); // Adjusted window size

calculateExpenseFrame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

calculateExpenseFrame.setLocationRelativeTo(null);

calculateExpenseFrame.setLayout(new BorderLayout());

calculateExpenseFrame.getContentPane().setBackground(new Color(135, 206, 250));

JPanel calculateExpensePanel = new JPanel();

calculateExpensePanel.setLayout(new GridLayout(3, 1));

calculateExpensePanel.setBackground(new Color(135, 206, 250));

JLabel headingLabel = new JLabel("Calculate Total Expense");

headingLabel.setHorizontalAlignment(SwingConstants.CENTER);

headingLabel.setFont(new Font("Verdana", Font.ITALIC, 27));

calculateExpensePanel.add(headingLabel);

JPanel tripNamePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

tripNamePanel.setBackground(new Color(135, 206, 250));

JLabel tripNameLabel = new JLabel("Trip Name:");

tripNameLabel.setFont(new Font("Arial",Font.ITALIC,21));

JTextField tripNameField = new JTextField(20);

tripNameField.setBackground(Color.LIGHT\_GRAY);

tripNamePanel.add(tripNameLabel);

tripNamePanel.add(tripNameField);

calculateExpensePanel.add(tripNamePanel);

JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));

buttonPanel.setBackground(new Color(135, 206, 250));

JButton calculateButton = new JButton("Calculate");

calculateButton.setBackground(Color.WHITE);

calculateButton.setPreferredSize(new Dimension(150, 40));

buttonPanel.add(calculateButton);

calculateExpensePanel.add(buttonPanel);

calculateExpenseFrame.add(calculateExpensePanel, BorderLayout.CENTER);

calculateButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String tripName = tripNameField.getText().trim();

int tripId = getTripId(tripName);

if (tripId != -1) {

try {

String query = "SELECT COUNT(\*) AS num\_members FROM trip\_members WHERE trip\_id = ?";

preparedStatement = connection.prepareStatement(query);

preparedStatement.setInt(1, tripId);

resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

int numMembers = resultSet.getInt("num\_members");

query = "SELECT SUM(amount) AS total\_expense FROM expenses WHERE trip\_id = ?";

preparedStatement = connection.prepareStatement(query);

preparedStatement.setInt(1, tripId);

resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

double totalExpense = resultSet.getDouble("total\_expense");

double expensePerMember = totalExpense / numMembers;

JOptionPane.showMessageDialog(null, "Total expense for trip '" + tripName + "': $" + totalExpense + "\nExpense per member: $" + expensePerMember);

calculateExpenseFrame.dispose();

} else {

JOptionPane.showMessageDialog(null, "No expenses found for this trip.");

}

} else {

JOptionPane.showMessageDialog(null, "No members found for this trip.");

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "Error calculating total expense: " + ex.getMessage());

}

} else {

JOptionPane.showMessageDialog(null, "Trip not found.");

}

}

});

calculateExpenseFrame.setVisible(true);

}

private static void addMemberToTrip(int tripId, String username) {

try {

String query = "SELECT user\_id FROM users WHERE username = ?";

preparedStatement = connection.prepareStatement(query);

preparedStatement.setString(1, username);

resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

int userId = resultSet.getInt("user\_id");

query = "INSERT INTO trip\_members (trip\_id, user\_id) VALUES (?, ?)";

preparedStatement = connection.prepareStatement(query);

preparedStatement.setInt(1, tripId);

preparedStatement.setInt(2, userId);

preparedStatement.executeUpdate();

JOptionPane.showMessageDialog(null, "Member added to trip successfully.");

} else {

JOptionPane.showMessageDialog(null, "Username not found.", "Error", JOptionPane.ERROR\_MESSAGE);

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Error adding member to trip: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private static int getTripId(String tripName) {

try {

String query = "SELECT trip\_id FROM trips WHERE trip\_name = ?";

preparedStatement = connection.prepareStatement(query);

preparedStatement.setString(1, tripName);

resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

return resultSet.getInt("trip\_id");

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Error getting trip ID: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

return -1;

}

private static int getUserId(String username) {

try {

String query = "SELECT user\_id FROM users WHERE username = ?";

preparedStatement = connection.prepareStatement(query);

preparedStatement.setString(1, username);

resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

return resultSet.getInt("user\_id");

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Error getting user ID: " + e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

return -1;

}

}

***MYSQL CODE***

CREATE DATABASE trip\_expenses;

USE trip\_expenses;

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) UNIQUE NOT NULL,

password VARCHAR(50) NOT NULL

);

CREATE TABLE trips (

trip\_id INT AUTO\_INCREMENT PRIMARY KEY,

trip\_name VARCHAR(100) UNIQUE NOT NULL

);

CREATE TABLE trip\_members (

trip\_member\_id INT AUTO\_INCREMENT PRIMARY KEY,

trip\_id INT,

user\_id INT,

FOREIGN KEY (trip\_id) REFERENCES trips(trip\_id),

FOREIGN KEY (user\_id) REFERENCES users(user\_id)

);

CREATE TABLE expenses (

expense\_id INT AUTO\_INCREMENT PRIMARY KEY,

trip\_id INT,

user\_id INT,

amount DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (trip\_id) REFERENCES trips(trip\_id),

FOREIGN KEY (user\_id) REFERENCES users(user\_id)

);

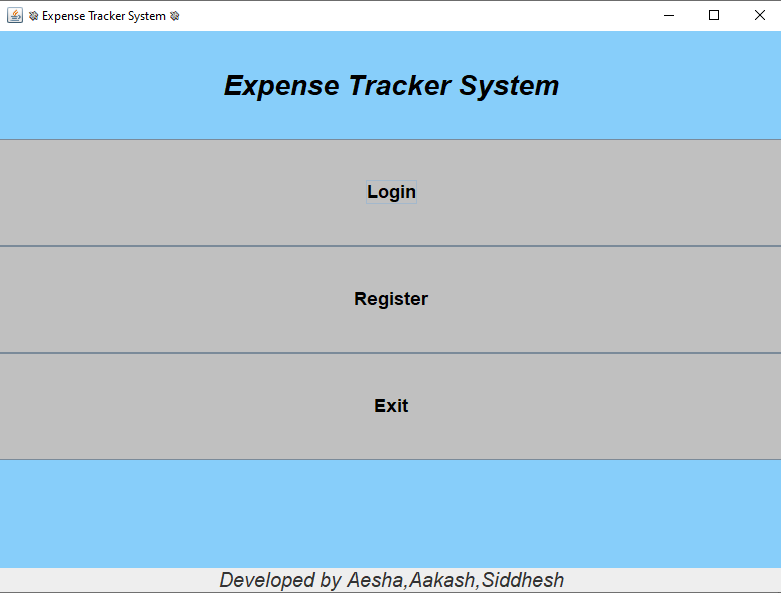
select \* from users;

select \* from trips;

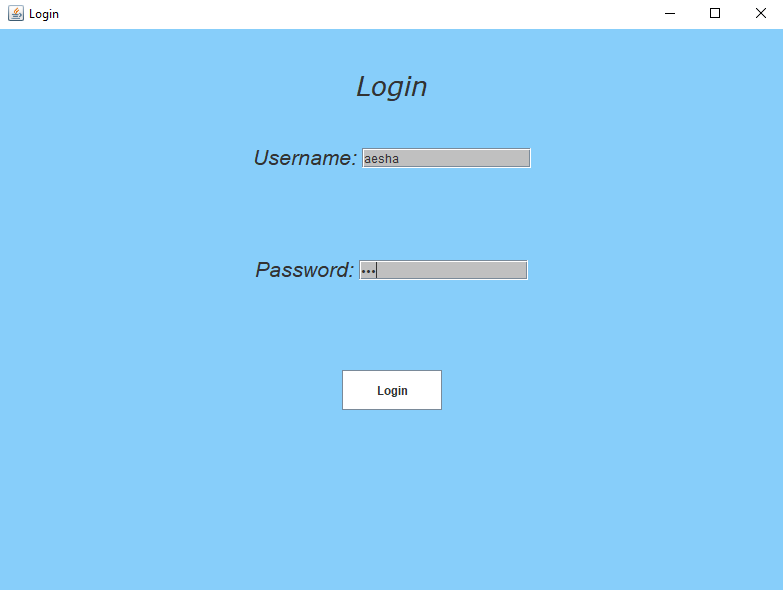
select \* from expenses;

*RESULTS*

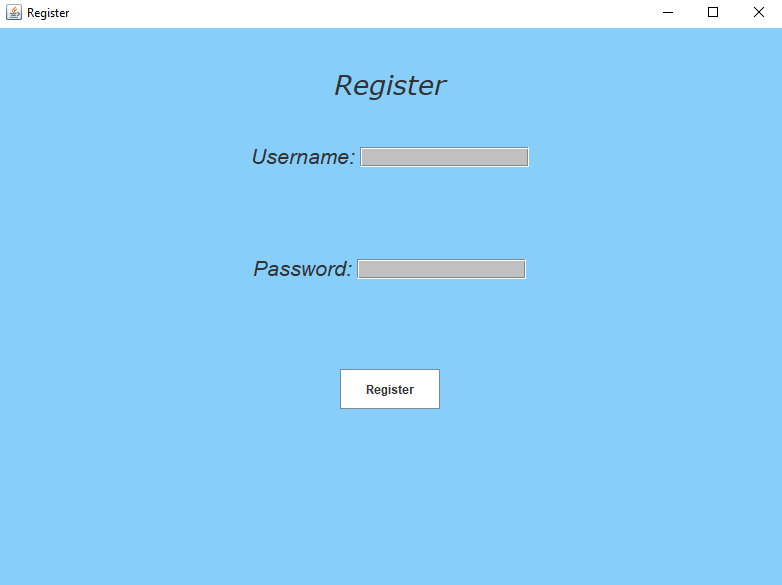
1. Homepage:



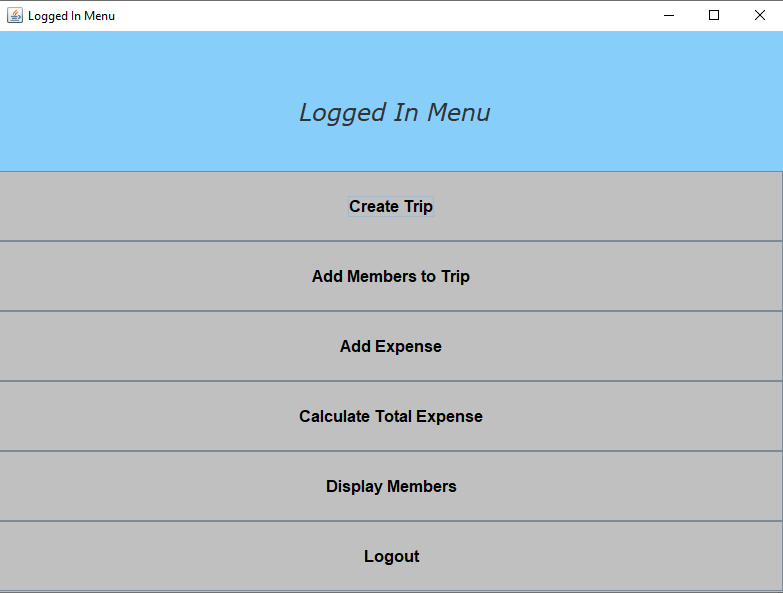
1. Login page:



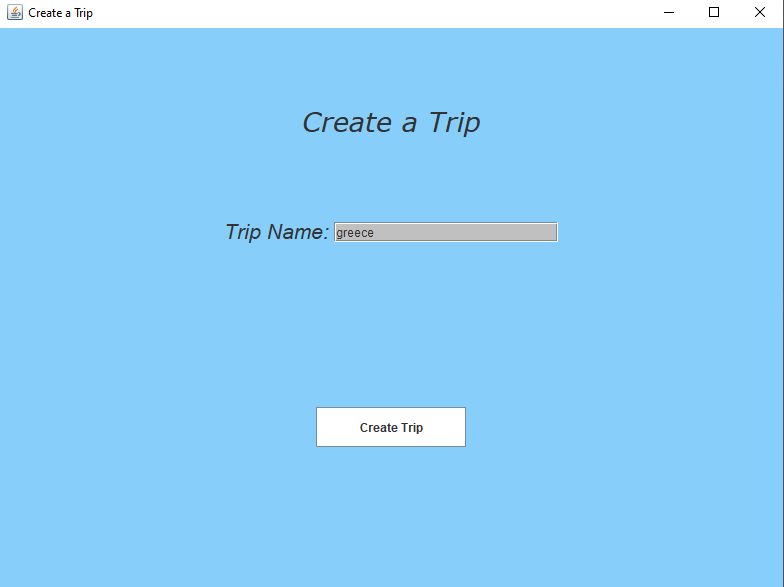
1. Register page:



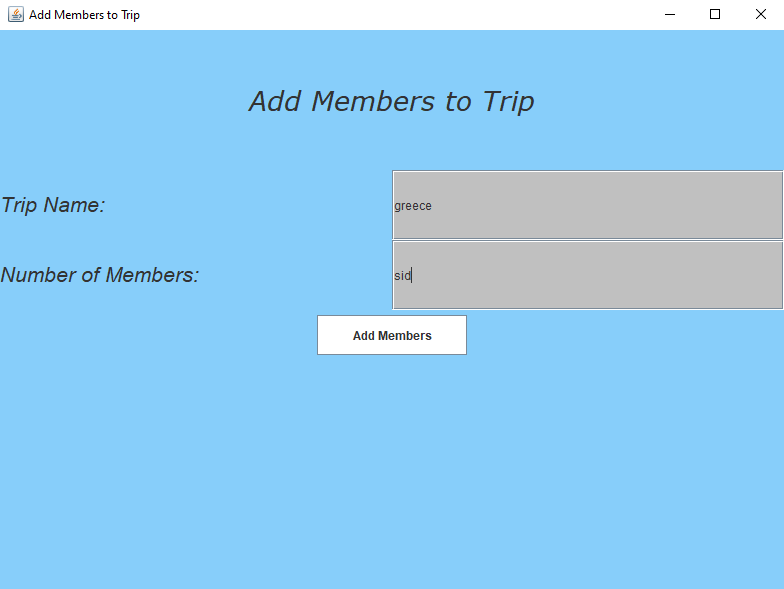
1. Logged in menu:



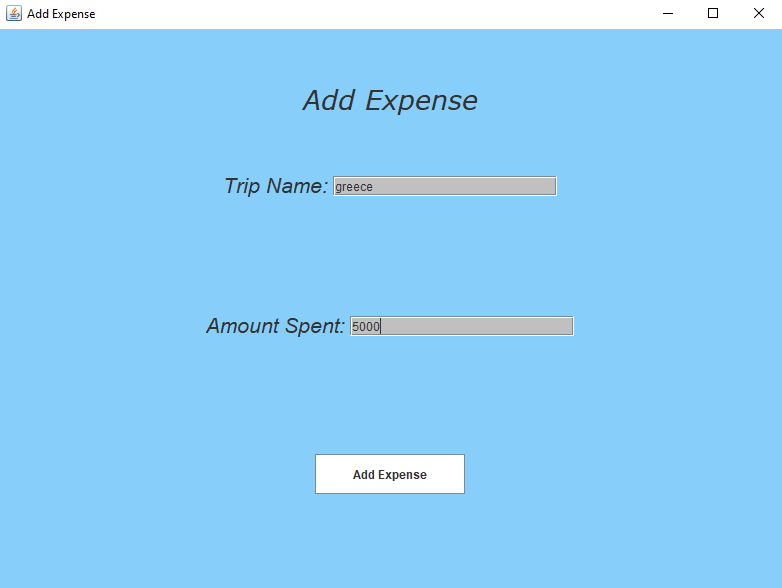
1. Create Trip:



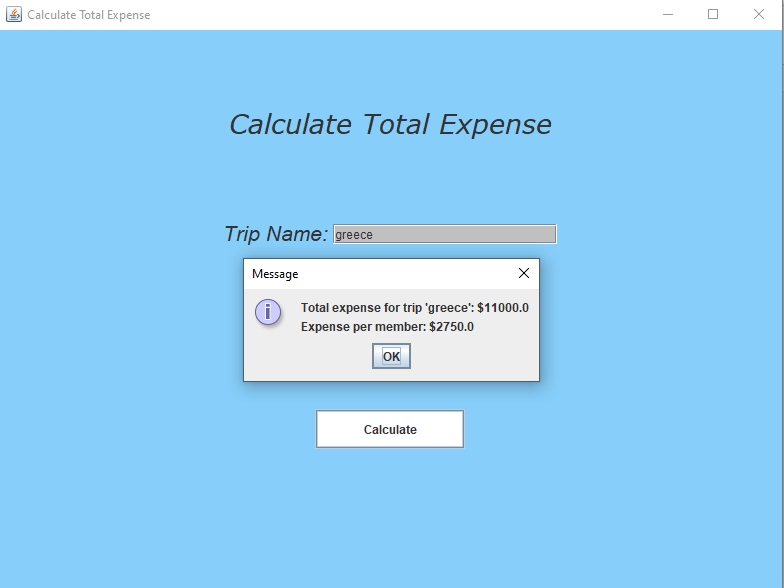
1. Add members to trip:



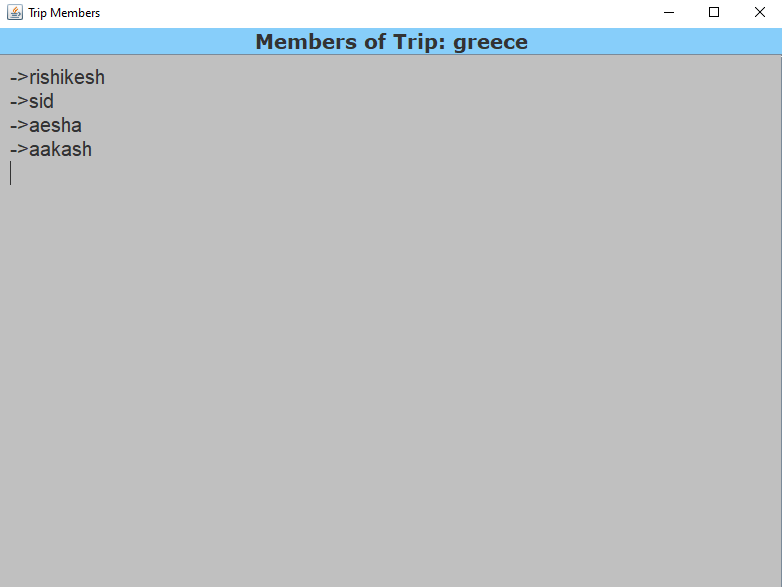
1. Add expense page:



1. Calculate total expense:



1. Display Members:



*CONCLUSION*

Expense Manager is a powerful tool that addresses the complex challenges of expense management during trips or group outings. With its user-friendly design, robust functionality, and secure data management, Expense Manager empowers individuals and groups to track expenses with confidence and ease. By providing features such as secure user authentication, intuitive trip management, detailed expense tracking, and comprehensive reporting, Expense Manager ensures transparency, accuracy, and accountability throughout the expense management process. Whether it's planning a vacation, organizing a group outing, or managing business travel, Expense Manager streamlines the process, saving time and effort for users. With its commitment to excellence in expense management, Expense Manager stands as the ultimate solution for hassle-free expense tracking, enabling users to make informed decisions and effectively manage their finances during trips or group activities.

*REFERENCES*

* <https://www.geeksforgeeks.org/data-structures/>
* <https://www.tutorialspoint.com/java_data_structures/index.htm>
* <https://in.video.search.yahoo.com/search/video;_ylt=AwrKB0hXtABm7EoMLES7HAx.;_ylu=Y29sbwNzZzMEcG9zAzEEdnRpZAMEc2VjA3Nj?p=database+connectivity+mysql+with+java+netbeans&vm=r&type=C211IN826D20150314&ei=UTF-8&fr=mcafee&turl=https%3A%2F%2Ftse4.mm.bing.net%2Fth%3Fid%3DOVP.A8yDWe-mbqUU9LZLiAhSewEsDh%26pid%3DApi%26w%3D296%26h%3D156%26c%3D7%26p%3D0&rurl=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DzerYyyJ9CRs&tit=Java+Database+Connectivity+%7C%7C+How+to+connect+MYSQL+8.0.33+and+NetBeans+%7C%7C+%23jdbc+%23jdbctutorial&pos=1&vid=631253d386160440555353f50a7081a4&sigr=I1B8JwFi9_yT&sigt=ewHHH1PzDGCu&sigi=dqcyG2q3cJC>.
* <https://r.search.yahoo.com/_ylt=AwrKB0hXtABm7EoMOUS7HAx.;_ylu=Y29sbwNzZzMEcG9zAzEEdnRpZAMEc2VjA3Ny/RV=2/RE=1711350999/RO=10/RU=https%3a%2f%2fwww.tutorialsfield.com%2fhow-to-connect-mysql-database-in-java-using-netbeans%2f/RK=2/RS=gbyD5Urz4_TIt8hS.R.tF.8lSLc->
* <https://docs.oracle.com/javase/tutorial/uiswing/learn/index.html>
* <https://r.search.yahoo.com/_ylt=Awr1QTyitABmzp8MtSq7HAx.;_ylu=Y29sbwNzZzMEcG9zAzEEdnRpZAMEc2VjA3Ny/RV=2/RE=1711351074/RO=10/RU=https%3a%2f%2fwww.javatpoint.com%2fjava-swing/RK=2/RS=iO52eVWnsEPtrL5_ib9uweV52UY->