TIPMATE:

TIP TRACKER

2022 - 2023

ABSTRACT

This report is a fragment of the assignments that have to be submitted to Dr. Mayank Sohani for partial fulfilment of Internal Continuous Assessments (ICA), as per the course policy. The Java GUI project aims to develop a user-friendly and efficient graphical user interface for a tip calculator. The project involves the implementation of a login and signup page for user authentication, as well as a database connectivity feature to store user data securely.

Prepared by:
Daksh Gehlot – N230 | 70472100206

Submitted to:
Dr. Mayank Sohani

Assistant Professor

SVKM's NMIMS MPSTME, Shirpur

NMIMS, MPSTME, Shirpur Campus		
Computer Engineering	**Class: -** B Tech/ MBA Tech (CE) **Semester:** - IV Sem.	
Subject: - Object Oriented Programming through JAVA	Academic Year: - 2022-23	

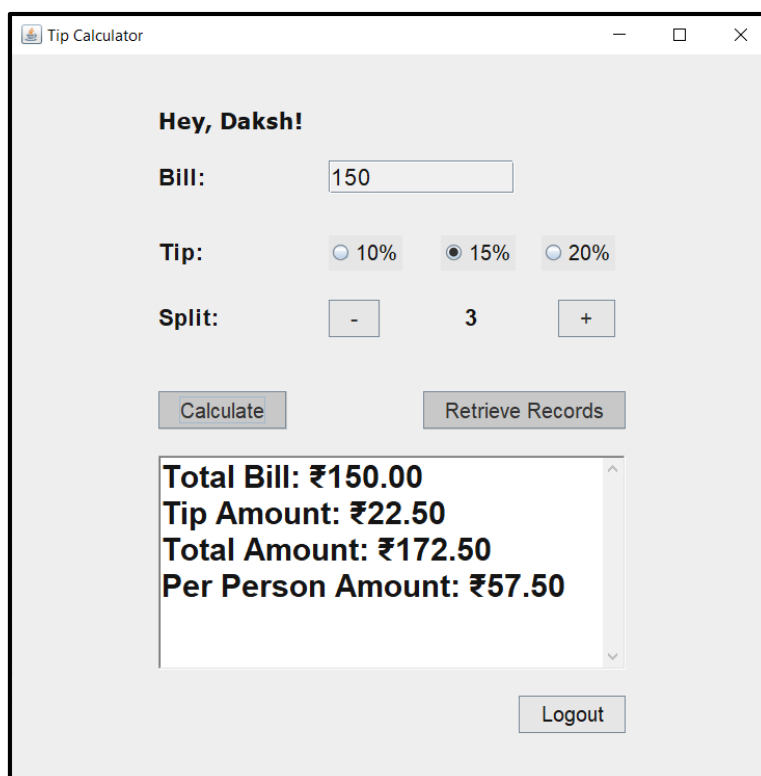# Index

INTRODUCTION

This report is based on the project of developing a Tip Calculator application with Login and Signup features. The project consists of multiple pages, including the Tip Calculator page, Login page, and Signup page. The Tip Calculator page is the main page of the application, where users can input their bill amount and calculate the tip amount based on the tip percentage they choose. In addition to that, the project also features a database connectivity feature that stores user data in a MySQL database and a split feature that allows users to split the bill amount with multiple people. The Login and Signup pages allow users to create an account, sign in, and save their calculation history.

The main objective of this project is to develop an efficient and user-friendly Tip Calculator application that can calculate tips with ease and accuracy. In this report, we will discuss the design, development, and testing of the application. Additionally, we will provide a detailed description of the functionalities and features of each page of the application.

NMIMS, MPSTME, Shirpur Campus		
Computer Engineering	**Class: -** B Tech/ MBA Tech (CE) **Semester:** - IV Sem.	
Subject: - Object Oriented Programming through JAVA	Academic Year: - 2022-23	

PROBLEM STATEMENT AND SOLUTION

Problem Statement:

A lot of people go to restaurants every day, where tipping is quite a common practise. Keeping this in mind, we developed an application that makes the process easier by providing people with the a lot of tipping options.

Proposed Solution:

Our project aimed at solving simple day to day tip calculations of a private business or an individual. We created our app using Java Swing, AWT and JDBC libraries, that can help people calculate the tip with respect to the total bill. The application has the functionality to split the bill between multiple people. It is an advanced version that displays the total per person (which is bill + tip), bill per person and tip per person. Additionally, it also has the feature to save transaction details and display them in a list in the history page.

Hardware Requirements:

Intel i3 3rd gen or higher or AMD Ryzen 3 1600 or higher

Software Requirements:

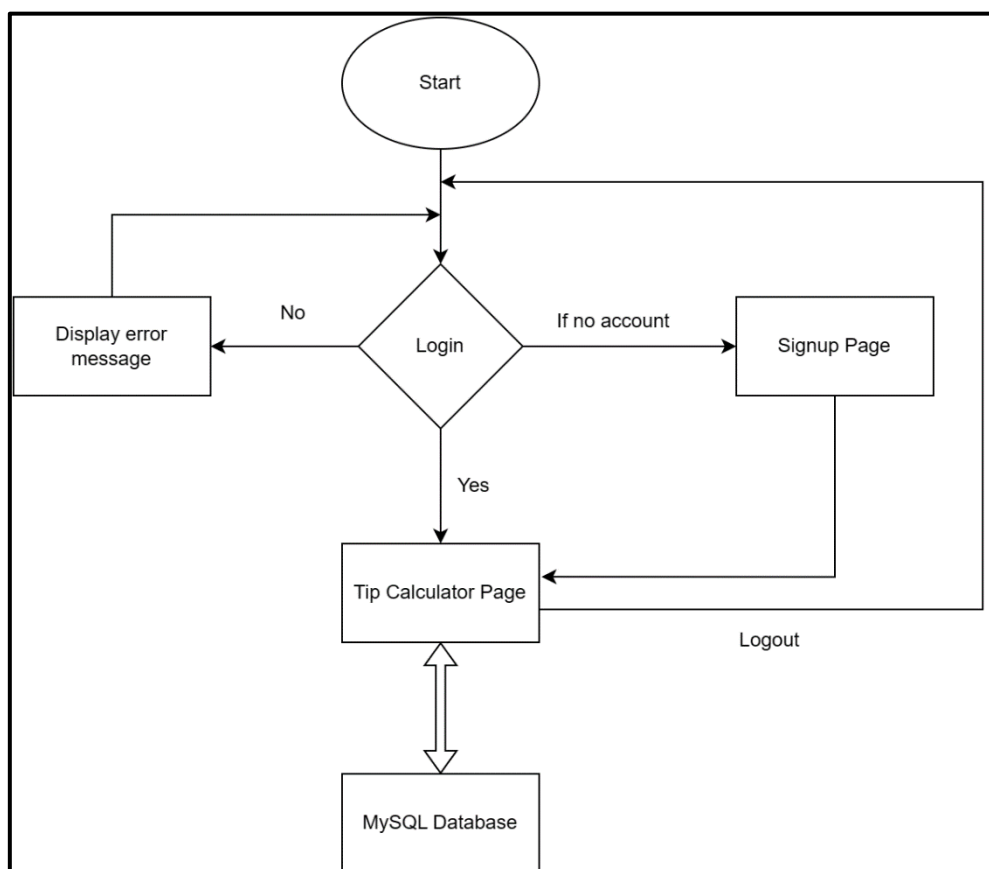Windows 7, 8, 10, 11 (32 or 64 bit), Eclipse IDE, MySQL database

MODULES USED

1. Swing: Swing is a Java graphical user interface (GUI) widget toolkit that allows developers to create sophisticated GUI applications for desktop and mobile platforms. It provides a set of components such as buttons, labels, text fields, and more, that can be used to create interactive user interfaces. In our project, we used Swing to create the GUI components for our tip calculator, login page, and signup page.

2. JDBC: JDBC stands for Java Database Connectivity, and it is a Java API that provides a standard way to connect and interact with databases. It allows Java programs to execute SQL statements, query and retrieve data from databases, and perform various database operations. In our project, we used JDBC to connect to a MySQL database and store user information.

3. AWT: AWT (Abstract Window Toolkit) is a Java library that provides a set of APIs for creating and managing graphical user interface (GUI) components such as windows, buttons, text fields, and other UI elements. It was introduced in the early versions of Java and has been available in all subsequent versions. AWT was one of the first GUI libraries introduced for Java and it is still in use today. It is often used for creating simple UIs, such as those found in utility applications or small desktop applications.

SOFTWARE ARCHITECTURE DIAGRAM

The software architecture of the project can be divided into three main components: the front-end user interface, the back-end database, and the application logic.

The front-end user interface is responsible for displaying the login, signup, and tip calculator pages to the user. This component is implemented using the AWT library and includes the graphical user interface (GUI) components, such as buttons and text fields, that the user interacts with.

The back-end database is responsible for storing and retrieving user information and tip calculation data. This component is implemented using the JDBC library and includes database management system (DBMS) components, such as the database itself, tables, and queries.

The application logic is responsible for managing the interaction between the front-end user interface and the back-end database. It defines how the login and signup pages work and how the calculations work in the tip calculator. The app logic is also responsible for creating and accessing data stored in the MySQL database.

SOURCE CODE

Login Page:

```java
import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

import java.sql.*;

public class LoginPage extends JFrame implements ActionListener {

/**

*

*/

private static final long serialVersionUID = 1L;

// Database credentials

static final String DB_URL = "jdbc:mysql://localhost:3306/nmims";

static final String USER = "root";

static final String PASS = "rick@astley";

// UI components

JLabel emailLabel, passwordLabel, titleLabel;

JTextField emailField;

JPasswordField passwordField;

JButton loginButton, signupButton;

public LoginPage() {

// Set up UI components

titleLabel = new JLabel("Login Page");
```

NMIMS, MPSTME, Shirpur Campus

| Computer Engineering | **Class: -** B Tech/ MBA Tech (CE) **Semester:** - IV Sem. |
| **Subject: -** Object Oriented Programming through JAVA | Academic Year: - 2022-23 |

```java
emailLabel = new JLabel("Email:");

emailField = new JTextField(20);

passwordLabel = new JLabel("Password:");

passwordField = new JPasswordField(20);

loginButton = new JButton("Login");

signupButton = new JButton("Sign Up");
// Set up layout manager

setLayout(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

gbc.gridwidth = GridBagConstraints.REMAINDER;

gbc.fill = GridBagConstraints.HORIZONTAL;

gbc.insets = new Insets(10, 10, 10, 10);
// Add components to layout

add(titleLabel, gbc);

titleLabel.setFont(new Font("Tahoma", Font.BOLD, 24));

titleLabel.setHorizontalAlignment(SwingConstants.CENTER);

add(emailLabel, gbc);

add(emailField, gbc);

add(passwordLabel, gbc);

add(passwordField, gbc);

add(loginButton, gbc);

add(signupButton, gbc);

emailField.setPreferredSize(new Dimension(emailField.getPreferredSize().width,
30));

passwordField.setPreferredSize(new
Dimension(passwordField.getPreferredSize().width, 30));
```

```java
// Add action listeners to buttons

loginButton.addActionListener(this);

signupButton.addActionListener(this);

// Set window properties

setTitle("Login Page");

setSize(500, 500);

setLocationRelativeTo(null);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

setVisible(true);

signupButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

setVisible(false);

SignupPage signupPage = new SignupPage();

}

});

}

public static void main(String[] args) {

new LoginPage();

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == loginButton) {

String email = emailField.getText();

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

if (email.equals("") || password.equals("")) {

JOptionPane.showMessageDialog(null, "Please fill in all fields.");
```

```java
}

else {

if (authenticateUser(email, password)) {

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

this.dispose();

TipCalculatorGUI tipCalculator = new TipCalculatorGUI((getUserName(email,
password)));

} else {

JOptionPane.showMessageDialog(null, "Incorrect email or password.");

}

}

} else if (e.getSource() == signupButton) {

emailField.setText("");

passwordField.setText("");

}

}

private boolean authenticateUser(String email, String password) {

try {
// Connect to database

Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);

// Prepare SQL statement

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

PreparedStatement stmt = conn.prepareStatement(sql);

stmt.setString(1, email);

stmt.setString(2, password);
```

```java
// Execute query

ResultSet rs = stmt.executeQuery();

// Check if user exists in database

if (rs.next()) {

return true;

} else {

return false;

}

} catch (SQLException ex) {

ex.printStackTrace();

return false;

}

}

private String getUserName(String email, String password) {

try {

// Connect to database

Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);

// Prepare SQL statement

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

PreparedStatement stmt = conn.prepareStatement(sql);

stmt.setString(1, email);

stmt.setString(2, password);

// Execute query

ResultSet rs = stmt.executeQuery();

rs.next();
```

```java
String userName = rs.getString("first_name");

return userName;

} catch (SQLException ex) {

ex.printStackTrace();

return "";

}

}

}
```

Sign Up Page:

```java
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

import java.sql.*;

public class SignupPage extends JFrame implements ActionListener {

/**
 *
 */

private static final long serialVersionUID = 1L;

// GUI components

JLabel titleLabel, nameLabel, lastNameLabel, phoneLabel, emailLabel,
passwordLabel;

JTextField nameField, lastNameField, phoneField, emailField;

JPasswordField passwordField;

JButton submitButton, resetButton;
```

```java
JPanel mainPanel;

// Database connection

Connection con;

public SignupPage() {

// Create database connection

try {

Class.forName("com.mysql.cj.jdbc.Driver");

String url = "jdbc:mysql://localhost:3306/nmims";

String user = "root";

String password = "rick@astley";

con = DriverManager.getConnection(url, user, password);

} catch (Exception e) {

System.out.println(e);

}

// Set the layout of the main panel

mainPanel = new JPanel();

mainPanel.setLayout(new GridBagLayout());

GridBagConstraints c = new GridBagConstraints();

c.fill = GridBagConstraints.HORIZONTAL;

c.insets = new Insets(10, 10, 10, 10);

// Add the title label to the main panel

titleLabel = new JLabel("Sign Up Form");

titleLabel.setFont(new Font("Tahoma", Font.BOLD, 30));

titleLabel.setHorizontalAlignment(SwingConstants.CENTER);

c.gridx = 0;
```

```java
c.gridy = 0;

c.gridwidth = 2;

mainPanel.add(titleLabel, c);

// Add the name label and text field to the main panel

nameLabel = new JLabel("First Name:");

c.gridx = 0;

c.gridy = 1;

c.gridwidth = 1;

mainPanel.add(nameLabel, c);

nameField = new JTextField();

c.gridx = 1;

c.gridy = 1;

nameField.setColumns(15);

nameField.setPreferredSize(new Dimension(nameField.getPreferredSize().width, 30));

mainPanel.add(nameField, c);

// Add the last name label and text field to the main panel

lastNameLabel = new JLabel("Last Name:");

c.gridx = 0;

c.gridy = 2;

mainPanel.add(lastNameLabel, c);

lastNameField = new JTextField();

c.gridx = 1;

c.gridy = 2;

lastNameField.setPreferredSize(new Dimension(lastNameField.getPreferredSize().width, 30));
```

NMIMS, MPSTME, Shirpur Campus

Computer Engineering	**Class: -** B Tech/ MBA Tech (CE) **Semester:** - IV Sem.
Subject: - Object Oriented Programming through JAVA	Academic Year: - 2022-23

```java
mainPanel.add(lastNameField, c);

// Add the phone label and text field to the main panel

phoneLabel = new JLabel("Phone Number:");

c.gridx = 0;

c.gridy = 3;

mainPanel.add(phoneLabel, c);

phoneField = new JTextField();

c.gridx = 1;

c.gridy = 3;

phoneField.setPreferredSize(new Dimension(phoneField.getPreferredSize().width,
30));

mainPanel.add(phoneField, c);

// Add the email label and text field to the main panel

emailLabel = new JLabel("Email:");

c.gridx = 0;

c.gridy = 4;

mainPanel.add(emailLabel, c);

emailField = new JTextField();

c.gridx = 1;

c.gridy = 4;

emailField.setPreferredSize(new Dimension(emailField.getPreferredSize().width,
30));

mainPanel.add(emailField, c);

// Add the password label and text field to the main panel

passwordLabel = new JLabel("Password:");

c.gridx = 0;
```

NMIMS, MPSTME, Shirpur Campus

Computer Engineering	**Class: -** B Tech/ MBA Tech (CE) **Semester:** - IV Sem.
Subject: - Object Oriented Programming through JAVA	Academic Year: - 2022-23

```java
c.gridy = 5;

mainPanel.add(passwordLabel, c);

passwordField = new JPasswordField();

c.gridx = 1;

c.gridy = 5;

passwordField.setPreferredSize(new
Dimension(passwordField.getPreferredSize().width, 30));

mainPanel.add(passwordField, c);

// Add the submit button and reset button to the main panel

submitButton = new JButton("Submit");

submitButton.addActionListener(this);

c.gridx = 0;

c.gridy = 6;

mainPanel.add(submitButton, c);

Dimension buttonSize = submitButton.getPreferredSize();

buttonSize.width = 150; // Set the desired width

submitButton.setPreferredSize(buttonSize);

resetButton = new JButton("Reset");

resetButton.addActionListener(this);

c.gridx = 1;

c.gridy = 6;

Dimension buttonSize1 = resetButton.getPreferredSize();

buttonSize1.width = 100; // Set the desired width

resetButton.setPreferredSize(buttonSize);

mainPanel.add(resetButton, c);
```

```java
resetButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Clear all text fields

nameField.setText("");

lastNameField.setText("");

phoneField.setText("");

emailField.setText("");

passwordField.setText("");

}

});

// Add main panel to content pane

setContentPane(mainPanel);

// Set applet size and make visible

setSize(500, 500);

setTitle("Sign Up Page");

setLocationRelativeTo(null);

setVisible(true);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == submitButton) {

String firstName = nameField.getText();

String lastName = lastNameField.getText();

String phoneNumber = phoneField.getText();

String email = emailField.getText();

char[] passwordChars = passwordField.getPassword();
```

```java
String password = new String(passwordChars);

if (firstName.equals("") || lastName.equals("") || phoneNumber.equals("") ||
email.equals("") || password.equals("")) {

JOptionPane.showMessageDialog(null, "Please fill in all fields.");

} else {

try {

// Establish database connection

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/nmims",
"root", "rick@astley");

Statement stmt = conn.createStatement();

// Execute SQL query to insert user data into the database

String insertQuery = "INSERT INTO users (first_name, last_name, phone_number,
email, password) VALUES ('" + firstName + "', '" + lastName + "', '" + phoneNumber
+ "', '" + email + "', '" + password + "')";

stmt.executeUpdate(insertQuery);

// Close database connection and show success message

conn.close();

JOptionPane.showMessageDialog(null, "Sign Up successful!");

this.dispose();

TipCalculatorGUI tipCalculator = new TipCalculatorGUI(firstName);

} catch (SQLException ex) {

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

}

}

}

}

//public static void main(String[] args) {
```

```
// SignupPage signupPage = new SignupPage();

//}

}
```

Tip Calculator Page:

```java
import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class TipCalculatorGUI extends JFrame {

private static final long serialVersionUID = 1L;

private JLabel billLabel = new JLabel("Bill:");

private JLabel tipPercentLabel = new JLabel("Tip:");

private JTextField billField = new JTextField(10);

private JRadioButton tip10Button = new JRadioButton("10%");

private JRadioButton tip15Button = new JRadioButton("15%");

private JRadioButton tip20Button = new JRadioButton("20%");

private ButtonGroup tipButtonGroup = new ButtonGroup();

private JLabel splitLabel = new JLabel("Split:");

private JButton minusButton = new JButton("-");
```

```java
private JLabel splitValueLabel = new JLabel("1");

private JButton addButton = new JButton("+");

private JButton showResultsButton = new JButton("Retrieve Records");

private JButton calculateButton = new JButton("Calculate");

private JButton logoutButton = new JButton("Logout");

private TextArea resultText = new TextArea(5, 20);

public TipCalculatorGUI(String name) {

JLabel userName = new JLabel("Hey, " + name + "!");

// Set component fonts

Font userNameFont = new Font("Tahoma", Font.BOLD, 18);

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

Font fieldFont = new Font("Arial", Font.PLAIN, 18);

Font buttonFont = new Font("Arial", Font.PLAIN, 16);

Font resultFont = new Font("Arial", Font.BOLD, 24);

userName.setFont(userNameFont);

billLabel.setFont(labelFont);

tipPercentLabel.setFont(labelFont);

billField.setFont(fieldFont);

logoutButton.setFont(buttonFont);

tip10Button.setFont(buttonFont);

tip15Button.setFont(buttonFont);

tip20Button.setFont(buttonFont);

splitLabel.setFont(labelFont);

minusButton.setFont(buttonFont);

splitValueLabel.setFont(labelFont);
```

```java
addButton.setFont(buttonFont);

calculateButton.setFont(buttonFont);

showResultsButton.setFont(buttonFont);

resultText.setFont(resultFont);

// Set component colors

Color backgroundColor = new Color(240, 240, 240);

Color foregroundColor = new Color(20, 20, 20);

Color highlightColor = new Color(200, 200, 200);

Color buttonColor = new Color(230, 230, 230);

userName.setForeground(foregroundColor);

billLabel.setForeground(foregroundColor);

tipPercentLabel.setForeground(foregroundColor);

billField.setBackground(backgroundColor);

billField.setForeground(foregroundColor);

logoutButton.setBackground(buttonColor);

tip10Button.setBackground(buttonColor);

tip15Button.setBackground(buttonColor);

tip20Button.setBackground(buttonColor);

logoutButton.setForeground(foregroundColor);

tip10Button.setForeground(foregroundColor);

tip15Button.setForeground(foregroundColor);

tip20Button.setForeground(foregroundColor);

tip10Button.setFocusPainted(false);

tip15Button.setFocusPainted(false);

tip20Button.setFocusPainted(false);
```

```java
tip10Button.setBorderPainted(false);

tip15Button.setBorderPainted(false);

tip20Button.setBorderPainted(false);

splitLabel.setForeground(foregroundColor);

minusButton.setBackground(buttonColor);

splitValueLabel.setBackground(backgroundColor);

splitValueLabel.setForeground(foregroundColor);

addButton.setBackground(buttonColor);

calculateButton.setBackground(highlightColor);

showResultsButton.setBackground(highlightColor);

resultText.setForeground(foregroundColor);
// Add action listeners

calculateButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

calculateTip();

}

});

minusButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int splitValue = Integer.parseInt(splitValueLabel.getText());

if (splitValue > 1) {

splitValue--;

splitValueLabel.setText(Integer.toString(splitValue));

}

}
```

NMIMS, MPSTME, Shirpur Campus

| Computer Engineering | **Class: -** B Tech/ MBA Tech (CE) **Semester:** - IV Sem. |
| Subject: - Object Oriented Programming through JAVA | Academic Year: - 2022-23 |

```java
});

addButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int splitValue = Integer.parseInt(splitValueLabel.getText());

splitValue++;

splitValueLabel.setText(Integer.toString(splitValue));

}

});

showResultsButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent evt) {

viewResultsActionPerformed(evt);

}

});
// Set layout manager and add components

JPanel mainPanel = new JPanel(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

gbc.gridx = 0;

gbc.gridy = 0;

gbc.anchor = GridBagConstraints.LINE_END;

gbc.insets = new Insets(10, 10, 10, 10);

mainPanel.add(userName, gbc);

gbc.gridx = 0;

gbc.gridy = 1;

gbc.anchor = GridBagConstraints.LINE_START;

gbc.insets = new Insets(10, 10, 10, 10);
```

NMIMS, MPSTME, Shirpur Campus

| Computer Engineering | **Class: -** B Tech/ MBA Tech (CE) **Semester:** - IV Sem. |
| **Subject: -** Object Oriented Programming through JAVA | Academic Year: - 2022-23 |

```java
mainPanel.add(billLabel, gbc);

gbc.gridx = 1;

gbc.anchor = GridBagConstraints.LINE_START;

mainPanel.add(billField, gbc);

gbc.gridx = 0;

gbc.gridy = 2;

gbc.gridwidth = 2;

gbc.insets = new Insets(0, 10, 10, 10);

mainPanel.add(new JSeparator(), gbc);

gbc.gridx = 0;

gbc.gridy = 3;

gbc.anchor = GridBagConstraints.LINE_START;

gbc.insets = new Insets(10, 10, 10, 10);

mainPanel.add(tipPercentLabel, gbc);

gbc.gridx = 0;

gbc.gridy = 3;

gbc.gridwidth = 1;

gbc.insets = new Insets(10, 10, 10, 10);

gbc.gridx = 1;

gbc.anchor = GridBagConstraints.LINE_START;

mainPanel.add(tip10Button, gbc);

gbc.gridx = 1;

gbc.anchor = GridBagConstraints.LINE_END;

mainPanel.add(tip15Button, gbc);

gbc.gridx = 2;
```

```java
gbc.anchor = GridBagConstraints.LINE_START;

mainPanel.add(tip20Button, gbc);

gbc.gridx = 0;

gbc.gridy = 4;

gbc.gridwidth = 2;

gbc.insets = new Insets(0, 10, 10, 10);

mainPanel.add(new JSeparator(), gbc);

gbc.gridx = 0;

gbc.gridy = 5;

gbc.insets = new Insets(0, 10, 10, 10);

mainPanel.add(splitLabel, gbc);

gbc.gridx = 1;

gbc.anchor = GridBagConstraints.LINE_START;

mainPanel.add(minusButton, gbc);

gbc.gridx = 1;

gbc.anchor = GridBagConstraints.CENTER;

mainPanel.add(splitValueLabel, gbc);

gbc.gridx = 1;

gbc.anchor = GridBagConstraints.LINE_END;

mainPanel.add(addButton, gbc);

gbc.gridx = 0;

gbc.gridy = 6;

gbc.gridwidth = 4;

gbc.insets = new Insets(10, 10, 10, 10);

mainPanel.add(new JSeparator(), gbc);
```

```java
gbc.gridx = 0;

gbc.gridy = 7;

gbc.gridwidth = 4;

gbc.anchor = GridBagConstraints.LINE_START;

gbc.insets = new Insets(10, 10, 10, 10);

mainPanel.add(calculateButton, gbc);

gbc.gridx = 1;

gbc.gridy = 7;

gbc.anchor = GridBagConstraints.LINE_END;

mainPanel.add(showResultsButton, gbc);

gbc.gridx = 0;

gbc.gridy = 8;

gbc.gridwidth = 4;

gbc.insets = new Insets(10, 10, 10, 10);

mainPanel.add(resultText, gbc);

gbc.gridx = 1;

gbc.gridy = 9;

gbc.anchor = GridBagConstraints.LINE_END;

mainPanel.add(logoutButton, gbc);

// Set radio buttons in a group

tipButtonGroup.add(tip10Button);

tipButtonGroup.add(tip15Button);

tipButtonGroup.add(tip20Button);

// Add main panel to content pane

setContentPane(mainPanel);
```

```java
// Set applet size and make visible

setSize(600, 600);

setTitle("Tip Calculator");

setLocationRelativeTo(null);

setVisible(true);

logoutButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

setVisible(false);

new LoginPage();

}

});

}

private void calculateTip() {

try {

double bill = Double.parseDouble(billField.getText());

double tipPercentage = 0.0;

if (tip10Button.isSelected()) {

tipPercentage = 0.1;

} else if (tip15Button.isSelected()) {

tipPercentage = 0.15;

} else if (tip20Button.isSelected()) {

tipPercentage = 0.2;

}

int splitValue = Integer.parseInt(splitValueLabel.getText());

double tipAmount = bill * tipPercentage;
```

```java
double totalAmount = bill + tipAmount;

double perPersonAmount = totalAmount / splitValue;

resultText.setText(

String.format("Total Bill: ₹%.2f\nTip Amount: ₹%.2f\nTotal Amount: ₹%.2f\nPer
Person Amount: ₹%.2f",

bill, tipAmount, totalAmount, perPersonAmount));

saveData(bill, tipPercentage, splitValue, tipAmount, totalAmount,
perPersonAmount);

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(this,

"Invalid input. Please enter a valid number for Bill amount and Split number.");

}

}

private void saveData(

double f_bill,

double f_tipPercentage,

int f_splitNumber,

double f_tipAmount,

double f_totalAmount,

double f_perPersonAmount) {

// Save results to database

try {

f_tipPercentage = 100 * f_tipPercentage;

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/nmims",
"root", "rick@astley");

PreparedStatement statement = conn.prepareStatement(
```

```java
"INSERT INTO tip_calculator_results (bill, tip_percentage, split_count,
tip_amount, total_amount, per_person_amount) VALUES (?, ?, ?, ?, ?, ?)");

statement.setDouble(1, f_bill);

statement.setDouble(2, f_tipPercentage);

statement.setInt(3, f_splitNumber);

statement.setDouble(4, f_tipAmount);

statement.setDouble(5, f_totalAmount);

statement.setDouble(6, f_perPersonAmount);

statement.executeUpdate();

statement.close();

conn.close();

System.out.println("Result saved to database.");

} catch (SQLException ex) {

System.out.println("Failed to save result to database.");

ex.printStackTrace();

}

}

private void viewResultsActionPerformed(ActionEvent evt) {

try {

// create a connection to the database

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/nmims",
"root", "rick@astley");

// create a prepared statement to select all records from the

// tip_calculator_results table

// order by id desc because we need to reverse the result set to get the 5 latest
rows.
```

```java
// ...as new data is appended at the end

String query = "SELECT * FROM tip_calculator_results ORDER BY id DESC";

PreparedStatement ps = con.prepareStatement(query);

// execute the query and get the result set

ResultSet rs = ps.executeQuery();

// create a string to hold the results

String results = "";

// loop through the result set and append latest 5 rows to the results string

int num_of_records = 5;

int count = 1;

while (rs.next() && num_of_records > 0) {

results += count + ") Bill amount: ₹" + rs.getDouble("bill") + ", Tip percentage:
"

+ rs.getDouble("tip_percentage") + "%, Number of people: " +
rs.getInt("split_count")

+ ", Total per person: ₹" + rs.getDouble("per_person_amount") + "\n";

num_of_records--;

count++;

}

// display the results in a message dialog

if (results.equals("")) {

JOptionPane.showMessageDialog(null, "No results found");

} else {

JOptionPane.showMessageDialog(null, results);

}

// close the database connection and prepared statement
```

```java
ps.close();

con.close();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "Error connecting to database");

System.out.print(ex);

}

}

}
```
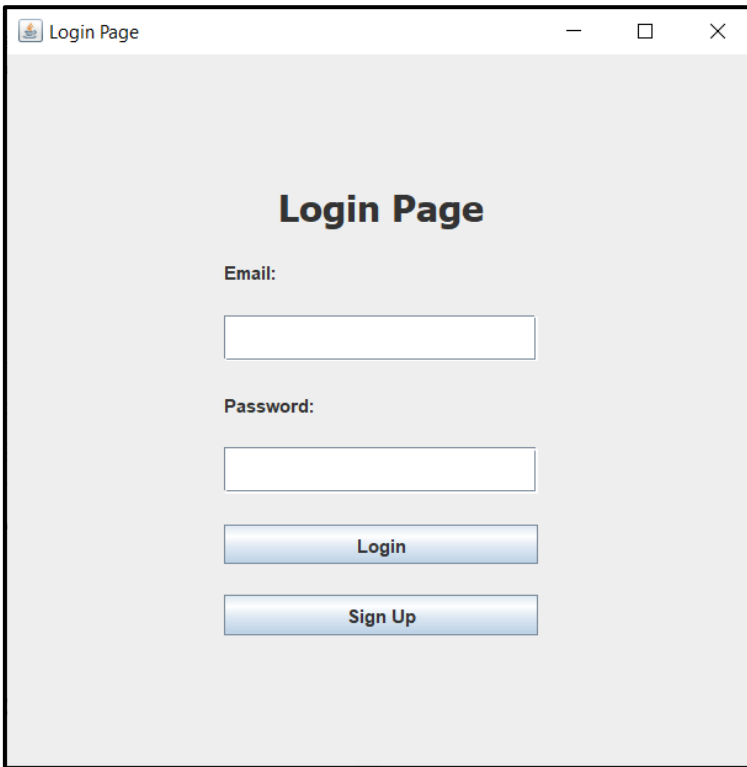
USER INTERFACE

Login Page:
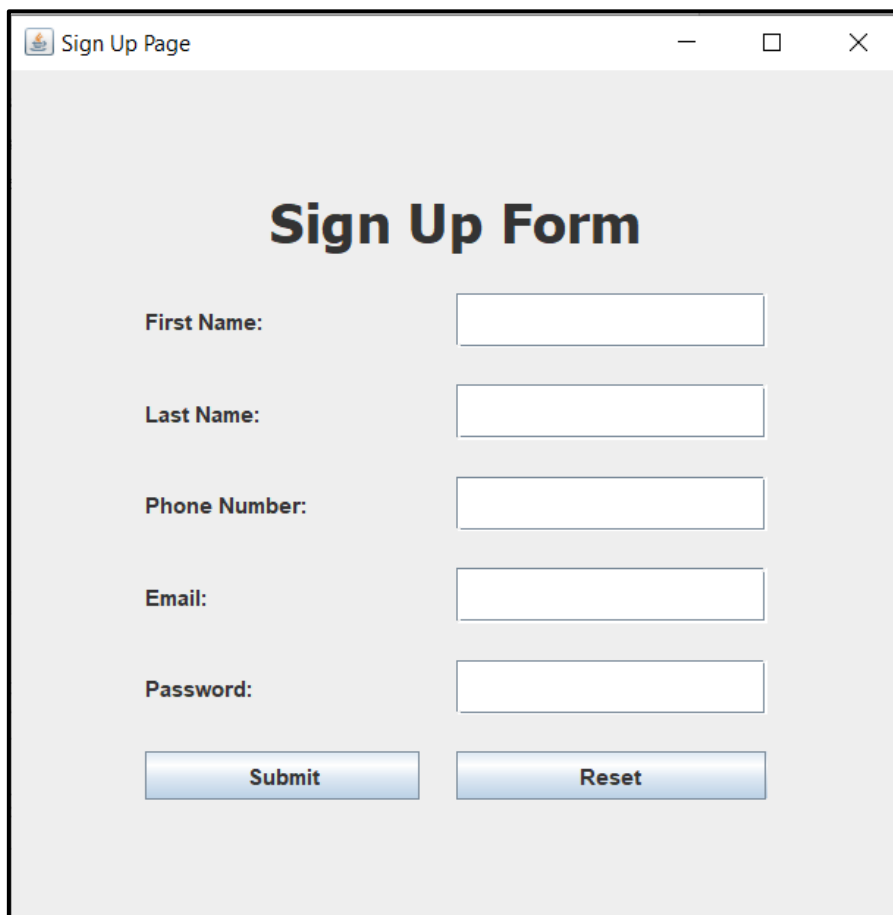
The login page is the first page that the user would see on launching the application. They can enter their email and password to use the calculator. If the entered credentials are present in the database, then the login would be successful. Otherwise, an error message will be shown for incorrect email and password. A new account can be created by going to the signup page by clicking the respective button.

Sign Up Page:
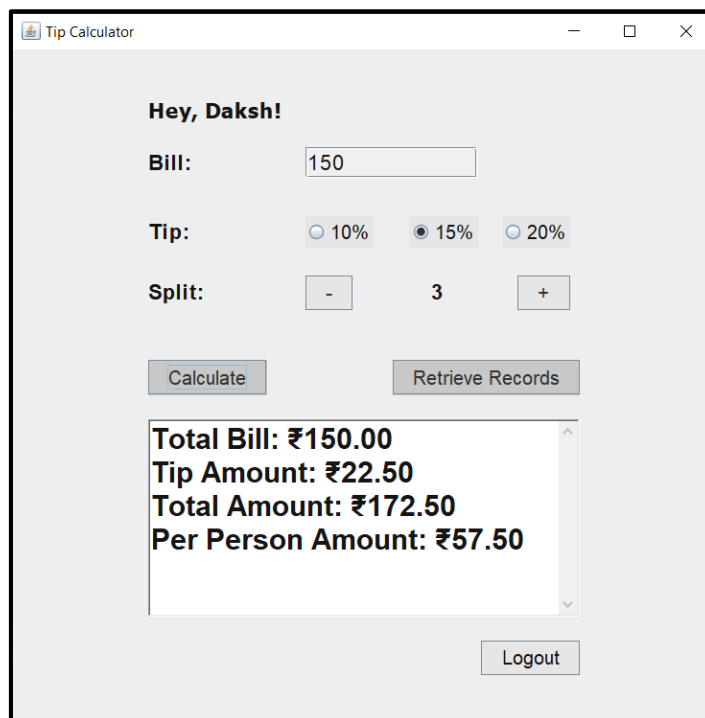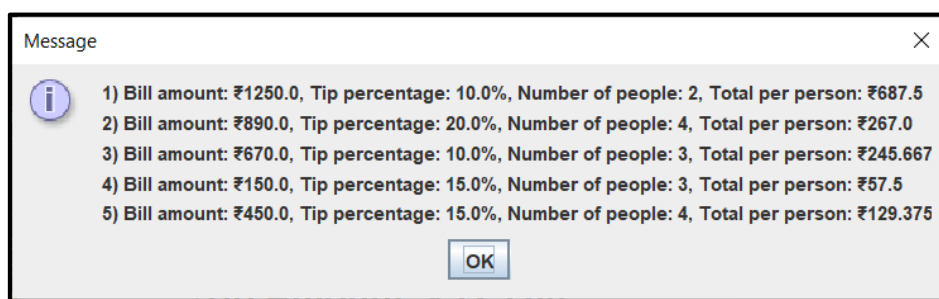
At this page, the user can input their credentials to create a new account. The form asks for basic details such as first name, last name, phone number, email and password. On clicking 'Submit', the details will be saved to a MySQL database, which can be later accessed for login authentication. If any of the fields are empty, then a message is displayed requesting the user to fill in all the details. There is also a 'reset' button that clears and resets all the fields.

Tip Calculator Page:

This is the main page of the application. At the top is a greeting message that takes the name of the account user according to the email and password entered at the login screen. After it is the 'Bill' field, where users can input the total bill amount. Then, there is an option to select the tip percentage from three available choices. This defines the percentage of the bill that the user wants to giveaway as the tip. After that is the split option that provides the feature to split the bill between multiple people.

On clicking 'calculate', respective calculations are done, and the values are displayed. These values are also stored in a MySQL database. Tapping the "Retrieve Records" button accesses and displays the five latest records. At the end of the window is the logout button which logs the user out.

MySQL DATABASE

Below is the description of the MySQL table for storing tip calculator results:

```
mysql> desc tip_calculator_results;
+-------------------+-----------+------+-----+-------------------+-------------------+
| Field             | Type      | Null | Key | Default           | Extra             |
+-------------------+-----------+------+-----+-------------------+-------------------+
| id                | int       | NO   | PRI | NULL              | auto_increment    |
| bill              | float     | NO   |     | NULL              |                   |
| tip_percentage    | float     | NO   |     | NULL              |                   |
| split_count       | int       | NO   |     | NULL              |                   |
| tip_amount        | float     | NO   |     | NULL              |                   |
| total_amount      | float     | NO   |     | NULL              |                   |
| per_person_amount | float     | NO   |     | NULL              |                   |
| created_at        | timestamp | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-------------------+-----------+------+-----+-------------------+-------------------+
8 rows in set (0.02 sec)
```

The tip calculator data stored in the database:

```
mysql> select * from tip_calculator_results;
+----+------+----------------+-------------+------------+--------------+-------------------+---------------------+
| id | bill | tip_percentage | split_count | tip_amount | total_amount | per_person_amount | created_at          |
+----+------+----------------+-------------+------------+--------------+-------------------+---------------------+
|  1 |  150 |           0.15 |           3 |       22.5 |        172.5 |              57.5 | 2023-03-01 01:15:00 |
|  2 |  150 |            0.2 |           4 |         30 |          180 |                45 | 2023-03-01 01:17:17 |
|  3 |  250 |             15 |           3 |       37.5 |        287.5 |           95.8333 | 2023-03-01 01:40:24 |
|  4 |  350 |             15 |           2 |       52.5 |        402.5 |            201.25 | 2023-03-01 01:42:45 |
|  5 |  890 |             20 |           6 |        178 |         1068 |               178 | 2023-03-01 01:42:50 |
|  6 |  600 |             15 |           3 |         90 |          690 |               230 | 2023-03-01 01:42:56 |
|  7 |  150 |             20 |           2 |         30 |          180 |                90 | 2023-03-01 01:53:59 |
|  8 | 1575 |              0 |           4 |          0 |         1575 |            393.75 | 2023-03-01 01:56:55 |
|  9 | 1575 |             15 |           4 |     236.25 |      1811.25 |           452.812 | 2023-03-01 01:56:59 |
| 10 |  780 |             15 |           4 |        117 |          897 |            224.25 | 2023-03-01 02:05:11 |
| 11 |  340 |             20 |           3 |         68 |          408 |               136 | 2023-03-01 02:06:36 |
| 12 |  150 |             20 |           4 |         30 |          180 |                45 | 2023-03-01 12:47:25 |
| 13 |  180 |             15 |           4 |         27 |          207 |             51.75 | 2023-03-01 12:50:20 |
| 14 |  159 |             10 |           5 |       15.9 |        174.9 |             34.98 | 2023-03-04 15:26:22 |
| 15 |  150 |             15 |           7 |       22.5 |        172.5 |           24.6429 | 2023-03-14 16:40:32 |
| 16 |  150 |             15 |           4 |       22.5 |        172.5 |            43.125 | 2023-03-16 13:07:07 |
| 17 |  450 |             15 |           4 |       67.5 |        517.5 |           129.375 | 2023-03-20 02:09:20 |
| 18 |  150 |             15 |           3 |       22.5 |        172.5 |              57.5 | 2023-03-20 21:28:38 |
| 19 |  670 |             10 |           3 |         67 |          737 |           245.667 | 2023-03-20 23:41:00 |
| 20 |  890 |             20 |           4 |        178 |         1068 |               267 | 2023-03-20 23:41:06 |
| 21 | 1250 |             10 |           2 |        125 |         1375 |             687.5 | 2023-03-20 23:41:18 |
+----+------+----------------+-------------+------------+--------------+-------------------+---------------------+
21 rows in set (0.00 sec)
```

Below is the description of the MySQL table for storing user authentication data:

```
mysql> desc users;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| id           | int          | NO   | PRI | NULL    | auto_increment |
| first_name   | varchar(255) | NO   |     | NULL    |                |
| last_name    | varchar(255) | NO   |     | NULL    |                |
| phone_number | varchar(20)  | NO   |     | NULL    |                |
| email        | varchar(255) | NO   | UNI | NULL    |                |
| password     | varchar(255) | NO   |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

The user authentication data stored in the database:

```
mysql> select * from users;
+----+------------+-----------+--------------+----------------+----------+
| id | first_name | last_name | phone_number | email          | password |
+----+------------+-----------+--------------+----------------+----------+
| 35 | Daksh      | Gehlot    | 12345        | daksh@com      | daksh    |
| 36 | Jack       | Ma        | 1234567      | jack@gmail.com | 0987654  |
| 37 | Tom        | Cruise    | 3456789      | tom@gmail.com  | lkjhgfdsa|
+----+------------+-----------+--------------+----------------+----------+
3 rows in set (0.00 sec)
```

PROJECT METHODOLOGY

The methodology of the "Tip Calculator" project involves a systematic approach to designing and implementing a system that takes in the bill input and gives you appropriate results. The project is divided into several stages, including planning and setup, software programming, and testing.

1. **Project Initiation**

- In this phase, we identified that the problem is to develop a tip calculator application using Java. The scope of the project is limited to a tip calculator that can take bill input from users, calculate the tip percentage based on user selection, split the bill between users, and display the final calculated data on the screen.

- Define project objectives, requirements, and success criteria: The objective of the project is to develop a tip calculator application that can perform the desired calculations accurately and quickly. The requirements include the use of Java programming language, Swing, AWT, and JDBC for user interface, database connectivity, and data manipulation. The success criteria are the ability of the application to meet the user's requirements, accuracy in calculating the tip, and easy-to-use interface.

- Obtain necessary resources, including software and hardware tools: The necessary resources for this project included Java development kit, an Integrated Development Environment (IDE) such as Eclipse or NetBeans, a database management system, and any other necessary libraries or tools.

2. **Analysis**

- Identify and analyze the users and stakeholders: In this phase, you identified the users of the application as people who want to calculate tips and split bills easily. The stakeholders include the application developer, potential investors, and users who may provide feedback.

- Create use cases and scenarios: Use cases and scenarios were created to understand how the users would interact with the application, how they would input bill details, select the tip percentage, split the bill, and view the final calculated data.

- Gather functional and non-functional requirements: Functional requirements included the ability to input bill details, select a tip percentage, split the bill, and view the final calculated data. Non-functional requirements included the need for accuracy, speed, user-friendliness, and reliability.

- Analyze the existing system, if any, and its limitations: In this case, since there is no existing system, this step was skipped.

3. **Design**

- Create a high-level design of the system: A high-level design of the system included identifying the key components of the system, such as the user interface, the database, and the algorithms used for calculations.

- Develop detailed design of the user interface and functionality: Detailed designs were created for the user interface, including the layout of the input fields, buttons, and other controls. Functionality was designed to ensure that the application could perform the necessary calculations accurately and efficiently.

- Choose appropriate technologies and tools: The appropriate technologies and tools for this project included Java, Swing, AWT, and JDBC.

- Identify and mitigate risks: Risks associated with the project, such as technical challenges, budget constraints, and time limitations, were identified and mitigated by incorporating contingency plans and risk management strategies.

4. **Development**

- Develop and test the system components, including the user interface, database, and algorithms: The user interface was developed using Java Swing and AWT, and the database connectivity was established using JDBC. The algorithms used for calculations were developed and tested for accuracy and efficiency.

- Integrate the components into a working system: The user interface, database, and calculations were integrated into a working system that could take input from the user, perform calculations, and display the results on the screen.

- Conduct functional and non-functional testing, including unit testing, integration testing, and system testing: Testing was conducted to ensure that the application was functioning correctly and meeting the requirements. Unit testing was used to test individual components, integration testing was used to test the integration between the different components, and system testing was used to test the application as a whole.

5. **Deployment**

- Deploy the application to a production environment: The application was deployed to a production environment, which in this case could be a local machine or a server.

- Conduct user acceptance testing: User acceptance testing was conducted to ensure that the application was meeting the needs of the users and functioning as expected.

- Provide user training and documentation: User training and documentation were provided to ensure that the users could use the application effectively and efficiently.

6. **Maintenance**

- Monitor and maintain the system to ensure optimal performance: The system was monitored and maintained to ensure that it was performing optimally and that any issues were addressed in a timely manner.

- Conduct periodic updates and maintenance: Periodic updates and maintenance were conducted to ensure that the application remained up-to-date with the latest technologies and software updates.

- Address any user feedback and incorporate necessary changes: User feedback was solicited and addressed to ensure that the application was meeting the needs of the users and any necessary changes were incorporated.

RESULT

The project was successfully implemented and achieved all the goals set out in the project plan. The login and signup pages allow users to securely access the tip calculator page. The tip calculator page works as intended and provides an accurate calculation of the tip amount based on user inputs.

The database connectivity feature was implemented using the MySQL database management system. The feature allows users to store and retrieve tip calculation history, making it easy to keep track of expenses. The application was thoroughly tested for errors and bugs, and all issues were resolved before submission.

Overall, the project was completed within the expected timeframe and met all requirements.

CONCLUSION

In conclusion, the tip calculator application project was a success. The project implemented all desired features, such as the login and signup pages, the tip calculator, and the database connectivity feature. The application is user-friendly, easy to use, and provides an accurate calculation of the tip amount.

The project provided valuable experience in software development, including requirements analysis, design, implementation, and testing. This project also gave us exposure to database management systems, which are critical components of most software applications.

In future iterations of this project, additional features such as the ability to split the tip among multiple individuals or the option to add tax to the total bill could be added to make the application even more useful.

Overall, this project was a great learning experience and provided valuable skills for future software development projects.

REFERENCES

1. javax.swing (Java Platform SE 7) (oracle.com)

2. Java Swing Tutorial - javatpoint

3. Introduction to Java Swing - GeeksforGeeks

4. java.awt (Java Platform SE 7) (oracle.com)

5. Java AWT Tutorial - javatpoint

6. Java JDBC API (oracle.com)

7. Spring Data JDBC - Reference Documentation

8. Stack Overflow - Where Developers Learn, Share, & Build Careers

9. GitHub