PROJECT BASED LEARNING (PBL)

Report on

RESTAURANT BOOKING SYSTEM

Submitted for partial fulfillment of the requirements for the subject

22ITC06 & JAVA PROGRAMMING

3rd SEMESTER / 2nd YEAR

BACHELOR OF TECHNOLOGY
PBL REPORT
Submitted by

Register No.	Name of the student
23IT040	S.MONISH
23IT046	N.SANDHIYA
23IT059	S.TANYA
23ITL09	K.SHANJAI

Project coordinator

Mrs.D.KIRUTHIKA AP/IT

DEPARTMENT OF INFORMATION TECHNOLOGY
NANDHA ENGINEERING COLLEGE (Autonomous)
ERODE – 638052

DECEMBER 2024

NANDHA ENGINEERING COLLEGE(Autonomous)
ERODE-638052
(An Autonomous Institution, Affiliated to Anna University, Chennai)

CERTIFICATE

This is to certify that the project work entitled "RESTAURANT BOOKING SYSTEM" is a bonafide work carried out by

Register No.	Name of the student
23IT040	S.MONISH
23IT046	N.SANDHIYA
23IT059	S.TANYA
23ITL09	K.SHANJAI

in partial fulfillment of the requirements for the course 22ITC06-JAVA PROGRAMMING, 3rd SEMESTER & 2nd YEAR Of BACHELOR OF

TECHNOLOGY IN INFORMATION TECHNOLOGY

during the academic year 2024-25

PROJECT GUIDE

Mrs.D.Kiruthika,

Assistant professor

Department of IT

Nandha Engineering College

HEAD OF THE DEPARTMENT

Dr.G.Sreekanth

Professor

Nandha Engineering college

Erode - 638052

ACKNOWLEDGEMENT

The development of the project as part of Project based Learning in the course **22ITC06 JAVA PROGRAMMING**, **3rd SEMESTER &2nd YEAR** though it was an arduous task, it has been made by the help of many people. We are pleased to express our thanks to the people whose suggestions, comments, criticisms greatly encouraged us in betterment of the project. We would like to express our sincere gratitude and indebtedness to project Guide **D.KIRUTHIKA**, for his/her valuable suggestions and interest throughout the course of this project.

We wish to convey our gratefulness to our **Principal Dr.N.Rengarajan, B.Sc., B.Tech., M.E., Ph.D.,** for his strong support and motivation in completing this project.

We take this opportunity to express our thanks to **Dr.G.SREEKANTH** Head of the Department & Project Co-Ordinator for their continuous monitoring, motivation and priceless assistance during the course of our project work.

TABLE OF CONTENTS

ABSTRACT

The Restaurant Booking System is a user-friendly web-based application designed to streamline the process of reserving tables at restaurants. This system addresses the inefficiencies and limitations of traditional reservation methods by offering a modern, automated solution for both customers and restaurant administrators.

The project incorporates a dual-interface model: one for customers and one for restaurant staff. Customers can browse available restaurants, view menus, check table availability, and make reservations in real time. The system allows users to select specific time slots, party sizes, and seating preferences. Additionally, it provides notifications and reminders for confirmed bookings.

For restaurant administrators, the system includes tools for managing reservations, monitoring occupancy, and updating menu or seating availability. Advanced analytics and reporting features enable better resource management and customer service improvements.

The project is built using a combination of front-end and back-end technologies, ensuring a seamless user experience and robust functionality. It employs secure data handling to protect user information and integrates payment gateways for advanced features like deposit-based reservations. By enhancing convenience for customers and operational efficiency for restaurants, this project aims to modernize the dining reservation experience, reducing the need for manual processes and improving overall customer satisfaction.

CHAPTER 1

INTRODUCTION

Overview

Introduce the concept of a restaurant booking system, explaining its importance in modern hospitality. Discuss how it improves reservation efficiency, minimizes errors, and enhances customer satisfaction.

Example: "Traditional methods of recording reservations via phone calls or handwritten logs are outdated. An automated system can streamline the process, providing customers with real-time availability and confirmation."

Problem Statement

Challenges of manual booking systems: time consumption, human errors, and limited accessibility.

"With increasing demand for reservations, restaurants face difficulties in managing bookings effectively without a centralized system."

Objective

To develop a system that allows customers to book tables online, enables staff to manage reservations, and integrates features such as automated confirmations and customer data management

CHAPTER 2

SOFTWARE REQUIREMENT

2.1. HARDWARE CONFIGURATION

SYSTEM: ASUS

PROCESSOR

RAM : 8GB RAM

SSD Capacity :516 GB

2.2. SOFTWARE REQUIREMENTS:

Operating System :Windows 11

Software required : JDBC, MY SQL,VS CODE

2.3. SOFTWARE

DESCRIPTIONMY SQL

SQL is Structured Query Language, which is a computer language for storing,manipulating and retrieving data stored in a relational database.SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase,Informix, Postgres and SQL Server use SQL as their standard database language.SQL is a language to operate databases; it includes database creation, deletion,fetching rows, modifying rows, etc. SQL is an ANSI(American

NationalStandards Institute) standard language, but there are many different versions of the SQL language.Allows users to access data in the relational database management systems.Allows users to describe the data.Allows users to define the data in a database and manipulate that data.Allows to embed within other languages using SQL modules, libraries & pre-compilers.Allows users to create and drop databases and tables.Allows users to create view, stored procedure, functions in a database.Allows users to set permissions on table procedures and views.

2.4.INTRODUCTION TO JDBC

Java Database Connectivity (JDBC) is an application programming interface(API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented toward relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the Java virtual machine (JVM) host environment. Query statements return a JDBC row result set. The row result set is used to walk over the result set. Individual columns in a row are retrieved either by name or by column number. There may be any number of rows in the result set. The row result set has metadata that describes the names of the columns and their types.

JDBC connections support creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT.

CHAPTER 3

PROJECT DESCRIPTION

DESCRIPTION OF MODULES

- Homepage
- Booking page
- Admin Dashboard
- Customer Management
- Payment module

This restaurant reservation system is designed to enhance the dining experience for customers and streamline operations for restaurant staff. The **Home Page** displays restaurant details, menu highlights, and an easy option to book a table. The **Booking Page** allows customers to select their preferred date, time, table size, and seating area, along with the option to add special requests. The **Admin Dashboard** enables staff to manage reservations in real-time, while the **Customer Management** module stores essential customer data for personalized service. The **Payment Module** integrates secure payment gateways for advance booking charges, ensuring smooth transactions and reducing no-shows. The system focuses on providing a seamless, efficient, and personalized experience for both customers and staff.

CHAPTER 4

CONCLUSION

The restaurant booking system represents a transformative solution for modernizing reservation management in the hospitality industry. By replacing traditional manual methods with a digital platform, this system enhances operational efficiency, reduces staff workload, and elevates customer satisfaction. Its core features, such as real-time availability tracking, automated confirmations, and customer data management, streamline the reservation process and minimize human errors. This not only ensures a seamless booking experience for customers but also helps restaurants optimize table utilization and increase revenue. Moreover, the system's scalability allows for future enhancements, such as loyalty programs, predictive analytics, and AI-driven features, making it adaptable to evolving industry needs. Overall, the restaurant booking system provides a robust, efficient, and customer-centric approach to reservation management, positioning restaurants to thrive in a competitive and dynamic market.

CHAPTER 5

FUTURE ENHANCEMENT

Future enhancements for the restaurant booking system could include integrating advanced features such as AI-driven table allocation to optimize seating arrangements, predictive analytics to forecast busy periods and manage resources efficiently, and loyalty programs to reward frequent customers. Additionally, voice-enabled booking through virtual assistants, real-time notifications via SMS or email, and multi-language support can enhance user experience. Expanding payment options to include digital wallets and integrating customer feedback mechanisms would further modernize the system, ensuring it meets the evolving needs of both customers and restaurant management.

REFERENCE

1. **RESEARCH PAPER AND ARTICLES**

   Rouse, M. (2019). *What is a Reservation System?* TechTarget.

   - **Discusses the fundamental concepts and benefits of automated reservation systems.**

2. **Kimes, S. E. (2011). *The Current State of Online Restaurant Reservations.***

   - **Explores trends in online reservation systems and their impact on the restaurant industry.**

3. **Kasavana, M. L. (2018). *Managing Technology in the Hospitality Industry.* Educational Institute of the American Hotel & Lodging Association.**

   - **Covers technological advancements in restaurant management.**

2. ONLINE PLATFORM AND SOFTWARE

1. **OpenTable**
   - **A popular online restaurant booking system.**
2. **Resy**
   - **Another reservation platform that focuses on providing modern solutions for diners and restaurants.**

3.TECHNOLOGIES AND FRAMEWORK

1. **Frontend Development: HTML5, CSS3, JavaScript (with libraries like React or Vue.js).**
2. **Backend Development: Python (Django/Flask), Node.js, or PHP (Laravel).**
3. **Database Management: MySQL, PostgreSQL, or MongoDB for managing reservation data.**
4. **APIs: Google Maps API for location-based services; Twilio for SMS reminders.**

4. BOOKS

1. ***Application.***
   - **Rinder, J. (2020).** ***Building a Scalable Web*** **Guides on building scalable systems like restaurant booking systems.**
2. **Reese, R. (2020).** ***Database Programming for the Web.***
   - **Covers the integration of databases into web-based projects.**

5. CODE REFERENCE

- **GitHub repositories with similar projects:**
  1. **Restaurant Management System on GitHub**
  2. **Open-source Booking Systems**

APPENDIX

CODING

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

public class RestaurantBookingForm {
    // A list to store booking details
    private static final List<String> bookingHistory = new ArrayList<>();

    public static void main(String[] args) {
        // Create the main frame
        JFrame frame = new JFrame("Restaurant Booking");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Set the frame to full-screen
        frame.setExtendedState(JFrame.MAXIMIZED_BOTH);
        frame.setLayout(new BorderLayout());

        // Add the custom background panel
        frame.add(new BackgroundPanel(), BorderLayout.CENTER);

        // Set frame visibility
        frame.setVisible(true);
    }
```

```
// Panel with background image
static class BackgroundPanel extends JPanel {
    private final Image backgroundImage;

    public BackgroundPanel() {
        // Load the background image (ensure the path is correct)
        backgroundImage = new ImageIcon("C:\\Users\\Monish S\\Pictures\\Screenshots\\images.jpeg").getImage();
        setLayout(new BorderLayout());

        // Add the main panel with the form and history
        add(createMainPanel(), BorderLayout.CENTER);
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        // Draw the background image, scaled to fit the panel
        g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);
    }
}

// Function to create the main panel
private static JPanel createMainPanel() {
    JPanel mainPanel = new JPanel();
    mainPanel.setOpaque(false); // Make the panel transparent to show the background image
    mainPanel.setLayout(new BorderLayout());

    // Add the form to the center of the main panel
    mainPanel.add(createFormPanel(), BorderLayout.CENTER);
    // Add the history section to the bottom of the main panel
    mainPanel.add(createHistoryPanel(), BorderLayout.SOUTH);

    return mainPanel;
}

// Function to create the form panel
private static JPanel createFormPanel() {
    JPanel formPanel = new JPanel();
    formPanel.setOpaque(false); // Make the panel transparent to show the background
    formPanel.setLayout(new GridBagLayout()); // Center everything

    GridBagConstraints gbc = new GridBagConstraints();
    gbc.insets = new Insets(10, 10, 10, 10); // Padding between elements
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.anchor = GridBagConstraints.CENTER;

    // Create components
    JLabel titleLabel = new JLabel("Restaurant Booking");
    titleLabel.setFont(new Font("Arial", Font.BOLD, 18));
    titleLabel.setForeground(Color.WHITE); // Make text visible on a background

    JLabel nameLabel = new JLabel("Name:");
    nameLabel.setForeground(Color.WHITE);
    JTextField nameField = new JTextField(20);

    JLabel emailLabel = new JLabel("Email:");
    emailLabel.setForeground(Color.WHITE);
    JTextField emailField = new JTextField(20);

    JLabel timeDateLabel = new JLabel("Time & Date:");
```

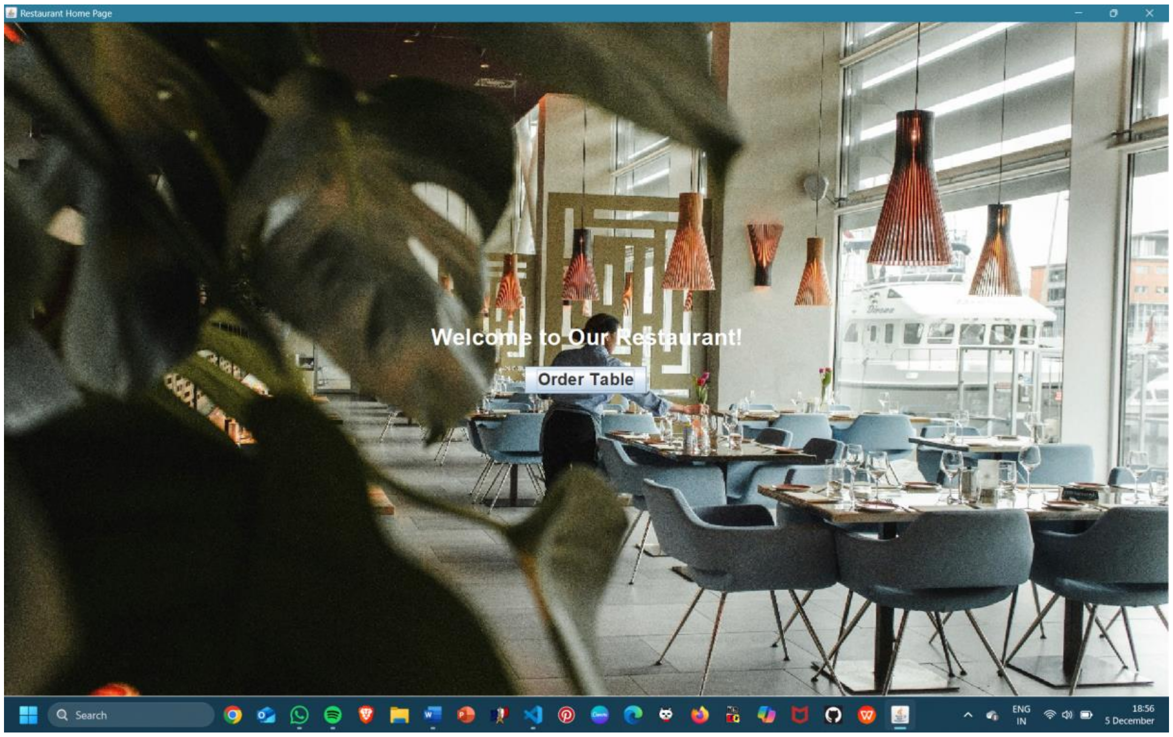
```
timeDateLabel.setForeground(Color.WHITE);

// Date Picker
JLabel dateLabel = new JLabel("Date:");
dateLabel.setForeground(Color.WHITE);
JSpinner dateSpinner = new JSpinner(new SpinnerDateModel());
JSpinner.DateEditor dateEditor = new JSpinner.DateEditor(dateSpinner, "dd-MM-yyyy");
dateSpinner.setEditor(dateEditor);

// Time Picker
JLabel timeLabel = new JLabel("Time:");
timeLabel.setForeground(Color.WHITE);
JSpinner timeSpinner = new JSpinner(new SpinnerDateModel());
JSpinner.DateEditor timeEditor = new JSpinner.DateEditor(timeSpinner, "HH:mm");
timeSpinner.setEditor(timeEditor);

JLabel durationLabel = new JLabel("Duration:");
durationLabel.setForeground(Color.WHITE);
JTextField durationField = new JTextField(20);

JLabel phoneLabel = new JLabel("Phone no:");
phoneLabel.setForeground(Color.WHITE);
JTextField phoneField = new JTextField(20);

JLabel addressLabel = new JLabel("Address:");
addressLabel.setForeground(Color.WHITE);
JTextField addressField = new JTextField(20);

JButton submitButton = new JButton("Submit");

// Add action listener to the Submit button
```

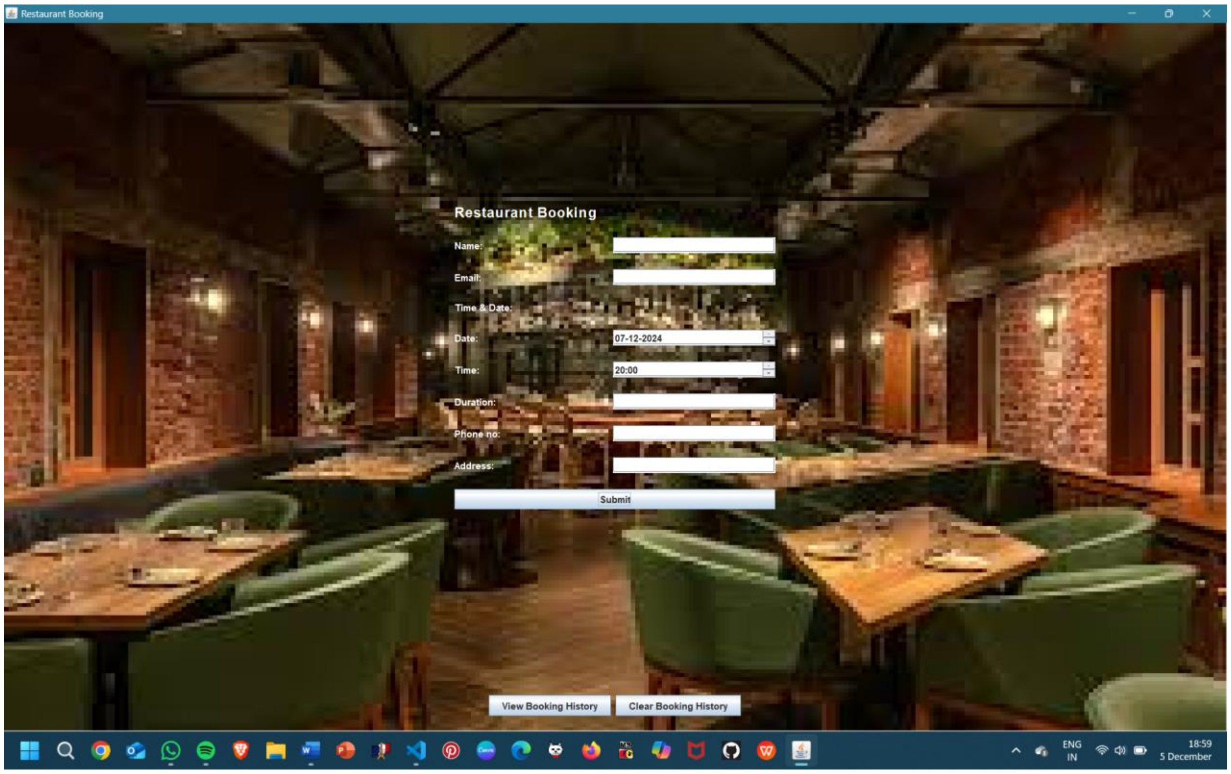
```
submitButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String name = nameField.getText();
        String email = emailField.getText();
        String date = new SimpleDateFormat("dd-MM-yyyy").format(dateSpinner.getValue());
        String time = new SimpleDateFormat("HH:mm").format(timeSpinner.getValue());
        String duration = durationField.getText();
        String phone = phoneField.getText();
        String address = addressField.getText();

        // Save booking details in history
        String bookingDetails = "Name: " + name + "\n" +
                "Email: " + email + "\n" +
                "Date: " + date + "\n" +
                "Time: " + time + "\n" +
                "Duration: " + duration + "\n" +
                "Phone: " + phone + "\n" +
                "Address: " + address;
        bookingHistory.add(bookingDetails);

        JOptionPane.showMessageDialog(null, "Booking Saved:\n" + bookingDetails);

        // Clear fields after submission
        nameField.setText("");
        emailField.setText("");
        durationField.setText("");
        phoneField.setText("");
        addressField.setText("");
    }
});
```

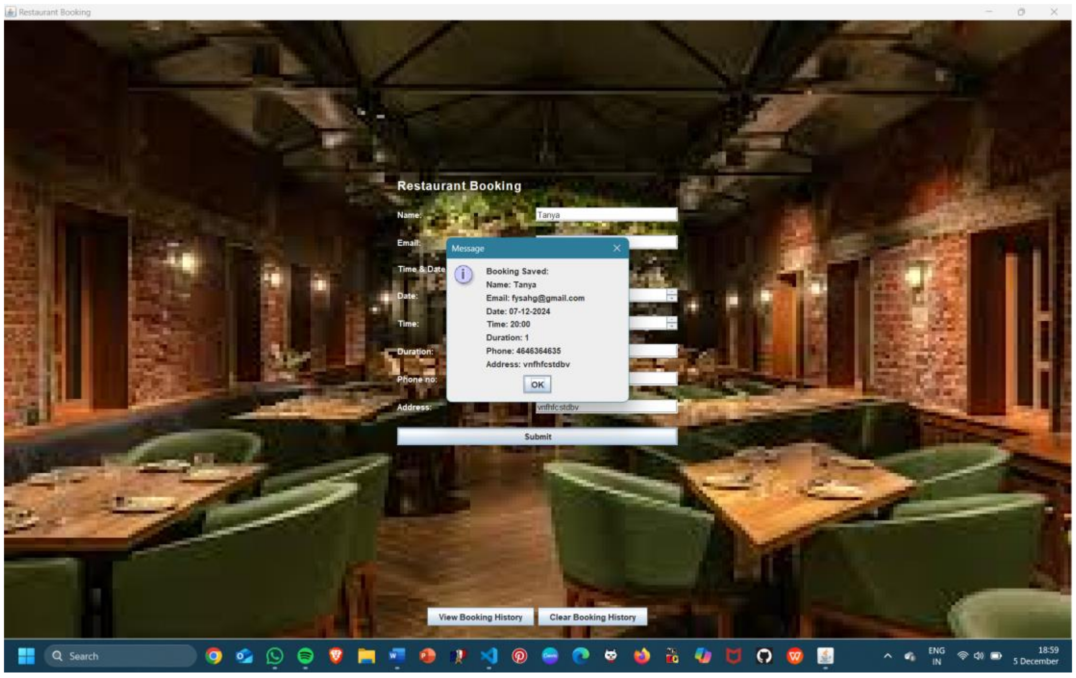
```
// Use GridBagConstraints to add components in rows
gbc.gridx = 0; gbc.gridy = 0;
formPanel.add(titleLabel, gbc);

gbc.gridx = 0; gbc.gridy = 1;
formPanel.add(nameLabel, gbc);
gbc.gridx = 1;
formPanel.add(nameField, gbc);

gbc.gridx = 0; gbc.gridy = 2;
formPanel.add(emailLabel, gbc);
gbc.gridx = 1;
formPanel.add(emailField, gbc);

gbc.gridx = 0; gbc.gridy = 3;
formPanel.add(timeDateLabel, gbc);

gbc.gridx = 0; gbc.gridy = 4;
formPanel.add(dateLabel, gbc);
gbc.gridx = 1;
formPanel.add(dateSpinner, gbc);

gbc.gridx = 0; gbc.gridy = 5;
formPanel.add(timeLabel, gbc);
gbc.gridx = 1;
formPanel.add(timeSpinner, gbc);

gbc.gridx = 0; gbc.gridy = 6;
formPanel.add(durationLabel, gbc);
gbc.gridx = 1;
```

```
    formPanel.add(durationField, gbc);

    gbc.gridx = 0; gbc.gridy = 7;
    formPanel.add(phoneLabel, gbc);
    gbc.gridx = 1;
    formPanel.add(phoneField, gbc);

    gbc.gridx = 0; gbc.gridy = 8;
    formPanel.add(addressLabel, gbc);
    gbc.gridx = 1;
    formPanel.add(addressField, gbc);

    gbc.gridx = 0; gbc.gridy = 9; gbc.gridwidth = 2;
    formPanel.add(submitButton, gbc);

    return formPanel;
}

// Function to create the history panel
private static JPanel createHistoryPanel() {
    JPanel historyPanel = new JPanel();
    historyPanel.setOpaque(false); // Transparent panel
    historyPanel.setPreferredSize(new Dimension(0, 50)); // Fixed height

    JButton viewHistoryButton = new JButton("View Booking History");
    JButton clearHistoryButton = new JButton("Clear Booking History");

    // Add action listener to the View Booking History button
    viewHistoryButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
                if (bookingHistory.isEmpty()) {
                    JOptionPane.showMessageDialog(null, "No bookings available.");
                } else {
                    StringBuilder historyText = new StringBuilder("Booking History:\n");
                    for (int i = 0; i < bookingHistory.size(); i++) {
                        historyText.append("Booking ").append(i + 1).append(":\n");
                        historyText.append(bookingHistory.get(i)).append("\n\n");
                    }
                    JOptionPane.showMessageDialog(null, historyText.toString());
                }
            }
        });

        // Add action listener to the Clear Booking History button
        clearHistoryButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                bookingHistory.clear();
                JOptionPane.showMessageDialog(null, "Booking history cleared!");
            }
        });

        // Add buttons to the panel
        historyPanel.add(viewHistoryButton);
        historyPanel.add(clearHistoryButton);

        return historyPanel;
    }
}
```

SCREENSHOT


Restaurant Home Page
Welcome to Our Restaurant!
Order Table
Search
ENG
IN
18:56
5 December


Restaurant Booking
Restaurant Booking
Name:
Email:
Time & Date:
Date:
07-12-2024
Time:
20:00
Duration:
Phone no:
Address:
Submit
View Booking History
Clear Booking History
ENG
IN
18:59
5 December



Name: Tanya
Email:
Time & Date
Date:
Time:
Duration:
Phone no:
Address: vnfhfcstdbv

Booking Saved:
Name: Tanya
Email: fysahg@gmail.com
Date: 07-12-2024
Time: 20:00
Duration: 1
Phone: 4646364635
Address: vnfhfcstdbv