

# **FASTFOOD MANAGEMENT SYSTEM**

## **A MINI PROJECT REPORT**

**Submitted by**

**MONESH G J    230701193**

**KAMALESH L   230701136**

**MANIKANDAN S   230701175**

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

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE**

**RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)**

**THANDALAM**

**CHENNAI-602105**

# **BONAFIDE CERTIFICATE**

Certified that this project report “**FASTFOOD BILLING  
SYSTEM**” is the bonafide work of “ **MONESH G J (230701193),  
KAMALESH L (230701136),MANIKANDAN S (230701175)**”

who carried out the project work under my supervision.

**Submitted for the Practical Examination held on \_\_\_\_\_**

**SIGNATURE**

**Mrs.K.MAHESHMEENA  
Assistant Professor,  
Science and Engineering,  
Rajalakshmi Engineering College,  
(Autonomous) Thandalam,  
Chennai-602 105**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

**ABSTRACT:**

The FASTFOOD Management System is a software project designed to streamline and manage the operations of an FASTFOOD or travel booking system by automating tasks such as flight scheduling, passenger reservations, ticket management, and staff allocation.

By integrating these functionalities, the system enhances efficiency and reduces human errors in managing FASTFOOD operations. It provides a user-friendly interface for administrators, staff, and customers to interact with the system, while a robust database manages critical data related to flights, bookings, users, and payments.

The backend logic ensures smooth handling of business workflows, including ticket bookings, cancellations, and updates, supported by error handling and reporting mechanisms that enhance system reliability and provide insights into operational issues. Designed for educational purposes, this project showcases how databasedriven applications can address real-world challenges in the aviation industry using widely used programming languages, frameworks, and tools.

# **TABLE OF CONTENTS**

## **CHAPTER 1-INTRODUCTION**

- 1.1 Introduction.....PG No.5**
- 1.2 Objectives.....PG No.6**
- 1.3 Modules.....PG No.6**

## **2-SURVEY OF TECHNOLOGIES**

- 2.1 Programming Language.....PG No.7**
- 2.2 Usage and Advantages.....PG No.8**

## **CHAPTER 3-REQUIREMENTS AND ANALYSIS**

- 3.1 Hardware Requirements.....PG No.9**
- 3.2 Software Requirements.....PG No.9**
- 3.3 Functional Requirements.....PG No.9**
- 3.4 Non-Functional Requirements.....PG No.10**
- 3.5 ER Diagram.....PG No.11-12**

## **CHAPTER 4-CODING**

- 4.1 Program Code.....PG No.13-45**

## **CHAPTER 5-RESULTS AND DISCUSSION**

- 5.1 Output.....PG No.46-49**

## **CHAPTER 6-CONCLUSION**

- 6.1 Conclusion..... .PG No.50**

## **CHAPTER 7-REFERNCES**

- 7.1 References.....PG No.50**

## **CHAPTER 1- INTRODUCTION**

## 1.1 INTRODUCTION

The **FASTFOOD Management System** is a comprehensive software solution designed to streamline and automate the operational, administrative, and customer-facing processes of an FASTFOOD. With the rapid growth of the aviation industry and the increasing complexity of managing large-scale operations, such systems play a pivotal role in ensuring efficiency, accuracy, and customer satisfaction.

FASTFOOD operations involve a diverse range of activities, including flight scheduling, ticket reservation, passenger and crew management, aircraft maintenance, and financial transactions. Traditionally, these tasks were handled manually or through disparate systems, leading to inefficiencies, delays, and potential errors. The FASTFOOD Management System addresses these challenges by integrating all essential functionalities into a unified platform, enabling real-time data processing, operational transparency, and seamless coordination across departments.

Key features of such systems include automated ticket booking, real-time flight status updates, secure payment processing, and advanced analytics for decision-making. They also incorporate robust security measures to protect sensitive customer data and comply with regulatory requirements. Modern FASTFOOD Management Systems leverage cutting-edge technologies such as cloud computing, artificial intelligence, and big data analytics to deliver scalable and adaptive solutions.

In addition to operational benefits, these systems enhance the passenger experience by providing intuitive interfaces for booking and managing travel, personalized recommendations, and prompt communication regarding flight changes or delays. For FASTFOODs, the system reduces operational costs, improves resource utilization, and ensures compliance with industry standards.

The FASTFOOD Management System thus serves as the backbone of efficient FASTFOOD operations, supporting growth in a competitive and fast-evolving industry.

## 1.2 OBJECTIVES:

The FASTFOOD Management System is designed to streamline and enhance the operational efficiency of an FASTFOOD. It enables seamless ticket booking and

reservation, offering passengers a user-friendly experience across multiple platforms. The system manages flight schedules, timings, and routes to optimize resources and minimize delays. It also maintains detailed customer records, including booking history and preferences, for personalized service. Secure and flexible payment methods ensure real-time transaction processing and accurate billing.

Passengers receive real-time updates on flight status, delays, and cancellations, enhancing their travel experience. The system supports backend operations such as crew management, aircraft maintenance, and operational analytics, improving overall reliability. Robust security features protect sensitive data, ensuring compliance with industry regulations. Additionally, the system is designed to scale, allowing for the addition of new features and adapting to increased user demands and business growth.

### 1.3 MODULES:

The FASTFOOD Management System consists of several interrelated modules designed to ensure efficient operations and enhance the user experience. The **User Management Module** handles passenger and staff registration, login, and profile management, with role-based access control for different users, including administrators and staff. The **Flight Management Module** manages flight scheduling, routes, and availability while handling updates such as delays, cancellations, and rescheduling.

The **Reservation and Ticketing Module** facilitates booking, cancellation, and rebooking, generating electronic tickets and allowing for seat selection. The **Payment Processing Module** integrates secure payment gateways for transactions, manages refunds, and maintains billing records. The **Customer Support Module** provides assistance via FAQs, complaint management, and live chat or automated responses to passenger queries.

The **Crew and Staff Management Module** schedules pilots, flight attendants, and ground crew, and tracks their availability, shifts, and performance. The **Aircraft Maintenance and Operations Module** ensures proper maintenance scheduling and compliance with safety regulations. The **Real-Time Notification Module** sends updates on flight statuses, gate changes, and promotional offers via email or SMS.

The **Admin Dashboard Module** provides tools for managing operations, analytics, and user data, while also generating reports for decision-making. The **Loyalty and Reward Program Module** tracks frequent flyer miles and reward points, offering redemption options and membership benefits. Lastly, the **Security Module** ensures robust user authentication, encryption, data protection, and fraud monitoring.

## CHAPTER 2-SURVEY OF TECHNOLOGIES

### 2.1 PROGRAMMING LANGUAGE

FASTFOOD Management Systems utilize various technologies to ensure efficiency, scalability, and user satisfaction. Here's a survey of the key technologies involved:

### 1. Programming Language

**Java:** Widely used for building robust backend systems due to its platform independence and scalability.

### 2. Web Development Frameworks

**Spring Boot (Java):** For creating scalable microservices. **3.**

### Database Management Systems (DBMS)

#### 4. Relational Databases (SQL):

**MySQL, PostgreSQL, or Oracle** are preferred for structured data storage, e.g., flight schedules, reservations, and passenger records.

## CHAPTER 3- REQUIREMENTS AND ANALYSIS

The **Requirements and Analysis** phase is critical in designing an efficient and userfriendly FASTFOOD Management System. Below is a detailed breakdown of the functional and non-functional requirements, followed by an analysis of the system.

### Requirements

#### 3.1 Functional Requirements

The functional requirements for the FASTFOOD Management System include various key features. **User Authentication and Authorization** ensures secure login with

rolebased access for passengers, administrators, and staff. **Flight Management** enables the addition, updating, and management of flight schedules, routes, and availability. The **Ticket Reservation and Booking** feature allows users to search for flights, book tickets, select seats, and cancel reservations. **Real-Time Updates** keep users informed about flight delays, cancellations, and gate changes.

The system includes **Payment Processing** that supports multiple payment methods like credit/debit cards, net banking, and wallets. **Customer Management** tracks user profiles, preferences, and booking history to enhance service. The **Admin Dashboard** allows administrators to generate reports, analyze performance metrics, and manage staff and flight operations. Lastly, **Feedback and Support** enables users to submit feedback and access customer support.

### 3.2. Non-Functional Requirements

The system should be capable of handling high transaction volumes during peak hours without delays, ensuring **performance**. It must also be **scalable** to accommodate increasing user traffic and future feature expansions. **Availability** is crucial, with a target uptime of 99.9% for critical functions like ticket booking. The system should implement robust **security** measures, including encryption, secure authentication, and regular audits to protect user data. **Usability** is important, with a user-friendly interface that is accessible across both web and mobile platforms. Lastly, the system must ensure **compliance** with regulatory requirements such as GDPR for data protection and relevant industry safety standards.

### 3.3 Analysis

#### 1. Problem Statement

Current FASTFOOD systems often face challenges such as inefficient booking processes, lack of real-time updates, and difficulty in scaling operations. This system aims to address these issues by automating processes, improving customer engagement, and ensuring operational efficiency.

#### 2. Feasibility Study

- **Technical Feasibility**
  - The required technologies, such as cloud platforms, databases, and web frameworks, are readily available and cost-effective.
- **Economic Feasibility**
  - Automating processes reduces manual intervention and operational costs over time.
- **Operational Feasibility**
  - The system is designed to integrate with existing infrastructure and workflows.



### 3. Stakeholder Analysis

- **Passengers**
  - Need an easy-to-use system for booking tickets, receiving updates, and managing their travel.
- **Administrators**
  - Need detailed reports and dashboards for operational decision-making.
- **Regulatory Authorities** ◦ Ensure compliance with safety and data protection standards.

### 4. System Flow Analysis

- **Input:** User inputs include search queries, booking details, and payment information.
- **Process:** The system validates, processes, and stores the data while updating flight availability.
- **Output:** Generates tickets, booking confirmations, and real-time notifications.

#### 5. Risk Analysis

- **Data Breaches:** Mitigated through encryption and multi-factor authentication.
- **System Downtime:** Addressed with redundant servers and load balancers.
- **Scalability Challenges:** Solved by leveraging cloud infrastructure.

By thoroughly addressing these requirements and analyzing the system's context, the FASTFOOD Management System ensures robust functionality, scalability, and customer satisfaction.

## 3.4 Hardware and Software Requirements

### Hardware Requirements:

#### Server Hardware:

1. **Processor:** Multi-core processor (Intel i7 or higher / AMD Ryzen 7 or higher)
2. **RAM:** Minimum of 16 GB (32 GB or more for high traffic systems)
3. **Storage:** At least 500 GB SSD (scalable storage as needed for database and logs)
4. **Network:** High-speed internet connection with backup for continuous operations
5. **Backup:** Regular and redundant backup systems for data security

#### Client Hardware:

1. **User Devices:** Desktop or laptop computers with at least 4 GB RAM and modern browsers
2. **Mobile Devices:** Smartphones or tablets (Android/iOS) for mobile app usage
3. **Printers:** For ticket printing, boarding passes, and other physical documentation

#### **Network Infrastructure:**

1. High-bandwidth network connections for handling online reservations and real-time updates
2. Load balancing and firewall to ensure smooth traffic management and security

#### **Software Requirements:**

##### **Operating System**

1. **Server Side:** Linux (Ubuntu, CentOS, or Debian) or Windows Server (depending on chosen tech stack)
2. **Client Side:** Windows, macOS, or Linux for desktop users; Android and iOS for mobile applications

##### **Database Management System (DBMS):**

1. **Relational Databases:** MySQL, PostgreSQL, or Oracle for managing

##### **Web and Application Development Frameworks:**

1. **Backend Frameworks:** Java (Spring Boot), Python (Django, Flask), or Node.js (Express)
2. **Frontend Frameworks:** React.js, Java FX for web interfaces

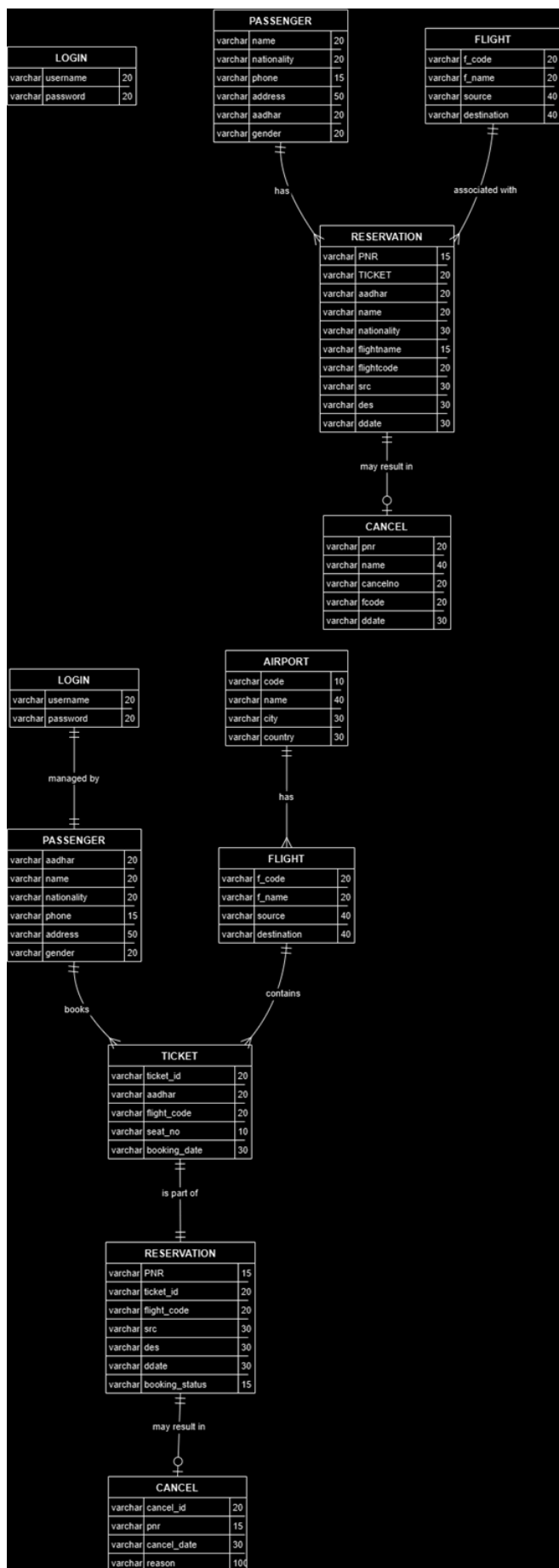
##### **Security Software:**

1. **SSL/TLS** for encryption of data transmission
2. **Firewall** and **IDS/IPS** systems for network security
3. **Multi-factor Authentication (MFA)** for secure user logins

##### **Development and Testing Tools:**

1. IDEs like Visual Studio Code, IntelliJ IDEA, or Eclipse for development
2. Version control with Git and GitHub/GitLab for collaboration
3. Automated testing tools like Selenium, JUnit, or Mocha for quality assurance

**3.5 ER DIAGRAM**



## CHAPTER 4-CODING

## 4.1 PROGRAM CODE

### 1.ADD CUSTOMER

```
package FASTFOODmanagementsystem;

import javax.swing.*;
import java.awt.*; import
java.awt.event.*;

public class AddCustomer extends JFrame implements ActionListener{

    JTextField tfname, tfphone, tfaadhar, tfnationality, tfaddress;

    JRadioButton rbmale, rbfemale;

    public AddCustomer() {
        getContentPane().setBackground(Color.WHITE);    setLayout(null);

        JLabel heading = new JLabel("ADD CUSTOMER DETAILS");
        heading.setBounds(220, 20, 500, 35);    heading.setFont(new
        Font("Tahoma", Font.PLAIN, 32));
        heading.setForeground(Color.BLUE); add(heading);

        JLabel lblname = new JLabel("Name"); lblname.setBounds(60,
        80, 150, 25); lblname.setFont(new Font("Tahoma",
        Font.PLAIN, 16)); add(lblname);

        tfname = new JTextField();
        tfname.setBounds(220, 80, 150, 25);    add(tfname);
```

```
JLabel lblnationality = new JLabel("Nationality");  
lblnationality.setBounds(60, 130, 150, 25);    lblnationality.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblnationality);
```

```
tfnationality = new JTextField();  
tfnationality.setBounds(220, 130, 150, 25);  
add(tfnationality);
```

```
JLabel lblaadhar = new JLabel("Aadhar Number");  
lblaadhar.setBounds(60, 180, 150, 25);    lblaadhar.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblaadhar);
```

```
tfaadhar = new JTextField();  
tfaadhar.setBounds(220, 180, 150, 25);  
add(tfaadhar);
```

```
JLabel lbladdress = new JLabel("Address");  
lbladdress.setBounds(60, 230, 150, 25);  
lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lbladdress);
```

```
tfaddress = new JTextField(); tfaddress.setBounds(220,  
230, 150, 25); add(tfaddress);
```

```
JLabel lblgender = new JLabel("Gender");  
lblgender.setBounds(60, 280, 150, 25);    lblgender.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblgender);
```

```
ButtonGroup gendergroup = new ButtonGroup();
```

```
rbmale = new JRadioButton("Male");  
rbmale.setBounds(220, 280, 70, 25);  
rbmale.setBackground(Color.WHITE);    add(rbmale);
```

```
rbfemale = new JRadioButton("Female");  
rbfemale.setBounds(300, 280, 70, 25);  
rbfemale.setBackground(Color.WHITE);    add(rbfemale);
```

```
gendergroup.add(rbmale);  
gendergroup.add(rbfemale);
```

```
JLabel lblphone = new JLabel("Phone");  
lblphone.setBounds(60, 330, 150, 25); lblphone.setFont(new  
Font("Tahoma", Font.PLAIN, 16)); add(lblphone);
```

```
tfphone = new JTextField();  
tfphone.setBounds(220, 330, 150, 25); add(tfphone);
```

```
JButton save = new JButton("SAVE");  
save.setBackground(Color.BLACK);  
save.setForeground(Color.WHITE);  
save.setBounds(220, 380, 150, 30);  
save.addActionListener(this);    add(save);
```

```
ImageIcon image = new  
ImageIcon(ClassLoader.getResource("FASTFOODmanagementsystem/icons/e  
mp.png"));
```

```
JLabel lblimage = new JLabel(image);  
lblimage.setBounds(450, 80, 280, 40);    add(lblimage);
```

```
setSize(900, 600);  
setLocation(300, 150);  
setVisible(true);  
}
```

```
public void actionPerformed(ActionEvent ae) {  
String name = tfname.getText();  
String nationality = tfnationality.getText();  
String phone = tfphone.getText();  
String address = tfaddress.getText();  
String aadhar = tfaadhar.getText();  
String gender = null;    if  
(rbmale.isSelected()) {    gender  
= "Male";  
    } else {  
gender = "Female";  
    }  
try {  
    Conn conn = new Conn();  
  
    String query = "insert into passenger values('"+name+"', '"+nationality+"',  
    '"+phone+"', '"+address+"', '"+aadhar+"', '"+gender+"')";  
  
    conn.s.executeUpdate(query);  
  
    JOptionPane.showMessageDialog(null, "Customer Details Added  
Successfully");  
  
    setVisible(false);  
} catch (Exception e) {  
    e.printStackTrace();  
}
```



```

    }
}

    public static void main(String[] args) {
new AddCustomer();

    }
}

2.boarding pass package
FASTFOODmanagementsystem;

import javax.swing.*;
import java.awt.*; import
java.awt.event.*; import
java.sql.*;

public class BoardingPass extends JFrame implements ActionListener {

    JTextField tfpnr;

    JLabel tfname, tfnationality, lblsrc, lbldest, labelfname, labelcode, labeldate;

    JButton fetchButton;

    public BoardingPass() {
getContentPane().setBackground(Color.WHITE);    setLayout(null);

        JLabel heading = new JLabel("AIR INDIA");
heading.setBounds(380, 10, 450, 35);    heading.setFont(new
Font("Tahoma", Font.PLAIN, 32));    add(heading);

        JLabel subheading = new JLabel("Boarding Pass");
subheading.setBounds(360, 50, 300, 30);

```

```
subheading.setFont(new Font("Tahoma", Font.PLAIN, 24));  
subheading.setForeground(Color.BLUE);    add(subheading);
```

```
JLabel lblaadhar = new JLabel("PNR DETAILS");  
lblaadhar.setBounds(60, 100, 150, 25); lblaadhar.setFont(new  
Font("Tahoma", Font.PLAIN, 16)); add(lblaadhar);
```

```
tfpnr = new JTextField(); tfpnr.setBounds(220,  
100, 150, 25); add(tfpnr);
```

```
fetchButton = new JButton("Enter");  
fetchButton.setBackground(Color.BLACK);  
fetchButton.setForeground(Color.WHITE);  
fetchButton.setBounds(380, 100, 120, 25);  
fetchButton.addActionListener(this);    add(fetchButton);
```

```
JLabel lblname = new JLabel("NAME");  
lblname.setBounds(60, 140, 150, 25);    lblname.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblname);
```

```
tfname = new JLabel();  
tfname.setBounds(220, 140, 150, 25);  
add(tfname);
```

```
JLabel lblnationality = new JLabel("NATIONALITY");  
lblnationality.setBounds(60, 180, 150, 25);    lblnationality.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblnationality);
```

```
tfnationality = new JLabel();  
tfnationality.setBounds(220, 180, 150, 25); add(tfnationality);
```

```
JLabel lbladdress = new JLabel("SRC");
```

```
lbladdress.setBounds(60, 220, 150, 25);
```

```
lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
add(lbladdress);
```

```
lblsrc = new JLabel(); lblsrc.setBounds(220,
```

```
220, 150, 25); add(lblsrc);
```

```
JLabel lblgender = new JLabel("DEST");
```

```
lblgender.setBounds(380, 220, 150, 25);    lblgender.setFont(new
```

```
Font("Tahoma", Font.PLAIN, 16));    add(lblgender);
```

```
lbldest = new JLabel();
```

```
lbldest.setBounds(540, 220, 150, 25);
```

```
add(lbldest);
```

```
JLabel lblfname = new JLabel("Flight Name");
```

```
lblfname.setBounds(60, 260, 150, 25);    lblfname.setFont(new
```

```
Font("Tahoma", Font.PLAIN, 16));    add(lblfname);
```

```
labelfname = new JLabel();
```

```
labelfname.setBounds(220, 260, 150, 25);
```

```
add(labelfname);
```

```
JLabel lblfcode = new JLabel("Flight Code");
```

```
lblfcode.setBounds(380, 260, 150, 25); lblfcode.setFont(new
```

```
Font("Tahoma", Font.PLAIN, 16)); add(lblfcode);
```

```
labelfcode = new JLabel();
```

```
labelfcode.setBounds(540, 260, 150, 25);
```

```
add(labelfcode);
```

```
JLabel lbldate = new JLabel("Date");
```

```
lbldate.setBounds(60, 300, 150, 25);    lbldate.setFont(new
```

```
Font("Tahoma", Font.PLAIN, 16));    add(lbldate);
```

```
labeldate = new JLabel();
```

```
labeldate.setBounds(220, 300, 150, 25);
```

```
add(labeldate);
```

```
ImageIcon i1 = new  
ImageIcon(ClassLoader.getResource("FASTFOODmanagementsystem/icons/a  
irindia .png"));
```

```
Image i2 = i1.getImage().getScaledInstance(300, 230,  
Image.SCALE_DEFAULT);
```

```
ImageIcon image = new ImageIcon(i2);
```

```
JLabel lblimage = new JLabel(image);
```

```
lblimage.setBounds(600, 0, 300, 300);
```

```
add(lblimage);
```

```
setSize(1000, 450);
```

```
setLocation(300, 150);
```

```
setVisible(true);
```

```
}
```

```
public void actionPerformed(ActionEvent ae) {
```

```
String pnr = tfpnr.getText();
```

```
try {
```

```
Conn conn = new Conn();
```

```
String query = "select * from reservation where PNR = '"+pnr+"'";
```

```
ResultSet rs = conn.s.executeQuery(query);
```

```
if (rs.next()) {
```

```
tfname.setText(rs.getString("name"));
```

```
tfnationality.setText(rs.getString("nationality"));
```

```
lblsrc.setText(rs.getString("src"));
```

```
lbldest.setText(rs.getString("des"));
```

```
labelfname.setText(rs.getString("flightname"));
```

```
labelfcode.setText(rs.getString("flightcode"));
```

```
labeldate.setText(rs.getString("ddate"));
```

```
    } else {
```

```
        JOptionPane.showMessageDialog(null, "Please enter correct PNR");
```

```
    }
```

```
    } catch (Exception e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
}
```

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

```
new BoardingPass();
```

```
}
```

```
}
```

3.book flight package

```
FASTFOODmanagementsystem;
```

```
import javax.swing.*;
```

```
import java.awt.*; import
```

```
java.awt.event.*; import
```

```
java.sql.*;
```

```
import com.toedter.calendar.JDateChooser; import
```

```
java.util.*;
```

```
public class BookFlight extends JFrame implements ActionListener{
```

```
    JTextField tfaadhar;
```

```
    JLabel tfname, tfnationality, tfaddress, labelgender, labelfname, labelfcode;
```

```
    JButton bookflight, fetchButton, flight;
```

```
    Choice source, destination;
```

```
    JDateChooser dcdate;
```

```
    public BookFlight() {
```

```
        getContentPane().setBackground(Color.WHITE);        setLayout(null);
```

```
        JLabel heading = new JLabel("Book Flight");
```

```
        heading.setBounds(420, 20, 500, 35);
```

```
        heading.setFont(new Font("Tahoma", Font.PLAIN, 32));
```

```
        heading.setForeground(Color.BLUE);        add(heading);
```

```
        JLabel lblaadhar = new JLabel("Aadhar"); lblaadhar.setBounds(60,
```

```
        80, 150, 25);
```

```
        lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16)); add(lblaadhar);
```

```
        tfaadhar = new JTextField();
```

```
        tfaadhar.setBounds(220, 80, 150, 25);
```

```
        add(tfaadhar);
```

```
        fetchButton = new JButton("Fetch User");
```

```
        fetchButton.setBackground(Color.BLACK);
```

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

```
fetchButton.setBounds(380, 80, 120, 25);  
fetchButton.addActionListener(this);    add(fetchButton);
```

```
JLabel lblname = new JLabel("Name");  
lblname.setBounds(60, 130, 150, 25);    lblname.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblname);
```

```
tfname = new JLabel();  
tfname.setBounds(220, 130, 150, 25);  
add(tfname);
```

```
JLabel lblnationality = new JLabel("Nationality");  
lblnationality.setBounds(60, 180, 150, 25);    lblnationality.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblnationality);
```

```
tfnationality = new JLabel(); tfnationality.setBounds(220,  
180, 150, 25); add(tfnationality);
```

```
JLabel lbladdress = new JLabel("Address");  
lbladdress.setBounds(60, 230, 150, 25); lbladdress.setFont(new  
Font("Tahoma", Font.PLAIN, 16)); add(lbladdress);
```

```
tfaddress = new JLabel(); tfaddress.setBounds(220,  
230, 150, 25);    add(tfaddress);
```

```
JLabel lblgender = new JLabel("Gender");  
lblgender.setBounds(60, 280, 150, 25);    lblgender.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblgender);
```

```
labelgender = new JLabel("Gender");  
labelgender.setBounds(220, 280, 150, 25);    add(labelgender);
```

```
JLabel lblsource = new JLabel("Source");  
lblsource.setBounds(60, 330, 150, 25);    lblsource.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblsource);
```

```
source = new Choice();  
source.setBounds(220, 330, 150, 25);  
add(source);
```

```
JLabel lbldest = new JLabel("Destination");  
lbldest.setBounds(60, 380, 150, 25);  
lbldest.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lbldest);
```

```
destination = new Choice(); destination.setBounds(220,  
380, 150, 25);    add(destination);
```

```
try {  
    Conn c = new Conn();  
    String query = "select * from flight";  
    ResultSet rs = c.s.executeQuery(query);  
  
    while(rs.next()) {  
source.add(rs.getString("source"));  
destination.add(rs.getString("destination"));  
    }  
  
} catch (Exception e) {  
    e.printStackTrace();
```



```
}
```

```
flight = new JButton("Fetch Flights");  
flight.setBackground(Color.BLACK);  
flight.setForeground(Color.WHITE);  
flight.setBounds(380, 380, 120, 25);  
flight.addActionListener(this);    add(flight);
```

```
JLabel lblfname = new JLabel("Flight Name");  
lblfname.setBounds(60, 430, 150, 25);  
lblfname.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lblfname);
```

```
labelfname = new JLabel();  
labelfname.setBounds(220, 430, 150, 25);  
add(labelfname);
```

```
JLabel lblfcode = new JLabel("Flight Code");  
lblfcode.setBounds(60, 480, 150, 25);    lblfcode.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblfcode);
```

```
labelfcode = new JLabel();  
labelfcode.setBounds(220, 480, 150, 25);  
add(labelfcode);
```

```
JLabel lbldate = new JLabel("Date of Travel");  
lbldate.setBounds(60, 530, 150, 25);    lbldate.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lbldate);
```

```
dcdate = new JDateChooser();  
dcdate.setBounds(220, 530, 150, 25);    add(dcdate);
```

```
        ImageIcon i1 = new  
        ImageIcon(ClassLoader.getResource("FASTFOODmanagementsystem/icons/d  
etails.  
jpg"));
```

```
        Image i2 = i1.getImage().getScaledInstance(450, 320,  
        Image.SCALE_DEFAULT);
```

```
        ImageIcon image = new ImageIcon(i2);
```

```
        JLabel lblimage = new JLabel(image);
```

```
        lblimage.setBounds(550, 80, 500, 410);
```

```
        add(lblimage);
```

```
        bookflight = new JButton("Book Flight");
```

```
        bookflight.setBackground(Color.BLACK);
```

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

```
        bookflight.setBounds(220, 580, 150, 25);
```

```
        bookflight.addActionListener(this);    add(bookflight);
```

```
        setSize(1100, 700);
```

```
        setLocation(200, 50);
```

```
        setVisible(true);
```

```
    }
```

```
    public void actionPerformed(ActionEvent ae) {
```

```
        if (ae.getSource() == fetchButton) {
```

```
            String aadhar = tfaadhar.getText();
```

```
        try {
```

```
            Conn conn = new Conn();
```

```
            String query = "select * from passenger where aadhar = '"+aadhar+"'";
```

```

        ResultSet rs = conn.s.executeQuery(query);

        if (rs.next()) {
tfname.setText(rs.getString("name"));
tfnationality.setText(rs.getString("nationality"));
tfaddress.setText(rs.getString("address"));
labelgender.setText(rs.getString("gender"));

        } else {
            JOptionPane.showMessageDialog(null, "Please enter correct aadhar");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
} else if (ae.getSource() == flight) {
    String src = source.getSelectedItemAt();
String dest = destination.getSelectedItemAt();

    try {
        Conn conn = new Conn();

        String query = "select * from flight where source = '"+src+"' and destination
= '"+dest+"'";

        ResultSet rs = conn.s.executeQuery(query);

        if (rs.next()) {
labelfname.setText(rs.getString("f_name"));
labelfcode.setText(rs.getString("f_code"));

        } else {
            JOptionPane.showMessageDialog(null, "No Flights Found");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

    } else {

        Random random = new Random();

        String aadhar = tfaadhar.getText();

        String name = tfname.getText();

        String nationality = tfnationality.getText();
        String flightname = labelfname.getText();

        String flightcode = labelfcode.getText();

        String src = source.getSelectedItemAt();

        String des = destination.getSelectedItemAt();

        String ddate = ((JTextField)
dcdate.getDateEditor().getUiComponent()).getText();

    try {

        Conn conn = new Conn();

        String query = "insert into reservation values('PNR-
"+random.nextInt(1000000)+"', 'TIC-"+random.nextInt(10000)+"', '"+aadhar+",
 '"+name+", '"+nationality+", '"+flightname+", '"+flightcode+", '"+src+", '"+des+",
 '"+ddate+"')";

        conn.s.executeUpdate(query);

        JOptionPane.showMessageDialog(null, "Ticket Booked Successfully");
        setVisible(false);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
new BookFlight();

```

```
}  
}
```

```
4.book flight package
```

```
FASTFOODmanagementsystem;
```

```
import javax.swing.*; import
```

```
java.awt.*; import java.awt.event.*;
```

```
import java.sql.*;
```

```
import com.toedter.calendar.JDateChooser; import
```

```
java.util.*;
```

```
public class BookFlight extends JFrame implements ActionListener{
```

```
    JTextField tfaadhar;
```

```
    JLabel tfname, tfnationality, tfaddress, labelgender, labelfname, labelcode;
```

```
    JButton bookflight, fetchButton, flight;
```

```
    Choice source, destination;
```

```
    JDateChooser dcdate;
```

```
    public BookFlight() {
```

```
        getContentPane().setBackground(Color.WHITE);        setLayout(null);
```

```
        JLabel heading = new JLabel("Book Flight");
```

```
        heading.setBounds(420, 20, 500, 35);
```

```
        heading.setFont(new Font("Tahoma", Font.PLAIN, 32));
```

```
        heading.setForeground(Color.BLUE);        add(heading);
```

```
        JLabel lblaadhar = new JLabel("Aadhar"); lblaadhar.setBounds(60,
```

```
        80, 150, 25);
```

```
        lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16)); add(lblaadhar);
```

```
tfaadhar = new JTextField();  
  
tfaadhar.setBounds(220, 80, 150, 25);  
  
add(tfaadhar);
```

```
  
fetchButton = new JButton("Fetch User");  
  
fetchButton.setBackground(Color.BLACK);  
  
fetchButton.setForeground(Color.WHITE);  
  
fetchButton.setBounds(380, 80, 120, 25);  
  
fetchButton.addActionListener(this);    add(fetchButton);
```

```
  
JLabel lblname = new JLabel("Name");  
  
lblname.setBounds(60, 130, 150, 25);    lblname.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblname);
```

```
  
tfname = new JLabel();  
  
tfname.setBounds(220, 130, 150, 25);  
  
add(tfname);
```

```
  
JLabel lblnationality = new JLabel("Nationality");  
  
lblnationality.setBounds(60, 180, 150, 25);    lblnationality.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblnationality);
```

```
  
tfnationality = new JLabel(); tfnationality.setBounds(220,  
180, 150, 25); add(tfnationality);
```

```
  
JLabel lbladdress = new JLabel("Address");  
  
lbladdress.setBounds(60, 230, 150, 25);  
  
lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));  
  
add(lbladdress);
```

```
tfaddress = new JLabel();  
tfaddress.setBounds(220, 230, 150, 25);  
add(tfaddress);
```

```
JLabel lblgender = new JLabel("Gender");  
lblgender.setBounds(60, 280, 150, 25);    lblgender.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblgender);
```

```
labelgender = new JLabel("Gender");  
labelgender.setBounds(220, 280, 150, 25);    add(labelgender);
```

```
JLabel lblsource = new JLabel("Source");  
lblsource.setBounds(60, 330, 150, 25);    lblsource.setFont(new  
Font("Tahoma", Font.PLAIN, 16));    add(lblsource);
```

```
source = new Choice();  
source.setBounds(220, 330, 150, 25);  
add(source);
```

```
JLabel lbldest = new JLabel("Destination");  
lbldest.setBounds(60, 380, 150, 25);  
lbldest.setFont(new Font("Tahoma", Font.PLAIN, 16));  
add(lbldest);
```

```
destination = new Choice(); destination.setBounds(220,  
380, 150, 25);    add(destination);
```

```
try {  
    Conn c = new Conn();  
    String query = "select * from flight";
```

```

ResultSet rs = c.s.executeQuery(query);

while(rs.next()) {
source.add(rs.getString("source"));
destination.add(rs.getString("destination"));
}

} catch (Exception e) {
e.printStackTrace();
}

flight = new JButton("Fetch Flights");
flight.setBackground(Color.BLACK);
flight.setForeground(Color.WHITE);
flight.setBounds(380, 380, 120, 25);
flight.addActionListener(this); add(flight);

JLabel lblfname = new JLabel("Flight Name");
lblfname.setBounds(60, 430, 150, 25);
lblfname.setFont(new Font("Tahoma", Font.PLAIN, 16));
add(lblfname);

labelfname = new JLabel();
labelfname.setBounds(220, 430, 150, 25);
add(labelfname);

JLabel lblfcode = new JLabel("Flight Code");
lblfcode.setBounds(60, 480, 150, 25);    lblfcode.setFont(new
Font("Tahoma", Font.PLAIN, 16));    add(lblfcode);

```



```
    labelcode = new JLabel();  
    labelcode.setBounds(220, 480, 150, 25);  
    add(labelcode);
```

```
    JLabel lbldate = new JLabel("Date of Travel");  
    lbldate.setBounds(60, 530, 150, 25);    lbldate.setFont(new  
    Font("Tahoma", Font.PLAIN, 16));    add(lbldate);
```

```
    dcddate = new JDateChooser();  
    dcddate.setBounds(220, 530, 150, 25);    add(dcddate);
```

```
    ImageIcon i1 = new  
    ImageIcon(ClassLoader.getResource("FASTFOODmanagementsystem/icons/d  
    etails.  
    jpg"));
```

```
    Image i2 = i1.getImage().getScaledInstance(450, 320,  
    Image.SCALE_DEFAULT);
```

```
    ImageIcon image = new ImageIcon(i2);
```

```
    JLabel lblimage = new JLabel(image);  
    lblimage.setBounds(550, 80, 500, 410);  
    add(lblimage);
```

```
    bookflight = new JButton("Book Flight");  
    bookflight.setBackground(Color.BLACK);  
    bookflight.setForeground(Color.WHITE);  
    bookflight.setBounds(220, 580, 150, 25);  
    bookflight.addActionListener(this);    add(bookflight);
```

```
    setSize(1100, 700);  
    setLocation(200, 50);  
    setVisible(true);  
    }
```

```

    public void actionPerformed(ActionEvent ae) {
if (ae.getSource() == fetchButton) {

    String aadhar = tfaadhar.getText();

try {

    Conn conn = new Conn();

    String query = "select * from passenger where aadhar = '"+aadhar+"'";

    ResultSet rs = conn.s.executeQuery(query);

        if (rs.next()) {
tfname.setText(rs.getString("name"));
tfnationality.setText(rs.getString("nationality"));
tfaddress.setText(rs.getString("address"));
labelgender.setText(rs.getString("gender"));

        } else {

            JOptionPane.showMessageDialog(null, "Please enter correct aadhar");

        }

    } catch (Exception e) {

        e.printStackTrace();

    }

    } else if (ae.getSource() == flight) {

        String src = source.getSelectedItemAt();

String dest = destination.getSelectedItemAt();

        try {

            Conn conn = new Conn();

            String query = "select * from flight where source = '"+src+"' and destination
= '"+dest+"'";

```

```

        ResultSet rs = conn.s.executeQuery(query);

        if (rs.next()) {
labelfname.setText(rs.getString("f_name"));
labelfcode.setText(rs.getString("f_code"));

            } else {

                JOptionPane.showMessageDialog(null, "No Flights Found");

            }

        } catch (Exception e) {

            e.printStackTrace();

        }

    } else {

        Random random = new Random();

        String aadhar = tfaadhar.getText();
        String name = tfname.getText();

        String nationality = tfnationality.getText();

        String flightname = labelfname.getText();

        String flightcode = labelfcode.getText();

        String src = source.getSelectedItem();

        String des = destination.getSelectedItem();

        String ddate = ((JTextField)
dcdate.getDateEditor().getUiComponent()).getText();

    try {

        Conn conn = new Conn();

        String query = "insert into reservation values('PNR-
"+random.nextInt(1000000)+"', 'TIC-"+random.nextInt(10000)+"', '"+aadhar+",
 '"+name+"', '"+nationality+"', '"+flightname+"', '"+flightcode+"', '"+src+"', '"+des+"',
 '"+ddate+"')";

        conn.s.executeUpdate(query);

```

```

        JOptionPane.showMessageDialog(null, "Ticket Booked Successfully");
        setVisible(false);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

```

    public static void main(String[] args) {
        new BookFlight();
    }
}

```

5.login package

FASTFOODmanagementsystem;

```

import javax.swing.*;
import java.awt.*; import
java.awt.event.*; import
java.sql.*;

```

```

public class Login extends JFrame implements ActionListener{

```

```

    JButton submit, reset, close;

```

```

    JTextField tfusername;

```

```

    JPasswordField tfpassword;

```

```

    public Login() {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
    }
}

```

```
JLabel lblusername = new JLabel("Username");

lblusername.setBounds(20, 20, 100, 20);    add(lblusername);


tfusername = new JTextField();

tfusername.setBounds(130, 20, 200, 20);    add(tfusername);


JLabel lblpassword = new JLabel("Password");

lblpassword.setBounds(20, 60, 100, 20);    add(lblpassword);


tfpassword = new JPasswordField();    tfpassword.setBounds(130, 60, 200,
20);    add(tfpassword);


reset = new JButton("Reset");

reset.setBounds(40, 120, 120, 20);

reset.addActionListener(this);    add(reset);


submit = new JButton("Submit");

submit.setBounds(190, 120, 120, 20);

submit.addActionListener(this);    add(submit);


close = new JButton("Close");

close.setBounds(120, 160, 120, 20);

close.addActionListener(this);    add(close);


setSize(400, 250);

setLocation(600, 250);

setVisible(true);

}
```

```

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == submit) {
        String username = tfusername.getText();
        String password = tfpassword.getText();

        try {
            Conn c = new Conn();

            String query = "select * from login where username = '"+username+"' and
password = '"+password+"'";

            ResultSet rs = c.s.executeQuery(query);
            if
(rs.next()) {                new
Home();
setVisible(false);
            } else {
                JOptionPane.showMessageDialog(null, "Invalid Username or
Password");
                setVisible(false);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    } else if (ae.getSource() == close) {
setVisible(false);
    } else if (ae.getSource() == reset) {
tfusername.setText("");          tfpassword.setText("");
    }
}
}

```

```
    public static void main(String[] args) {  
        new Login();  
    }  
}
```

5.home page

```
package FASTFOODmanagementsystem;
```

```
import javax.swing.*;
```

```
import java.awt.*; import
```

```
java.awt.event.*;
```

```
public class Home extends JFrame implements ActionListener {
```

```
    public Home() {  
        setLayout(null);
```

```
        // Background Image
```

```
        ImageIcon i1 = new  
        ImageIcon(ClassLoader.getResource("FASTFOODmanagementsystem/icons/f  
        ront.jp  
        g"));
```

```
        JLabel image = new JLabel(i1);  
        image.setBounds(0, 0, 1600, 800);    add(image);
```

```
        // Heading
```

```
        JLabel heading = new JLabel("GT FASTFOODS");  
        heading.setBounds(500, 40, 1000, 40);  
        heading.setForeground(Color.black);    heading.setFont(new  
        Font("arial", Font.PLAIN, 36));    image.add(heading);
```

```
        // Menu Bar
```

```
JMenuBar menubar = new JMenuBar();  
setJMenuBar(menubar);
```

```
// Details Menu
```

```
JMenu details = new JMenu("Details");  
menubar.add(details);
```

```
JMenuItem flightDetails = new JMenuItem("Flight Details");  
flightDetails.addActionListener(this);    details.add(flightDetails);
```

```
JMenuItem customerDetails = new JMenuItem("Add Customer Details");  
customerDetails.addActionListener(this);    details.add(customerDetails);
```

```
JMenuItem bookFlight = new JMenuItem("Book Flight");  
bookFlight.addActionListener(this);    details.add(bookFlight);
```

```
JMenuItem journeyDetails = new JMenuItem("Journey Details");  
journeyDetails.addActionListener(this);    details.add(journeyDetails);
```

```
JMenuItem ticketCancellation = new JMenuItem("Cancel Ticket");  
ticketCancellation.addActionListener(this);    details.add(ticketCancellation);
```

```
// Ticket Menu
```

```
JMenu ticket = new JMenu("Ticket");  
menubar.add(ticket);
```

```
// Boarding Pass Menu Item
```

```
JMenuItem boardingPass = new JMenuItem("Boarding Pass");  
boardingPass.addActionListener(this); // Add ActionListener to Boarding Pass  
ticket.add(boardingPass);
```



```

// Set Frame Properties

setExtendedState(JFrame.MAXIMIZED_BOTH);

setVisible(true);
}

@Override public void
actionPerformed(ActionEvent ae) {

    String text = ae.getActionCommand();

    if (text.equals("Add Customer Details")) {
new AddCustomer();

    } else if (text.equals("Flight Details")) {
new FlightInfo();

    } else if (text.equals("Book Flight")) {
new BookFlight();

    } else if (text.equals("Journey Details")) {
new JourneyDetails();

    } else if (text.equals("Cancel Ticket")) {
new Cancel();

    } else if (text.equals("Boarding Pass")) {        new BoardingPass(); //
Create and display the Boarding Pass window

    }

}

public static void main(String[] args) {
new Home();

}

}

```

6.Flight details:

```
package FASTFOODmanagementsystem;
```

```
import javax.swing.*; import
java.awt.*; import java.sql.*;
import net.proteanit.sql.DbUtils;

public class FlightInfo extends JFrame{

    public FlightInfo() {

        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JTable table = new JTable();

        try {

            Conn conn = new Conn();

            ResultSet rs = conn.s.executeQuery("select * from flight");
            table.setModel(DbUtils.resultSetToTableModel(rs));

        } catch(Exception e) {
            e.printStackTrace();
        }

        JScrollPane jsp = new JScrollPane(table);
        jsp.setBounds(0, 0, 800, 500);    add(jsp);

        setSize(800, 500);
        setLocation(400, 200);
        setVisible(true);
    }
}
```

```
    public static void main(String[] args) {  
new FlightInfo();  
    }  
}
```

7.Connection:

```
package FASTFOODmanagementsystem;
```

```
import java.sql.*;
```

```
public class Conn {
```

```
    Connection c;
```

```
    Statement s;
```

```
    public Conn() {
```

```
        try {
```

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

```
            c =
```

```
DriverManager.getConnection("jdbc:mysql:///FASTFOODmanagementsystem",  
"root", "0409");            s =
```

```
c.createStatement();        }
```

```
catch (Exception e) {
```

```
    e.printStackTrace();
```

```
    }
```

```
    }
```

```
}
```

# CHAPTER 5-RESULTS

Fast Food POS System - INR

Menu

Burger - ₹150.00

Fries - ₹80.00

Coke - ₹50.00

Pizza - ₹200.00

Pasta - ₹180.00

Noodles - ₹120.00

Milkshake - ₹120.00

Ice Cream - ₹100.00

Lemonade - ₹70.00

Hotdog - ₹110.00

Sandwich - ₹130.00

French Fries - ₹90.00

Onion Rings - ₹100.00

Garlic Bread - ₹80.00

Soda - ₹60.00

Vegetable Burger - ₹140.00

Cheese Burger - ₹180.00

Chocolate Sundae - ₹150.00

Fruit Salad - ₹90.00

Tea - ₹40.00

Coffee - ₹60.00

Fried Chicken - ₹220.00

Veg Wrap - ₹160.00

Chicken Wings - ₹200.00

Chocolate Milkshake - ₹140.00

Sub-Total: ₹380.00

Tax (18%): ₹68.40

Grand Total: ₹448.40

Add to Order

Remove from Order

Checkout

Order

Pizza - ₹200.00

Milkshake - ₹120.00

Coffee - ₹60.00

Menu

Burger - ₹150.00

Fries - ₹80.00

Coke - ₹50.00

Pizza - ₹200.00

Pasta - ₹180.00

Noodles - ₹120.00

Milkshake - ₹120.00

Ice Cream - ₹100.00

Lemonade - ₹70.00

Hotdog - ₹110.00

Sandwich - ₹130.00

French Fries - ₹90.00

Onion Rings - ₹100.00

Garlic Bread - ₹80.00

Soda - ₹60.00

Vegetable Burger - ₹140.00

Cheese Burger - ₹180.00

Chocolate Sundae - ₹150.00

Fruit Salad - ₹90.00

Tea - ₹40.00

Coffee - ₹60.00

Fried Chicken - ₹220.00

Veg Wrap - ₹160.00

Chicken Wings - ₹200.00

Chocolate Milkshake - ₹140.00

Sub-Total: ₹0.00

Tax (18%): ₹0.00

Grand Total: ₹0.00

Add to Order

Remove from Order

Checkout

Order

Fast Food POS System - INR

Menu

Burger - ₹150.00

Fries - ₹80.00

Coke - ₹50.00

Pizza - ₹200.00

Pasta - ₹180.00

Noodles - ₹120.00

Milkshake - ₹120.00

Ice Cream - ₹100.00

Lemonade - ₹70.00

Hotdog - ₹110.00

Sandwich - ₹130.00

French Fries - ₹90.00

Onion Rings - ₹100.00

Garlic Bread - ₹80.00

Soda - ₹60.00

Vegetable Burger - ₹140.00

Cheese Burger - ₹180.00

Chocolate Sundae - ₹150.00

Fruit Salad - ₹90.00

Tea - ₹40.00

Coffee - ₹60.00

Fried Chicken - ₹220.00

Veg Wrap - ₹160.00

Chicken Wings - ₹200.00

Chocolate Milkshake - ₹140.00

Sub-Total: ₹720.00

Tax (18%): ₹129.60

Grand Total: ₹849.60

Add to Order

Remove from Order

Checkout

Order

Burger - ₹150.00

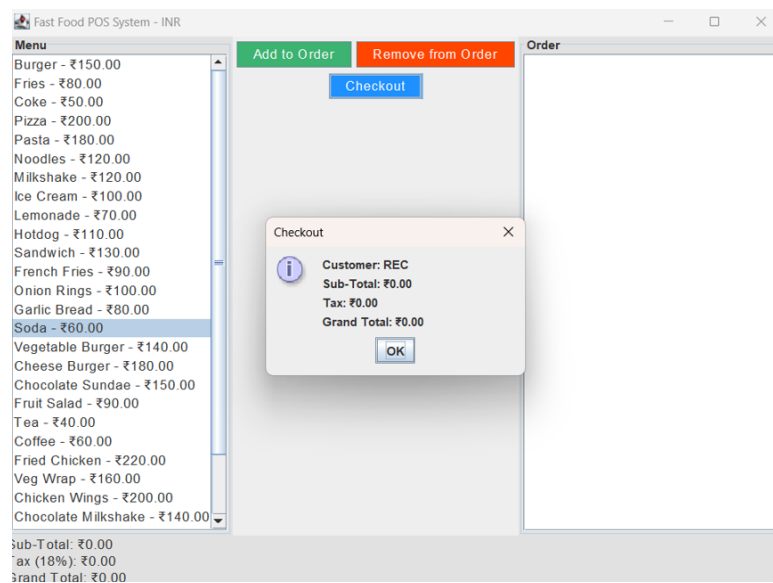
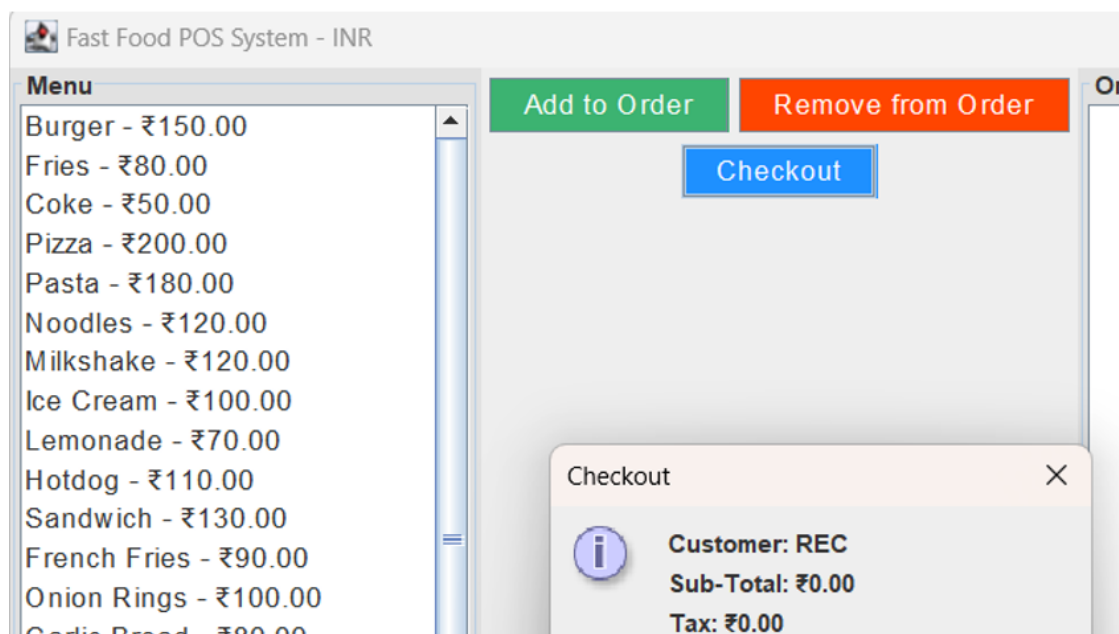
Pizza - ₹200.00

Milkshake - ₹120.00

Sandwich - ₹130.00

Coffee - ₹60.00

Soda - ₹60.00



```
mysql> use airlinemanagementsystem;
Database changed
mysql> show airlinemanagementsystem;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'a
irlinemanagementsystem' at line 1
mysql> show tables;
+-----+
| Tables_in_airlinemanagementsystem |
+-----+
| cancel                             |
| flight                             |
| login                              |
| passenger                           |
| reservation                         |
+-----+
5 rows in set (0.01 sec)

mysql> select * from cancel;
+-----+
| pnr      | name  | cancelno | fcode | ddate      |
+-----+
| PNR-307741 | monish | 823565   | 5054  | 21 Nov 2024 |
+-----+
1 row in set (0.01 sec)
```

```
mysql> show databases;
```

Database
airlinemanagementsystem
ebs
information_schema
library_hust_management
mysql
performance_schema
sys
testdb

```
--> ;
```

id	customer_name	total_amount	order_date
1	monesh	377.60	2024-11-11 21:14:50
2	ajai	566.40	2024-11-11 21:15:00
3	madhan	354.00	2024-11-11 21:15:15
4	kamalesh	495.60	2024-11-11 21:17:53
5	mani	153.40	2024-11-11 21:23:40
6	mani	0.00	2024-11-12 08:13:42
7	ikram	59.00	2024-11-12 08:23:40
9	Kavish	4094.60	2024-11-12 09:27:47
10	Harini	483.80	2024-11-12 09:28:46
11	hasan	601.80	2024-11-12 09:42:13
12	mhjahjjiddfv	413.00	2024-11-12 10:01:25
13	REC	0.00	2024-11-19 22:29:46
14	Meghana	495.60	2024-11-23 10:56:04

13 rows in set (0.00 sec)

## **CHAPTER 6-CONCLUSION**

An FASTFOOD Management System plays a crucial role in the efficient operation of FASTFOODs, encompassing various aspects such as reservations, ticketing, flight scheduling, passenger management, and resource allocation. By automating these processes, the system improves operational efficiency, enhances customer satisfaction, and ensures better management of resources. The integration of advanced technologies and real-time data processing further helps FASTFOODs in making informed decisions and adapting to changing conditions in the aviation industry. Ultimately, the FASTFOOD Management System not only supports operational goals but also helps FASTFOODs stay competitive in a fast-paced and dynamic market

## **CHAPTER 7-REFERENCES**

- **Java Swing**- <https://www.geeksforgeeks.org/introduction-to-java-swing/>
- **MySql**- <https://www.w3schools.com/MySQL/default.asp>
- **JDBC**- <https://www.javatpoint.com/example-to-connect-to-the-mysql-database>