**WELLCOME TO**

LuxuryStay

USER’S GUIDE:

Aptech North Nazimabad | ADSE I | NOV 14,2025



|  |  |  |
| --- | --- | --- |
| **ADVISOR/TEACHER/MENTOR** | SIR OWAIS AHMED | |
| **BATCH CODE** | 2303G1 | |
| **SEMESTER** | ADSE I | |
| **SERIAL NUMBER** | **ENROLLMENT NUMBER** | **STUDENT NAME** |
| 1 | Student1483150 | MUHAMMAD HASSAN |
| 2 | Student1485457 | MUHAMMAD ALI |
| 3 | Student1429988 | SAMIULLAH |
| 4 | Student1485459 | MUDASSAR AHMED |

eProject Design

# DEFINITION OF Dfd’s

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

The Step by Step DFD’s Diagram is shown below.

### DFD’S Diagram diagram fully structured view

Dashbaord

ROOM

Dining

Booking

TASKs

Admin Panel

# PROCESS DIAGRAM

# PROCESS DIAGRAM DEFINITION

Process diagrams, called “flow diagrams” by TOGAF, are used to model the sequence of activities within a process. Process modeling formalizes practices and describes the manner in which they should take place.

Flow diagrams represent process participants, activity sequences, information exchanged during a process, and trigger events. Processes can also detail the different checks, choices, and coordinations that exist within a sequence of activities.

A *process diagram* consists of activities, events, and gateways, which a sequence flow puts in a *flow sequence*. Activities, events, and gateways are summarized under the term *flow object*.

Process diagrams, called “flow diagrams” by TOGAF, are **used to model the sequence of activities within a process**. Process modeling formalizes practices and describes the manner in which they should take place

# 

# Problem Statement

# Background

# &quot;LuxuryStay Hospitality&quot;, a high-end hotel chain known for its exceptional service and luxurious

# accommodations. With a network of hotels in various locations, LuxuryStay aims to

# revolutionize its operational efficiency by implementing a cutting-edge Hotel Management

# System (HMS).

# LuxuryStay envisions a modern, scalable, and user-friendly Hotel Management System to

# transform its operations. The system will streamline workflows, enhance guest experiences, and

# provide insightful data for strategic decision-making.

# Functional Requirements

# User Management:

# Admin Dashboard: An admin panel to manage user roles and permissions.

# Staff Profiles: Ability to create, modify, and deactivate staff accounts with different access levels

# (manager, receptionist, housekeeping, etc.).

# Guest Profiles: Creation and management of guest profiles with details like personal

# information, contact, preferences, etc.

# Room Management:

# Room Inventory: Maintain a comprehensive inventory of rooms with details on room types,

# availability, status (cleaning, occupied, available), and pricing.

# Room Booking: Allow staff to reserve rooms, check availability, assign rooms to guests, and

# manage bookings efficiently.

# Room Status Updates: Enable real-time updates on room status (cleaning, maintenance, vacant,

# occupied) for better coordination among staff.

# Reservation and Check-in/out:

# Reservation System: Facilitate booking management, allowing guests to make reservations

# online or through staff, providing confirmation details.

# Check-in/Check-out: Smooth check-in and check-out processes with automated procedures for

# room allocation, key issuance, billing, and updating room status.

# Billing and Invoicing:

# Billing System: Generate accurate bills based on room rates, additional services (food, laundry,

# etc.), and duration of stay.

# Invoicing: Provide guests with detailed invoices that can be printed or emailed, including

# breakdowns of charges.

# Housekeeping and Maintenance:

# Housekeeping Management: Allow housekeeping staff to view room status, schedule cleaning

# tasks, mark tasks as completed, and report maintenance issues.

# Maintenance Requests: Enable guests or staff to report maintenance issues and track their

# resolution status.

# Reporting and Analytics:

# Reporting Dashboard: Provide management with insights via customizable reports on

# occupancy rates, revenue, guest feedback, etc.

# Analytics: Utilize data analytics to forecast demand, optimize pricing, and improve service

# offerings.

# Security and Compliance:

# User Authentication: Implement secure login/logout mechanisms for staff and guests with role-

# based access control.

# Feedback and Guest Services:

# Feedback Mechanism: Provide a way for guests to leave feedback and ratings, helping in

# improving services.

# Additional Services: Allow guests to request additional services like room service, wake-up calls,

# transportation, etc.

# System Administration:

# System Settings: Admin controls for system configurations, including setting room rates,

# defining policies, managing taxes, etc.

# System Notifications: Alerts and notifications for staff regarding bookings, maintenance

# requests, and other critical updates.

# Non-Functional Requirements

# Performance:

# Response Time: The application should respond to user interactions within 1-2 seconds for most

# operations.

# Scalability: The system should be able to handle a growing number of users and data without

# significant performance degradation.

# Concurrent Users: The application should support hundreds of concurrent users without

# performance bottlenecks.

# Security:

# Data Encryption: All sensitive user data, including passwords and personal information, must be

# securely encrypted during storage and transmission.

# Authentication: User authentication should be secure and use industry-standard practices to

# prevent unauthorized access.

# Authorization: Access control must be implemented to ensure users can only access their own

# data or public data, as per their settings.

# Privacy:

# Data Privacy: The application must comply with data privacy regulations such as GDPR, ensuring

# user data is handled and stored with care.

# User Consent: Users should have control over the data they share and provide informed

# consent for data processing and sharing.

# Reliability:

# Uptime: The application should aim for a minimum of 99% uptime, with scheduled maintenance

# communicated in advance.

# Data Backup: Regular automated data backups must be performed to prevent data loss in case

# of system failures.

# Usability:

# User Interface Design: The application should have an intuitive, user-friendly interface with

# consistent navigation and a responsive design that works on various devices.

# Accessibility: The application must adhere to accessibility standards (e.g., WCAG) to ensure it is

# usable by individuals with disabilities.

# Compatibility:

# Cross-Browser Compatibility: The application should function correctly on popular web

# browsers, including Chrome, Firefox, Safari, and Edge.

# Mobile Compatibility: The application should be responsive and work well on various mobile

# devices and screen sizes.

# Scalability:

# Horizontal Scalability: The architecture should support horizontal scaling to accommodate

# increased user loads as the user base grows.

# Performance Monitoring:

# Logging and Monitoring: The system should include logging and monitoring tools to track

# application performance, errors, and user activity for debugging and analysis.

# Testing and Quality Assurance:

# Test Coverage: A comprehensive test suite should be maintained, covering unit testing,

# integration testing, and end-to-end testing.

# Security Testing: Regular security assessments, including penetration testing, must be

# conducted to identify vulnerabilities.

# Documentation:

# User Documentation: Provide user guides, FAQs, and tutorials to help users understand and

# navigate the application.

# Developer Documentation: Maintain developer documentation to assist in further development

# and maintenance.

# Video: Provide video displaying complete working of the application.

# REQUIREMENTS OF THE PROJECT WHICH ASSIGNED US FROM APTECH

# Welcome to eProjects

Welcome to eProjects

Dear Student,

It gives us immense pleasure to address you about eProjects. This is a new way of learning, which would value add the existing course material that has been provided to you.

Kindly find attached the eProject Specification.

In case of any queries do write back to us.

**Your Project/Assignment Start date is:         30-Oct-2025**

**Your Project/Assignment End date is:           29-Nov-2025**

You need to complete your project and submit the following by the end-date.

1. **Submit a video clip demonstrating the working of your Project.**

1. Working Application

a.        Source Code

b.        Compiled Code

1. eProject Report

n       eProject Report should comprise-

q      Acknowledgements

q      eProject Synopsis

q      eProject Analysis

q      eProject Design

n       DFD’s

n       FlowCharts

n       Process Diagrams

n       Database Design / Structure

q      Screen Shot’s

q      Source Code with Comments

q      User Guide

q      Developer’s Guide

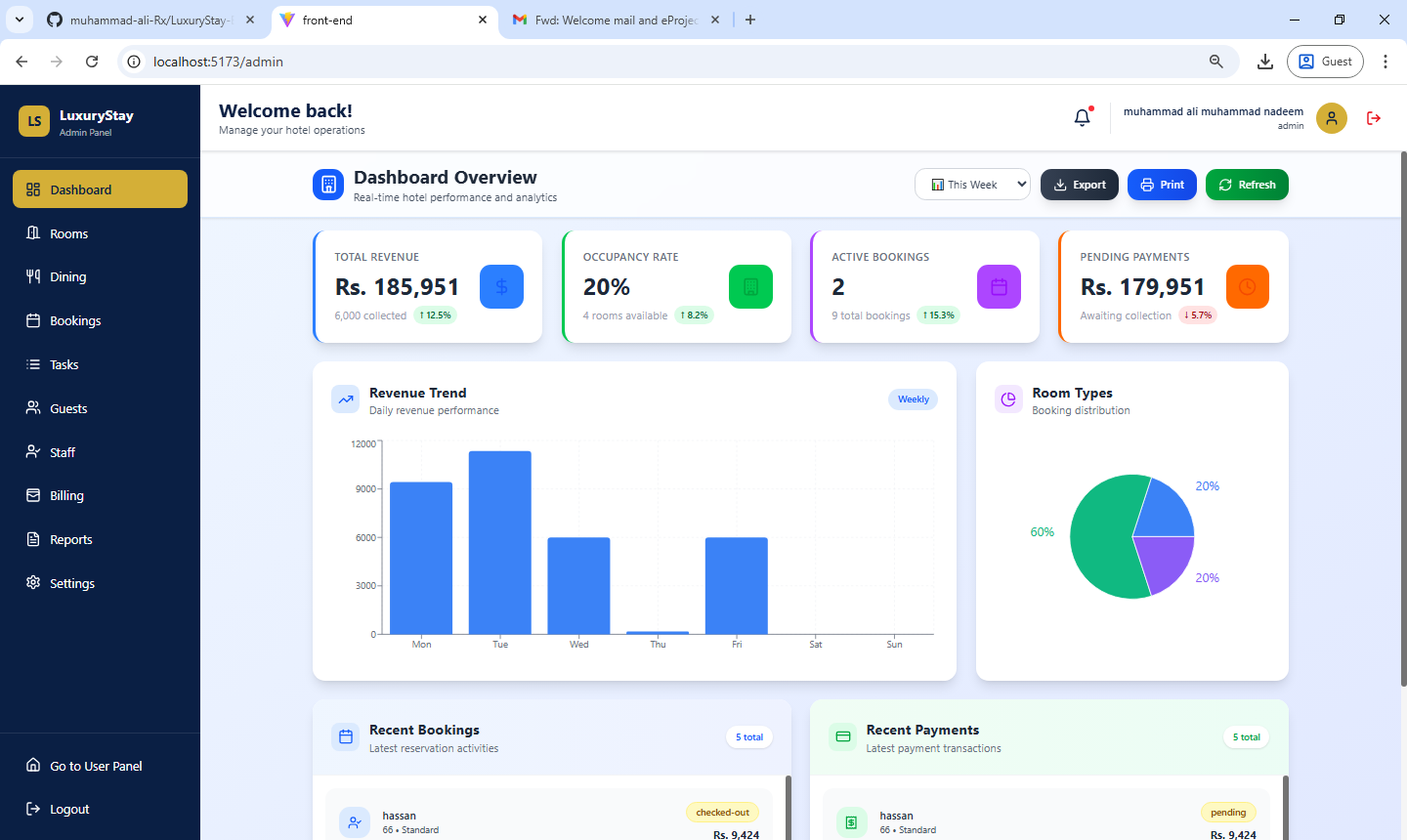
n       Module Descriptions

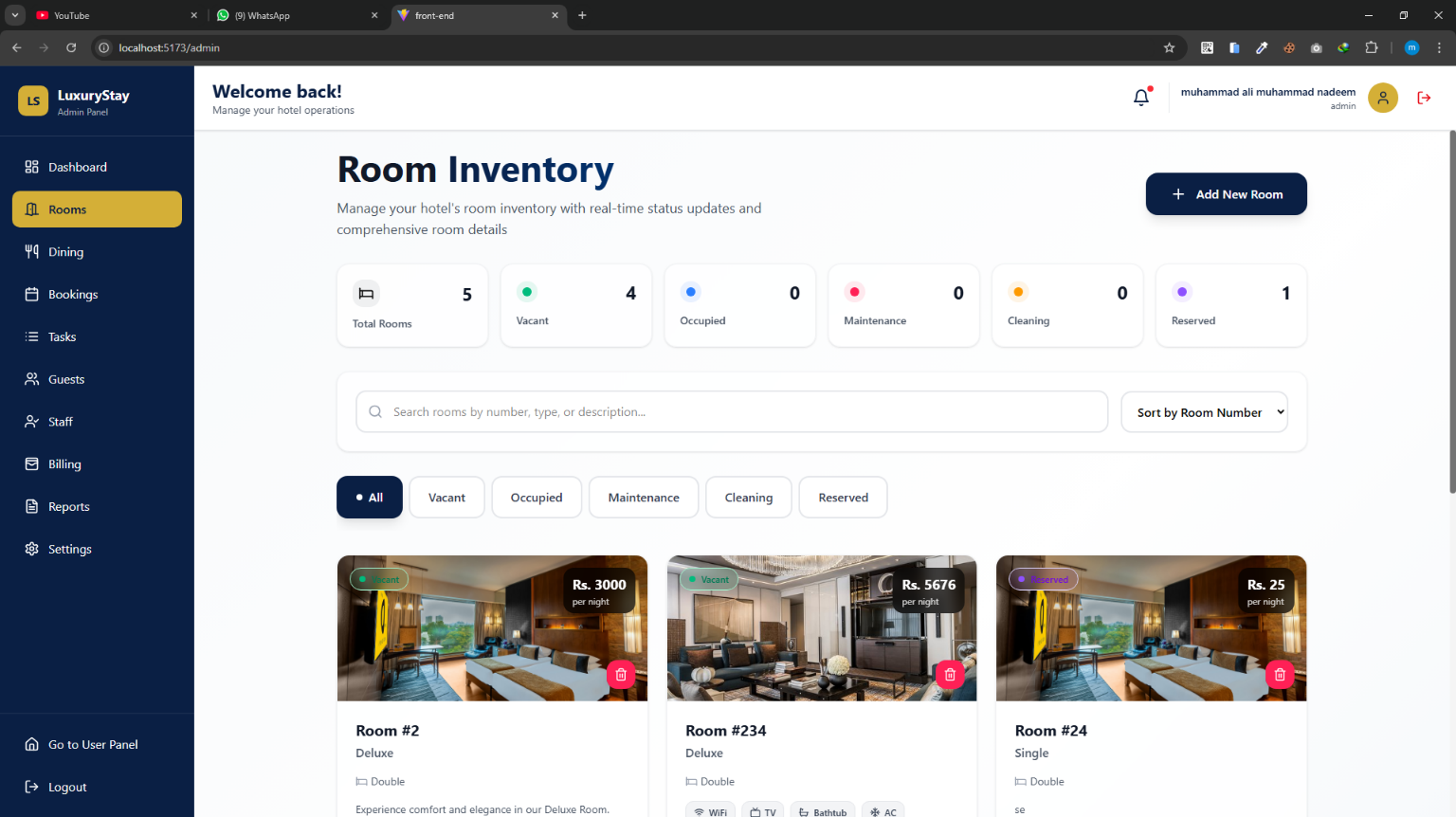
* *User Guide means User Manual, it should have details how a user can operate the application*

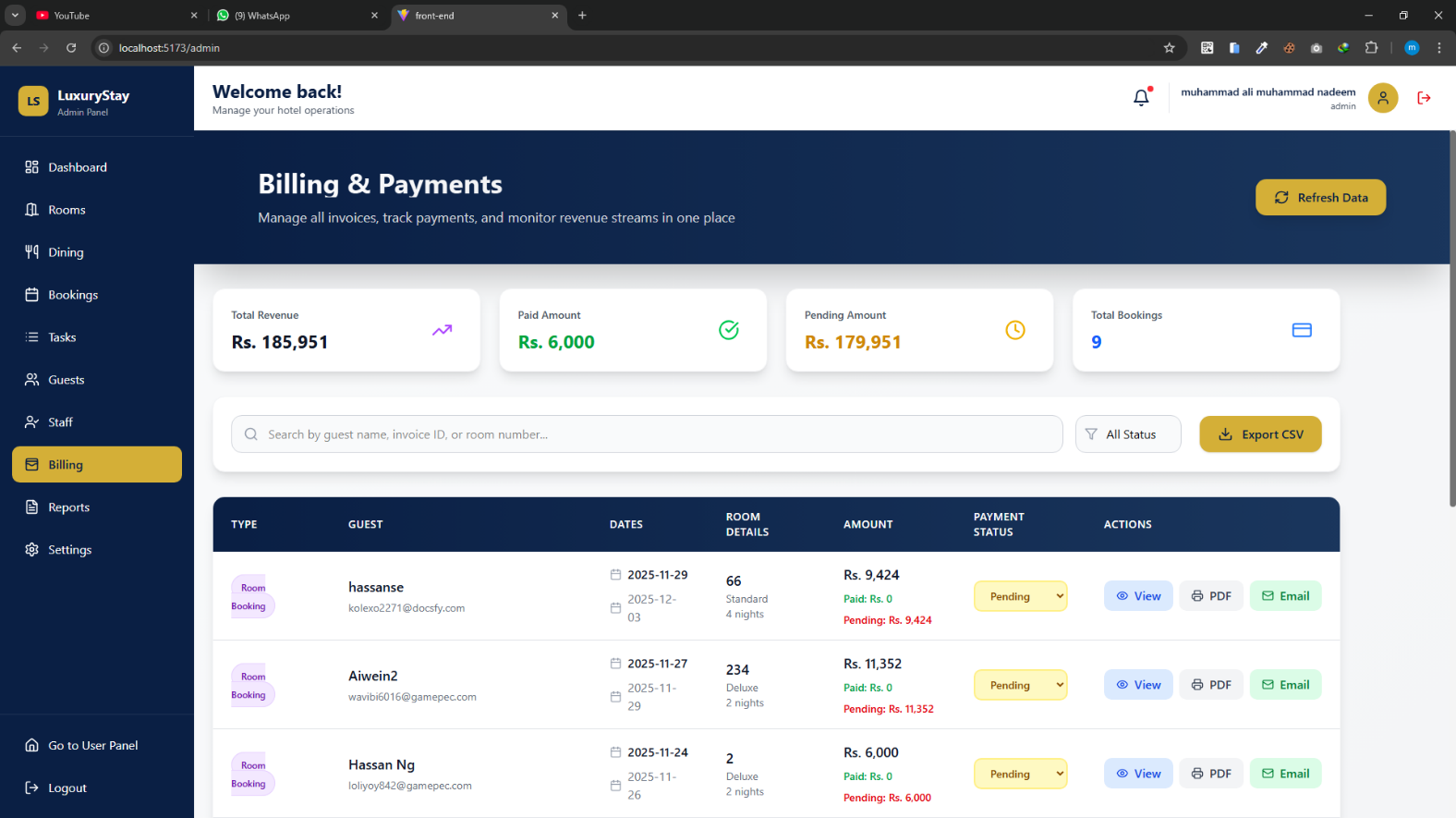
# DEVELOPER’S GUIDE

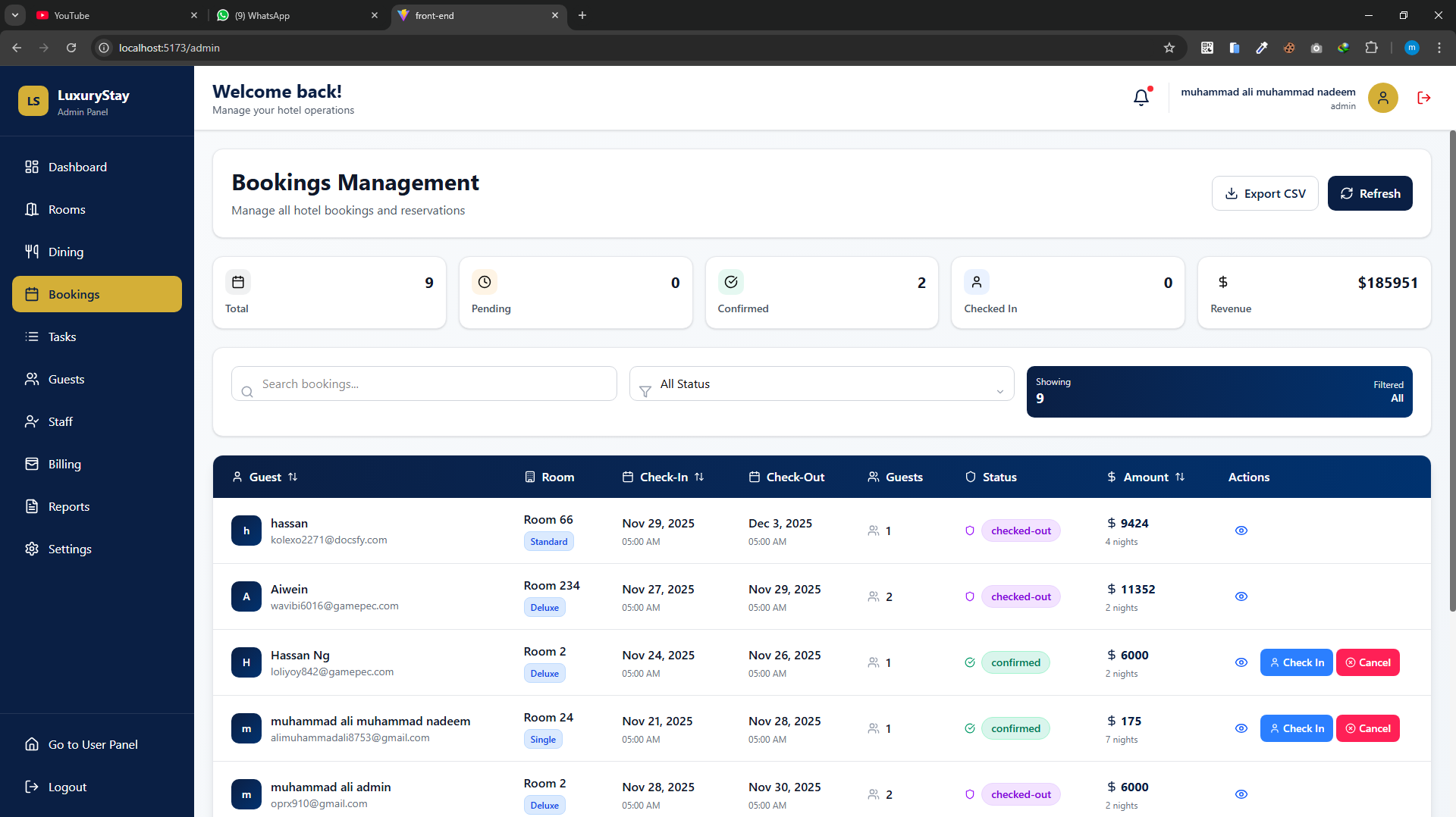
When user starts the website, the below page will show on the screen

**ADMIN PANEL:**



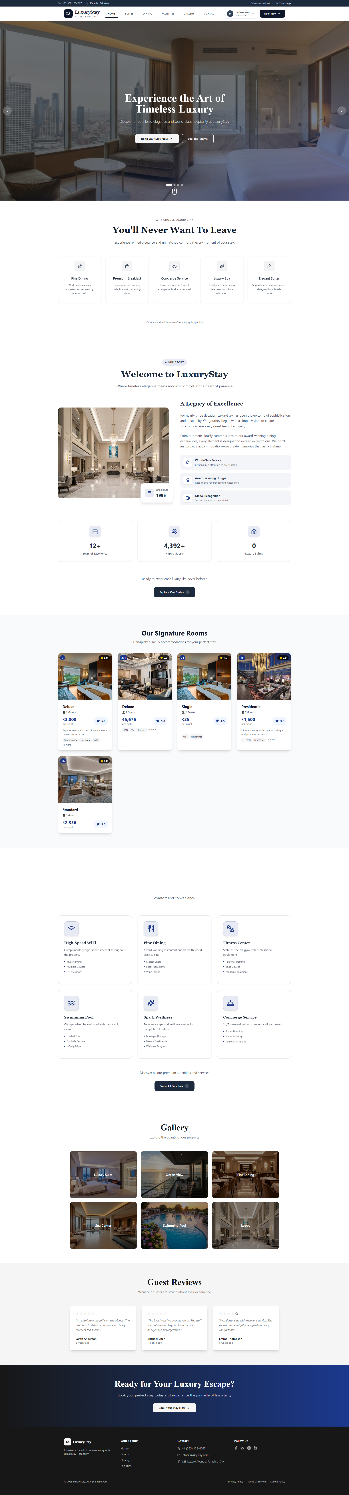


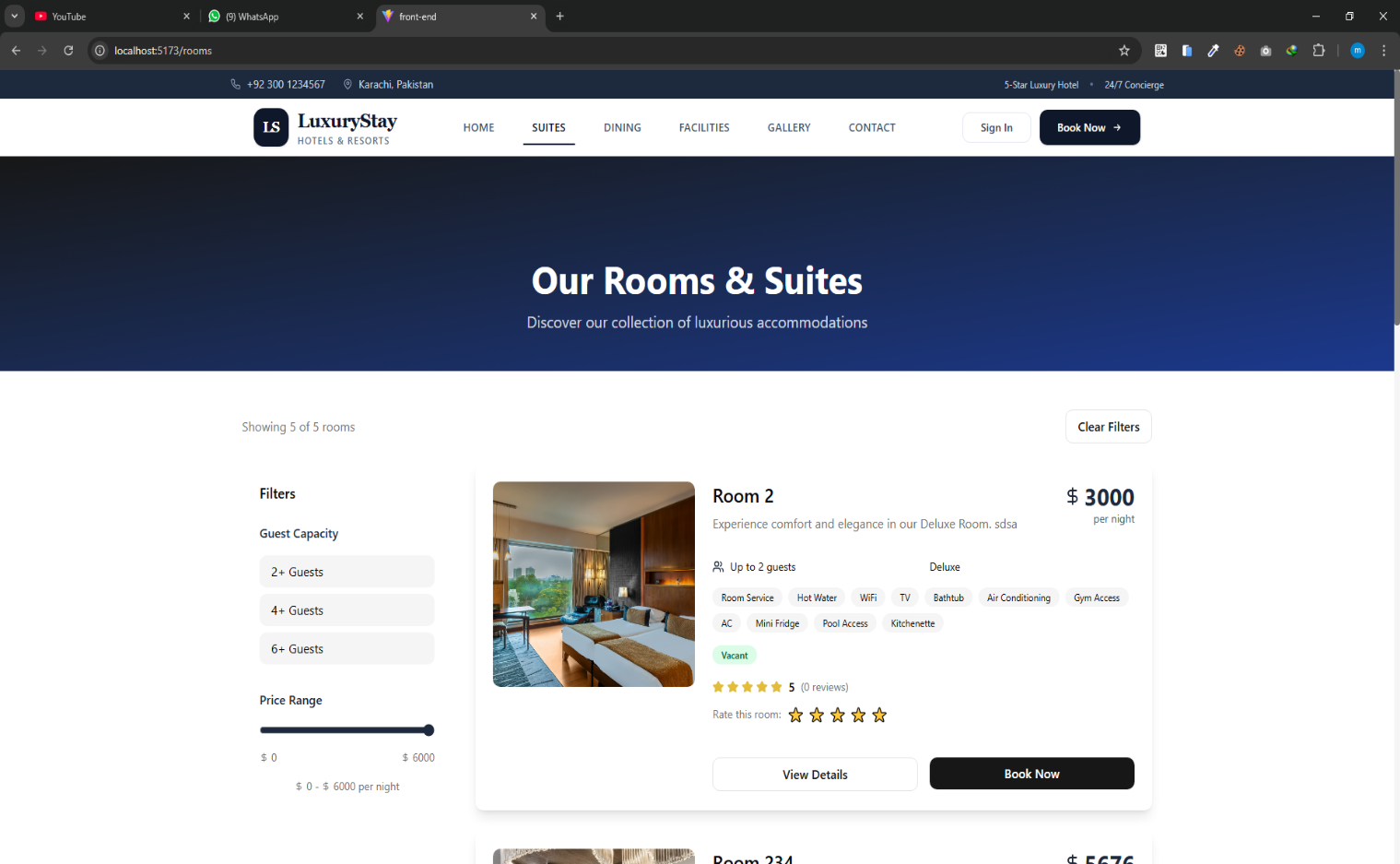


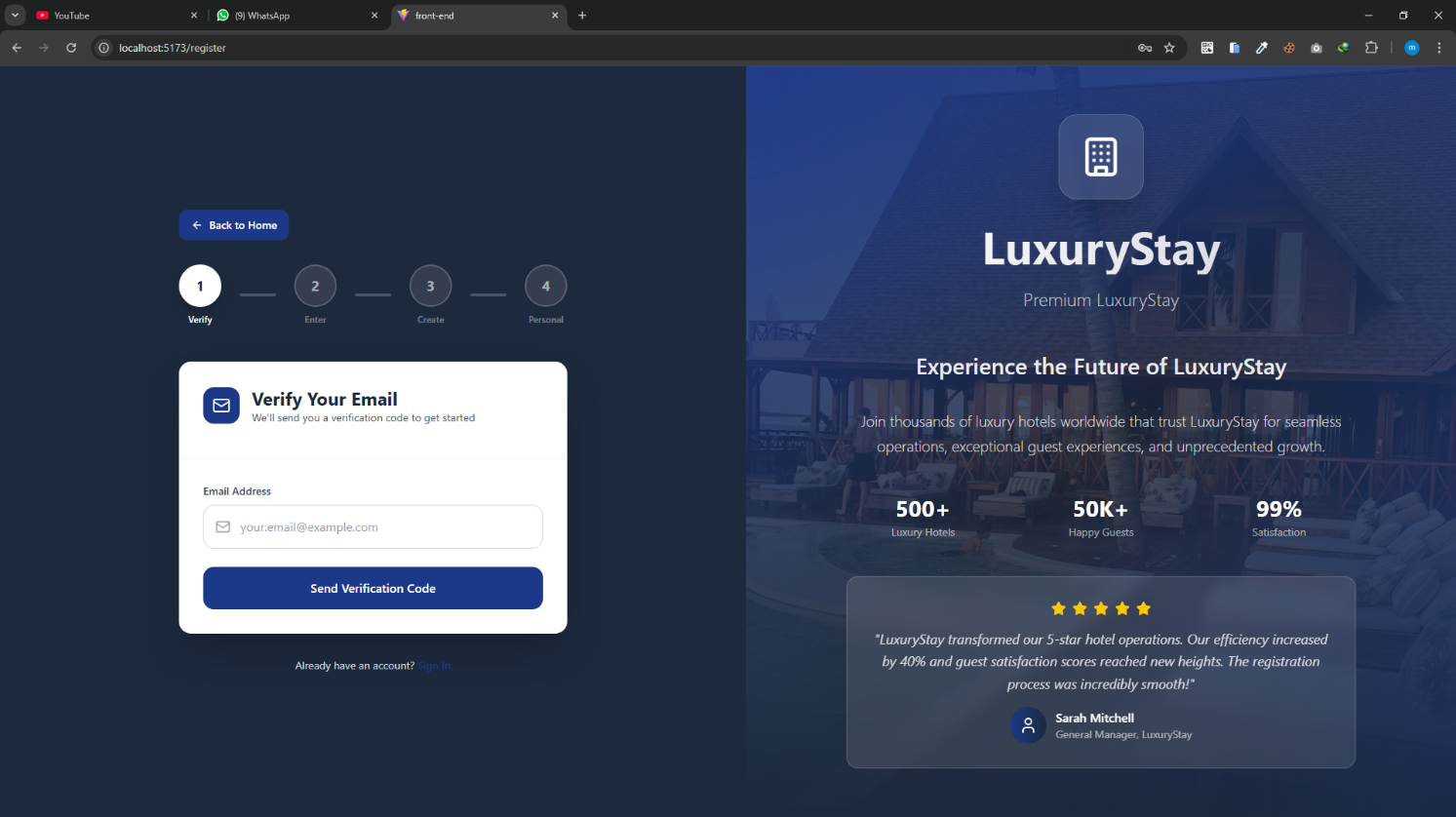


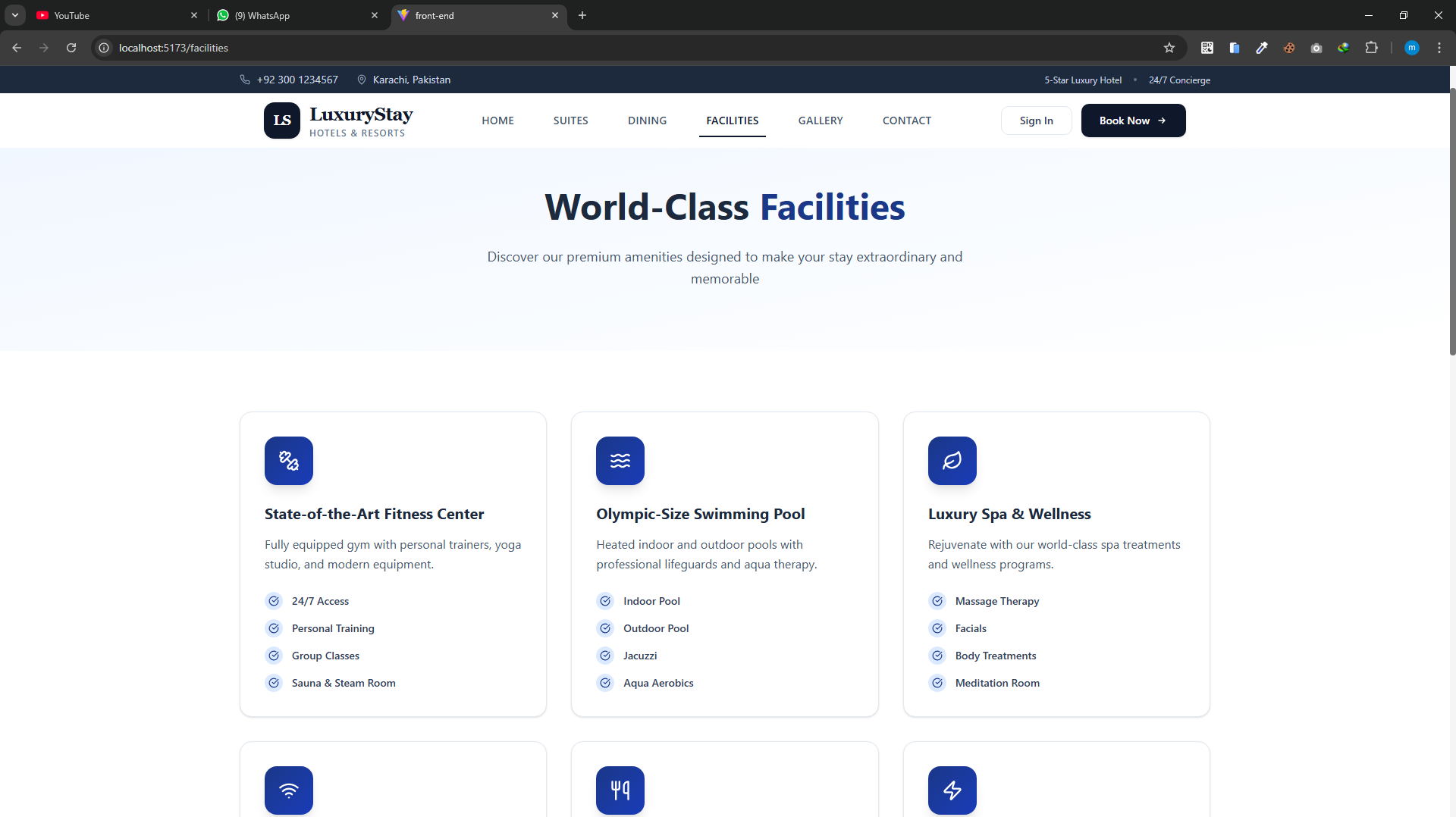
**USER PANEL;**





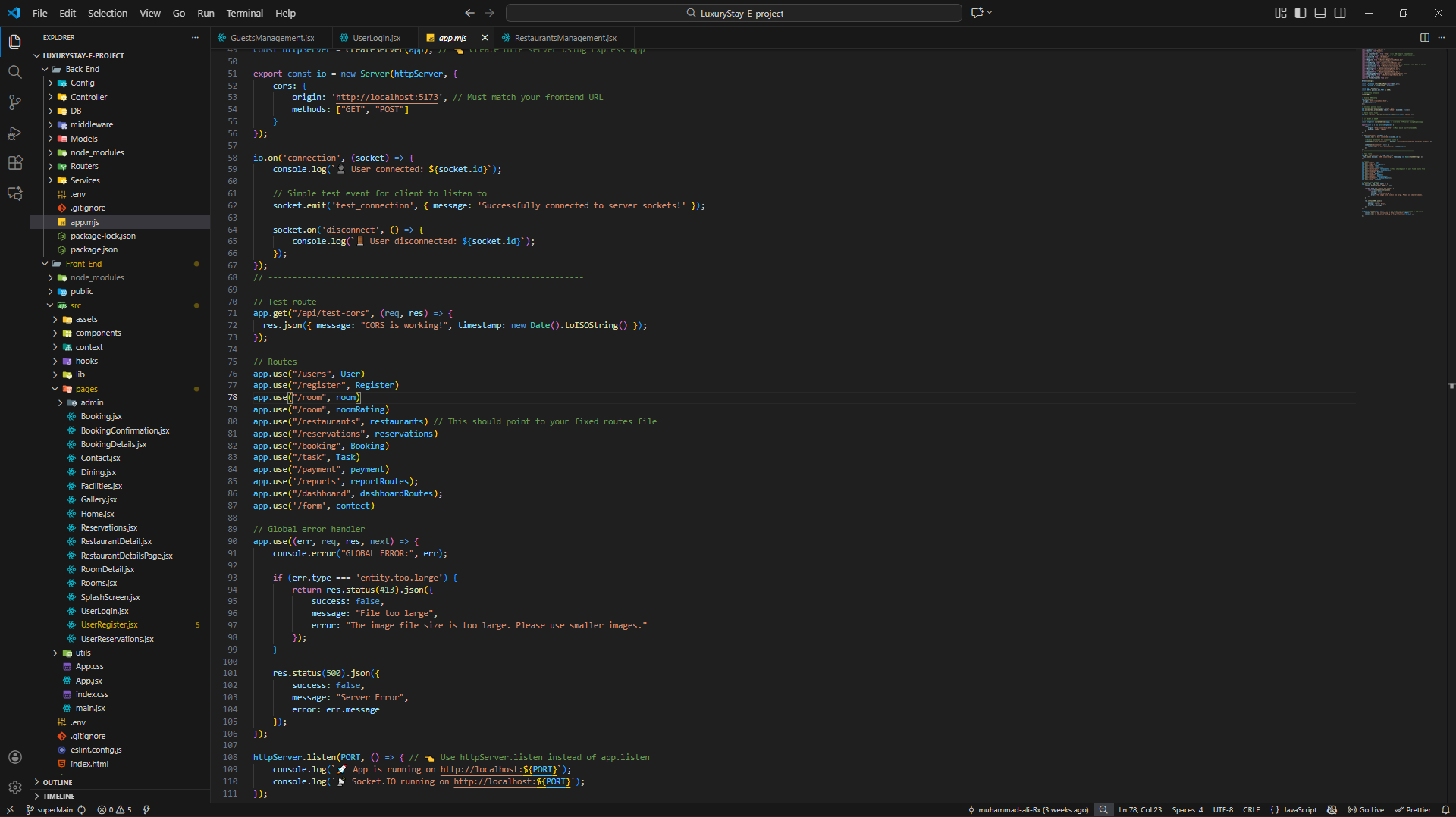


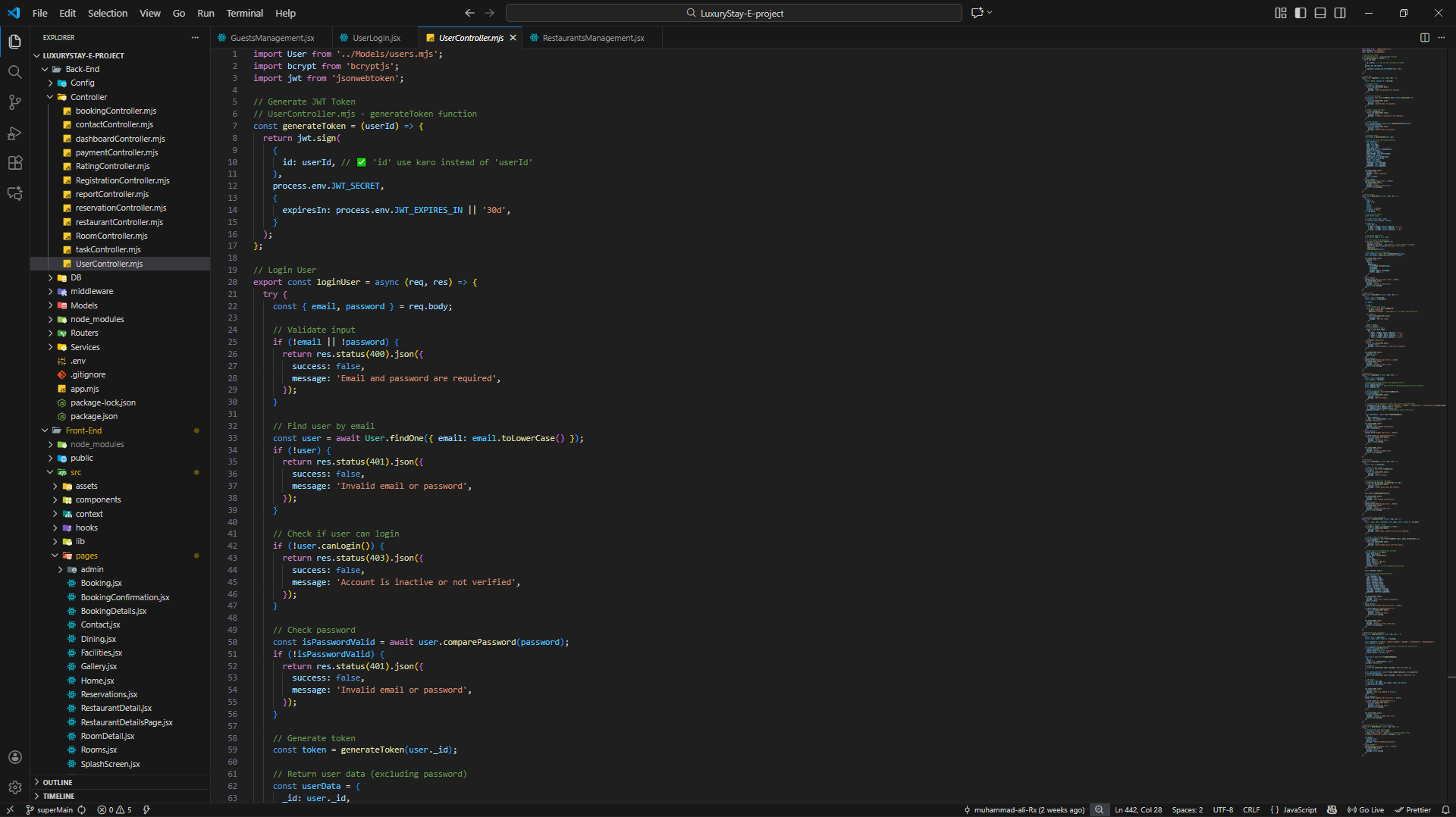


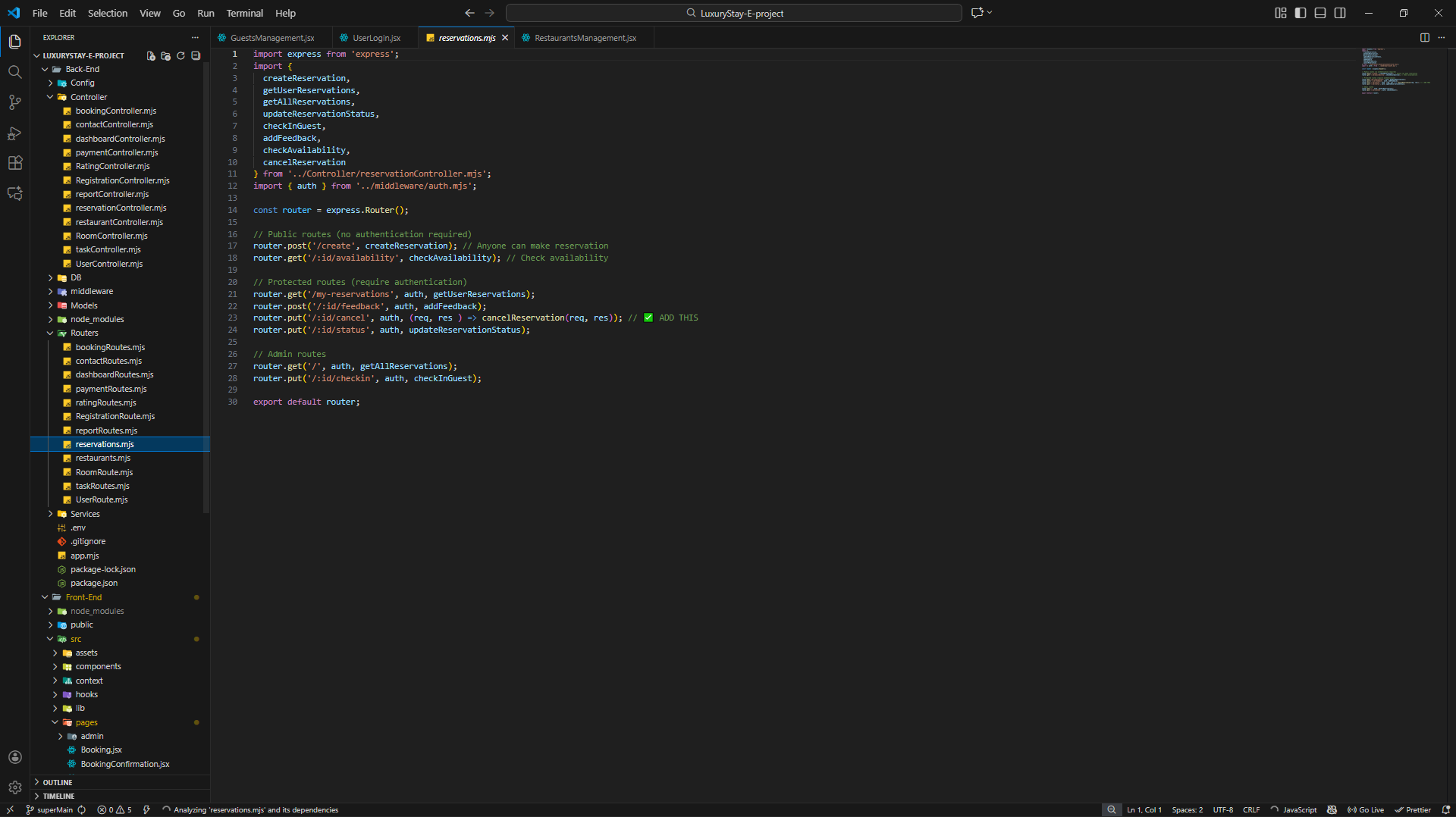


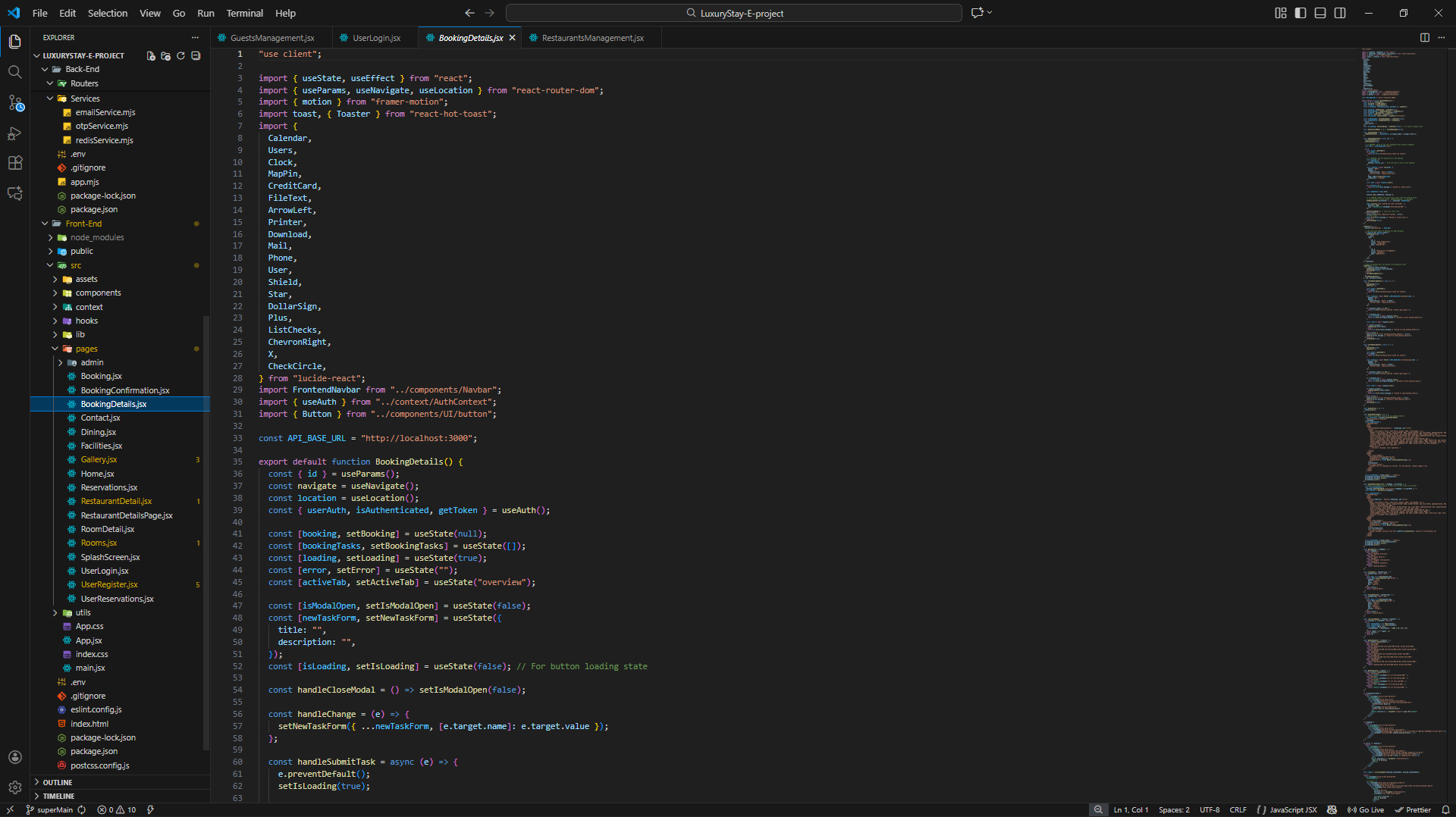


Source Code

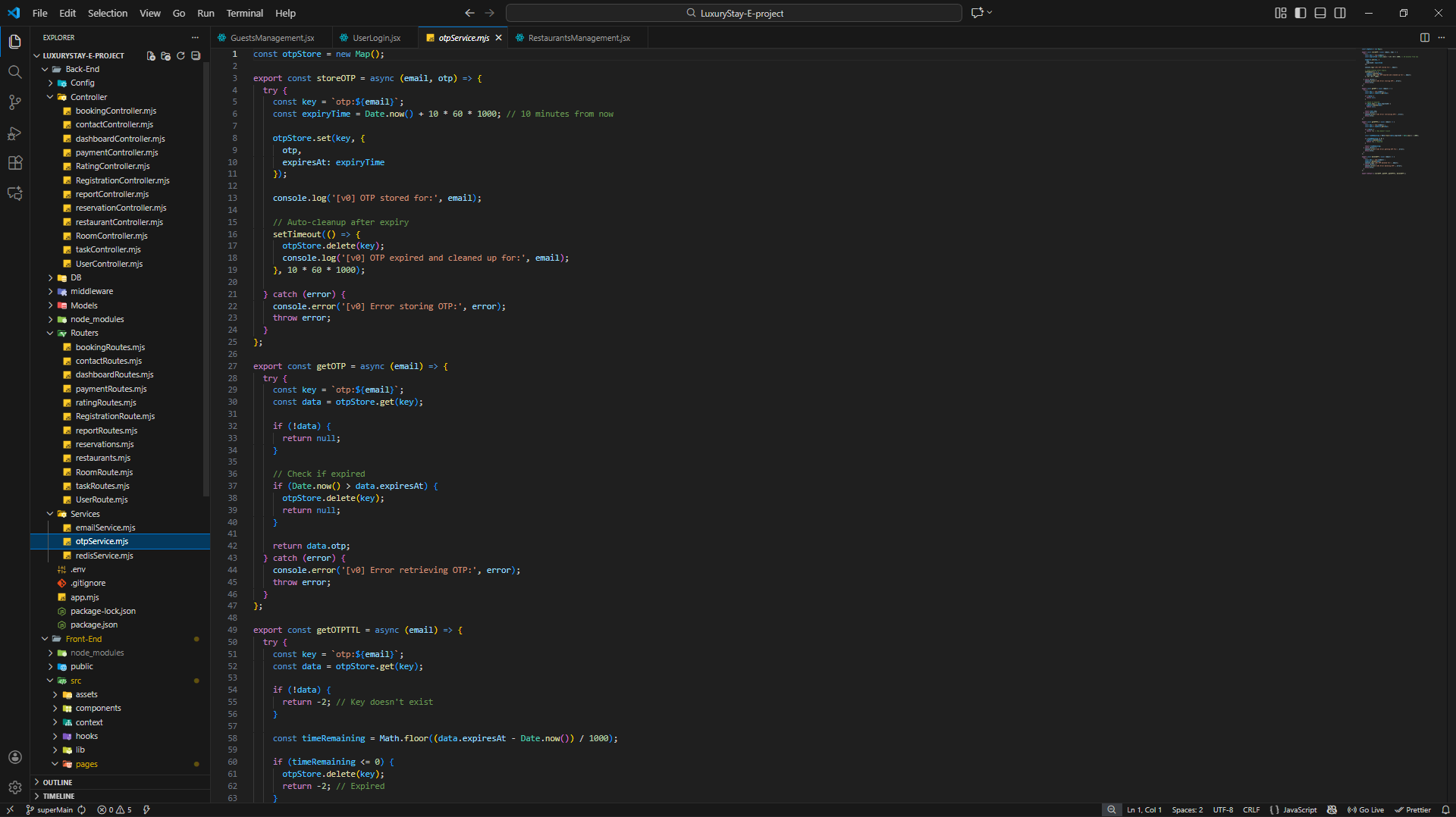


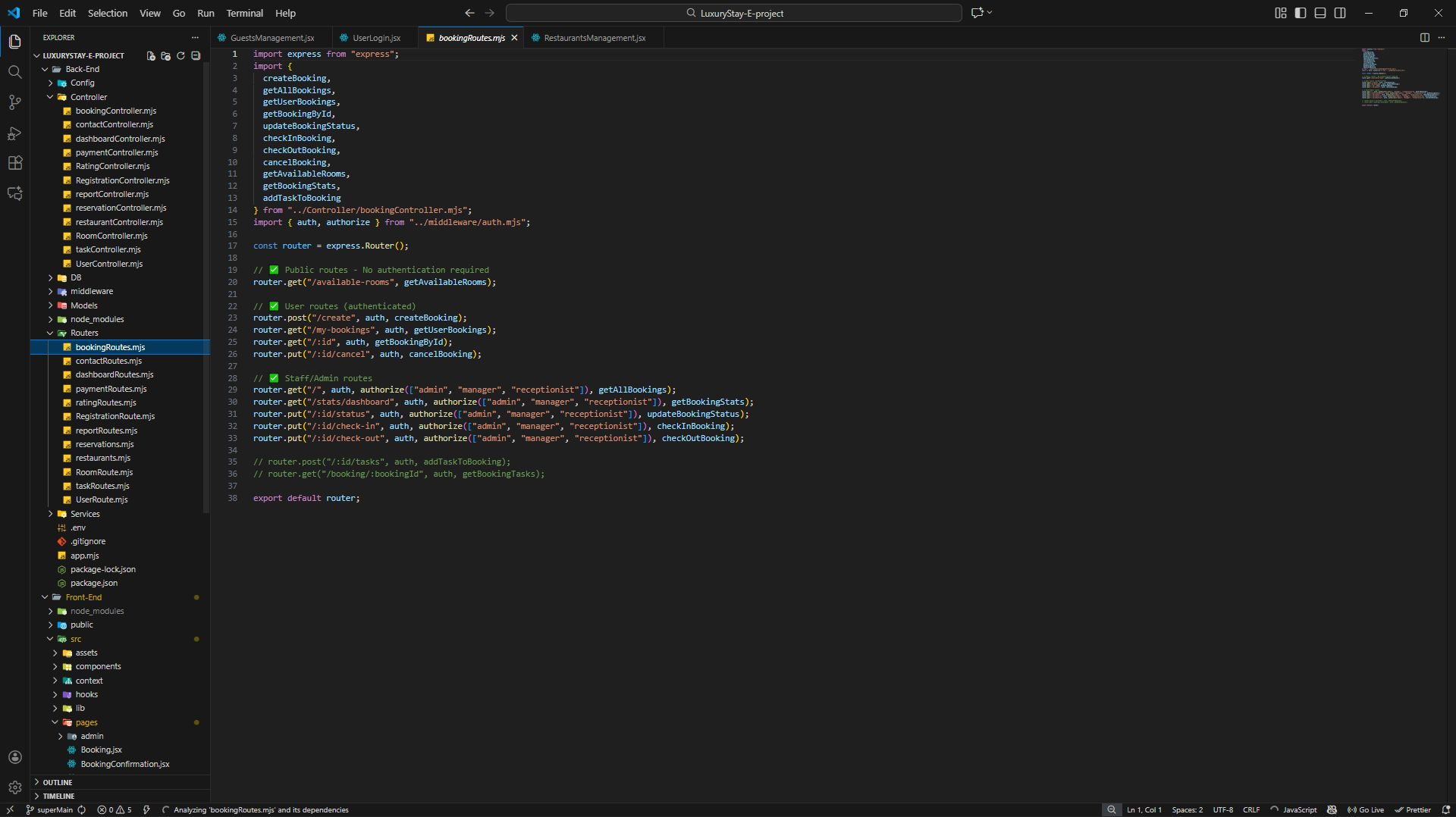












The End