National University of Computer & Emerging Sciences

Hotel Management System

Project Report

Group Code= A

Group Members:

21k-3881 Muhammad Taha Jawaid

21k-3842 Naresh Kumar

21k-3825 Sohaib Baig

21k-3840 Israr Ali

21k-3841 Ahmad Noor Khan

21k-3871 Mohsan

1. Project Feasibility

Technical Feasibility

- The system will be built using a web-based architecture (HTML, CSS, JavaScript, MySQL, PHP)
- Features include secure login/logout, real-time booking, guest management, billing, laundry, and food service tracking.

Operational Feasibility

- Hotel staff will manage reservations, bills, laundry, and food orders easily from a centralized dashboard.
- Guests can view and manage their bookings through a simplified UI.

Economic Feasibility

- Low-cost development using open-source technologies.
- Reduces the need for manual labor, saving operational costs.
- Increases revenue through better room allocation, digital billing.

Project Vision

To develop a smart, user-friendly Hotel Management System that enables seamless room booking, task tracking, and comprehensive billing operations to enhance guest experiences and hotel staff productivity.

Business Case

Market Demand

- Increasing trend of hotels automating operations post-COVID.
- Many budget and mid-size hotels lack an integrated platform for managing reservations, laundry, food, and billing.

Revenue Streams

- One-time software purchase for hotels.
- Monthly/Annual subscription with premium support.

2. Project Initiation

Agile Charter

1. Executive Summary

The Hotel Management System (HMS) is a digital platform developed to automate and streamline hotel operations including room bookings, guest information management, receptionist activity tracking, and billing. Developed using Agile methodology, this project is aimed at delivering incremental value through iterative development, feedback collection, and continuous improvement. The system provides user authentication, dashboards for operational visibility, and efficient resource management while maintaining data accuracy and billing transparency.

2. Project Purpose and Justification

This project addresses the need for a reliable and efficient system to manage hotel operations. Manual processes often result in operational delays and errors. Automating core activities such as reservations, guest record management, and billing can significantly improve customer satisfaction, ensure transparency, and enhance productivity of hotel staff. The HMS provides secure, dynamic dashboards, and integrates extra hotel services into guest records, allowing seamless operations.

3. Business Objectives

1. To build a secure login/logout and authentication system.
2. To implement guest and reservation management features with CRUD operations.
4. To generate automated billing for transparency.
5. To integrate service management for laundry, food, and room cleaning.

4. Project Description

The HMS application allows hotel receptionists and administrators to manage daily hotel tasks through a centralized web-based platform. The functionalities include guest check-in/check-out, room assignment, reservations, billing automation, and performance tracking. Receptionists receive commissions based on completed reservations and associated service usage. Hotel services can be tracked per reservation. The system ensures that billing is automated and not editable by any user.

5. Requirements

- Secure login/logout functionality.
- CRUD operations for guest records and reservation management.
- Add extra services to reservations (e.g. laundry, food, room cleaning).
- Automated, secure billing system.
- Dashboard showing reservations, room types, and performance statistics.

6. Constraints

- Limited time frame due to academic deadlines.
- Restricted to free development tools and platforms.
- No access to real hotel management APIs or datasets.

7. Assumptions

- Team members will contribute equally.
- Scope and requirements will remain stable once development begins.
- All modules will be developed using open-source or free tools.

8. Preliminary Scope Statement

In Scope:

- Authentication module
- Guest and room management
- Reservation and service tracking
- Dashboard
- Billing management system

9. Risks

Risk	Likelihood	Impact	Mitigation Strategy
Time management issues	High	High	Use sprints and regular stand-ups
Team member unavailability	Medium	Medium	Reassign tasks or have backup contributors

Misunderstood requirements	Medium	High	Continuous feedback and prototyping

10. Budget Summary

Item	Estimated Cost (PKR)
Internet and Power Sharing	2,000
Software Tools & Design	1,000
Miscellaneous	2,000
Total Estimated Budget	5,000

11. Authorization

This Agile Project Charter is acknowledged and approved by the team members and the assigned supervisor. Final sign-off will be taken at the time of project completion and demonstration.

3. Plan Release 1

Breakdown of Epics

Epic 1: User Authentication & Authorization

- Login and logout functionality with role-based access.

Epic 2: Room Reservation Management

- Add, update, and manage reservations.

Epic 3: Guest Management

- Add and update guest profiles.

Epic 4: Dashboard & Home Screen

- Show user greeting, reservation stats, billing count.

Epic 5: Room Type & Availability

- Define and display room types and availability.

Epic 6: Billing Automation

- Generate immutable bills upon reservation completion.

Epic 7: Service Add-ons Management

- Enable receptionist to add services.

Estimate Stories with Poker Planning

(Team Size: 6 Members)

Story	Story Points
User login/logout implementation	3
Create reservation (basic info + room selection)	5
Update reservation details	3
Add guest profile	3
Update guest profile	2
Dashboard metrics (reservations & bills)	5
Calculate and display available rooms	5
Auto-generate bills upon reservation	8
Add services to bill	5
Track receptionist commissions	5

Create Release Plan

Release 1: Core Functionalities

Login/logout, reservations, guests, billing, room types, dashboard

Release 2: Performance

performance tracking, services, security

4. Iteration 0 (Pre-development Phase)

Architectural Spikes

Frontend and Backend created using the following languages

- PHP
- HTML
- CSS
- Database in MySQL
- JavaScript

Prepare for Iteration 1

Implement basic authentication module

Define user stories & acceptance criteria

Setup project management board

5. Iteration 1 - 4 (Development Sprints)

Sprint 1: Authentication & Core Setup

Stories:

Login/logout

Role-based access

Initial dashboard (user greeting, reservation count)

Activities:

Code & unit tests

Setup navigation bar and user context

Backend login API + session management

Testing:

Auth flow test

Session persistence test

Acceptance testing

Sprint 2: Reservation & Guest Management

Stories:

Add/update reservation

Add/update guest

Track reservation per receptionist

Activities:

Backend model integration

CRUD UI for reservation/guest

Testing update flows

Testing:

UI form validation

Backend integration tests

Sprint 3: Billing & Room Management

Stories:

Define/display room types

Show available rooms

Generate bills automatically

Activities:

Implement room catalog

Room availability logic

Billing microservice

Sprint 4: Testing and Final Release

Stories:

Add optional services to the bill

Count processed bills per receptionist

Activities:

UI service add-on section

Backend for commissions

Final styling + responsiveness

Conduct Code Reviews

- **Frequency:** Daily or every time a Pull Request (PR) is raised.
- **Checklist:**
    - Code readability and naming conventions.
    - Proper validations and error handling.
    - No hard-coded credentials or values.

Execute Functional & Integration Testing

- **Functional Testing:**
    - Ensure each module performs its intended function (e.g., bill is generated only after reservation).
- **Integration Testing:**
    - Validate interactions between modules (e.g., reservation links correctly with guest and billing).
    - Test service add-ons workflow (e.g., add laundry service → bill updates automatically).

Testing & Validation

Frontend Testing (HTML/CSS/JS)

Manual UI Testing

- **Objective:** Ensure visual elements work as expected and user experience is smooth.
- **Activities:**
    - Check all buttons, forms, and links.
    - Confirm CSS styles load correctly in all supported browsers.

Cross-Browser Testing

- **Browsers Tested:** Chrome

Functional Frontend Testing Scenarios

- Login & logout buttons redirect appropriately.

- Forms for reservation/guest info submit and clear correctly.
- Feedback for incorrect inputs (e.g., empty fields, invalid formats).
- Room cards display the correct type, price, and image.

Backend Testing (PHP)

Unit Testing in PHP

- **Tool: PHP Unit**
- **Test Examples:**
    - Validate reservation form input before insertion.
    - Ensure bill generation functions return correct totals.
    - Confirm login checks session and credentials properly.

Database Validation

- Manually check your MySQL tables for:
    - Correct insertions (guests, reservations, services).
    - Proper relational integrity (e.g., bill → reservation → guest).
- Use MySQL .

Acceptance Testing for Each Sprint

- **Each module will be tested based on real-world user flows:**
    - Receptionist logs in and creates a reservation.
    - Receptionist updates service used by the guest, bill updates automatically.
    - System displays count of bills and reservations for the receptionist only.
- **Client UAT Checklist includes:**
    - Screens load properly without errors.
    - Workflows are logical and error-free.

Agile Meetings

1. Daily Standups

- **Duration:** 15 minutes max
- **Format:**
    - What was done yesterday?
    - What will be done today?
    - Any blockers?
- **Goal:** Maintain momentum and quickly address obstacles

Iteration Retrospective

- **Timing:** End of every sprint (every 3 weeks)
- **Format:**
    - What went well?
    - What didn’t go well?

  - What can we improve for next sprint?
- **Outcome:** Action items to be implemented in next sprint

6. Project Close-Out

Final Activities

Conduct Final User Acceptance Testing (UAT):

1. UAT Objective

Ensure the hotel reservation system meets the business requirements and is ready for production by validating all major user-facing functionalities.

2. Key Features to Test

Based on your features, the following will be tested:

- Login/Logout
- Booking
- Reservation
- Add Guest
- Dashboard Homepage
- Auto Generate Billing
- Type of Available Rooms

3. Test Schedule

- Day 1–2: Functional testing (login, booking, dashboard)
- Day 3–4: Scenario-based testing (guest additions, room availability)
- Day 5: Billing and end-to-end flow

4. UAT Entry Criteria

- The system is fully developed and internally tested
- Test data is available (rooms, users, guests)

5. UAT Exit Criteria

- Minor issues documented with workarounds
- Sign-off from business users

Address final bug fixes and enhancements:

Bug fixes:

Feature	Issue	Fix Implemented
Login/Logout	Login screen doesn't show error on invalid credentials	Added proper error messaging with retry logic

	Logout button doesn’t always clear session	Ensured session/token invalidation on logout
Booking	Booking button clickable multiple times	Added debounce/throttle to prevent multiple submissions
Reservation	Incorrect time zone on reservation timestamps	Standardized time zone handling across backend/frontend
Add Guest	Form allows submission with missing required fields	Added frontend + backend validation
Billing	Incorrect tax calculation if bill is generated manually	Auto bill generation but receptionist can add food service charges manually
Room Types	Room availability not updated after cancellation	Added real-time room status refresh logic

Enhancements:

Reservaion: Add ability to modify reservation dates.
Billing: Detailed bill breakdown per day and service.
Room Types: Filter by room amenities (e.g., AC, sea view, suite)

Obtain client sign-off:

A formal client sign-off document confirms the client has reviewed the system, verified that it meets the agreed-upon requirements, and approves it for deployment or production use.

Post-Launch support plan:

Phase 1: Monitoring system performance:

Objectives

- Identify and resolve performance bottlenecks
- Collect usage data to inform future enhancements

Monitoring Workflow

1. Set up monitoring dashboards
2. Monitor logs and system metrics daily
3. Weekly review of performance reports

Phase 2: Gather user feedback and iterate:

Objectives

- Collect qualitative and quantitative feedback from real users
- Identify usability issues, feature gaps, and improvement areas

- Prioritize and implement enhancements for better user experience and performance

Who to Involve

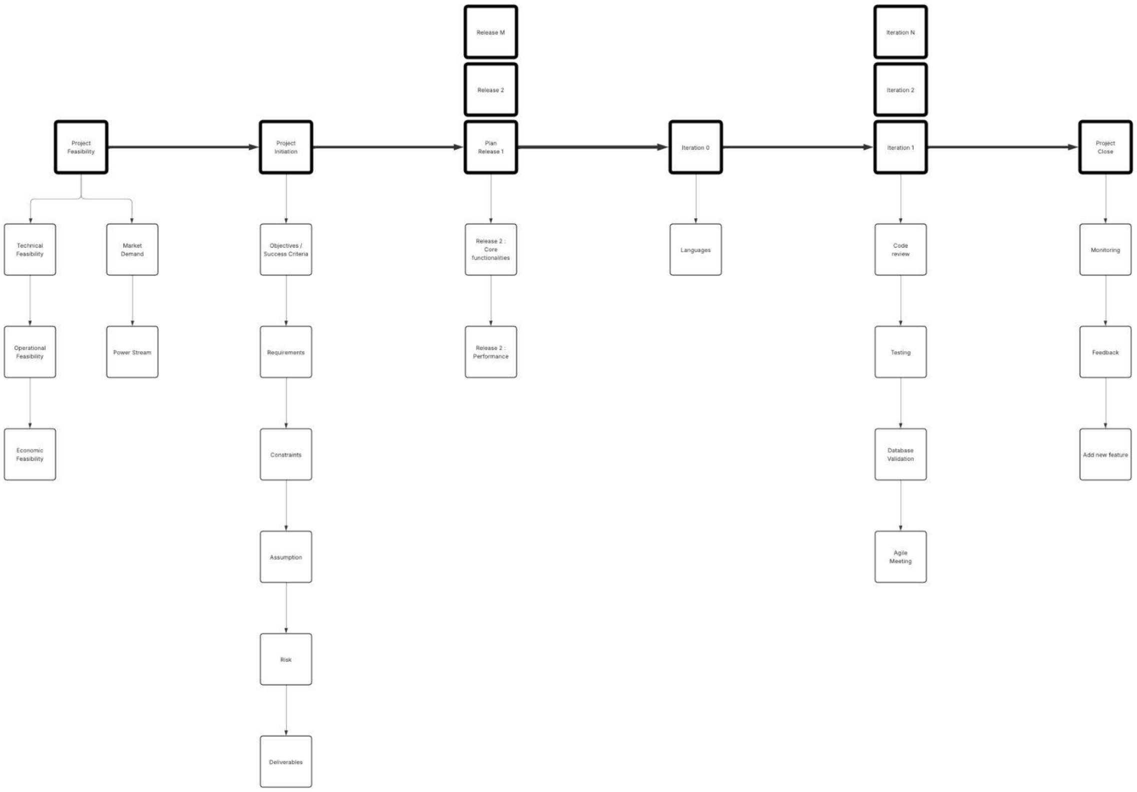
- Front Desk Staff / Receptionists (daily users)
- Customers/Guests

Feedback Iteration Workflow

1. **Collect & Tag Feedback**
   Use tags like usability, performance, bug,
2. **Analyze & Prioritize**
   Apply a simple prioritization model:
   - Urgent + High Impact = Fix Immediately
   - Low Impact + Easy = Quick Win
   - High Effort + Medium Impact = Schedule in future release

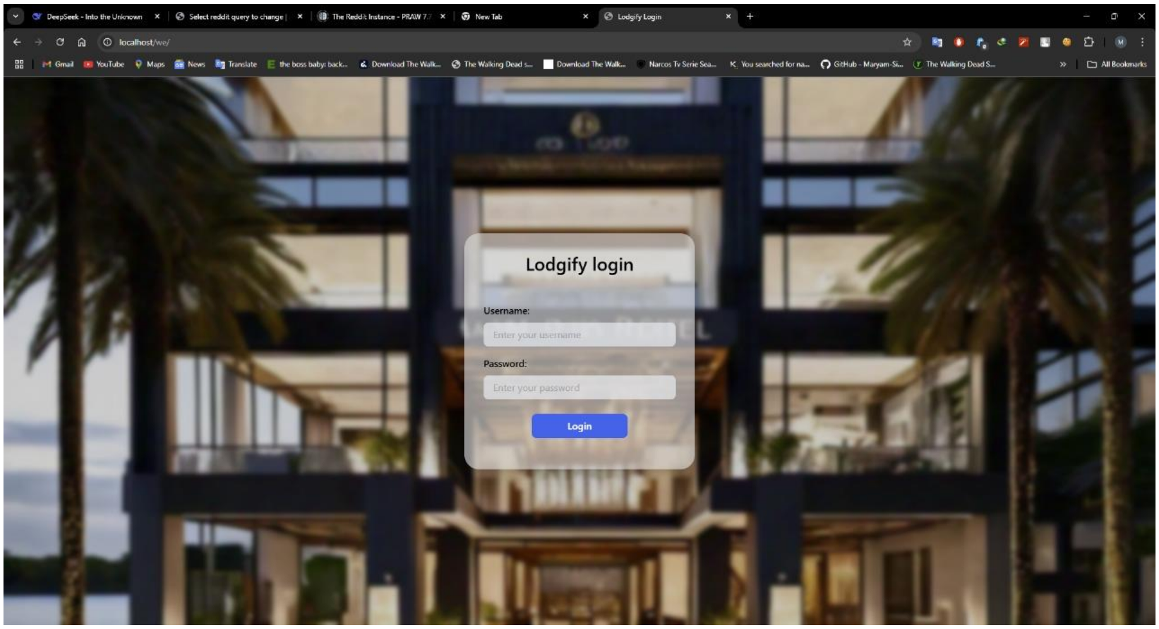
Phase 3: Scale and add new features:

Feature	Why Add It
Multi-property Support	For hotel chains or managing multiple branches
Role-Based Access Control	Define permissions (admin, front desk, housekeeping, accounting)
Online Payment Integration	Allow guests to pre-pay with Stripe, PayPal, or local gateways
Mobile App or PWA	Provide better access for staff and guests on mobile devices
Notifications & Alerts	Email/SMS alerts for bookings, cancellations, reminders
Language & Currency Support	For international guests and multi-region hotels


Release M
Release 2
Iteration N
Iteration 2
Project Feasibility
Project Initiation
Plan Release 1
Iteration 0
Iteration 1
Project Close
Technical Feasibility
Market Demand
Objectives / Success Criteria
Release 2 : Core functionalities
Languages
Code review
Monitoring
Operational Feasibility
Power Stream
Requirements
Release 2 : Performance
Testing
Feedback
Economic Feasibility
Constraints
Database Validation
Add new feature
Assumption
Agile Meeting
Risk
Deliverables

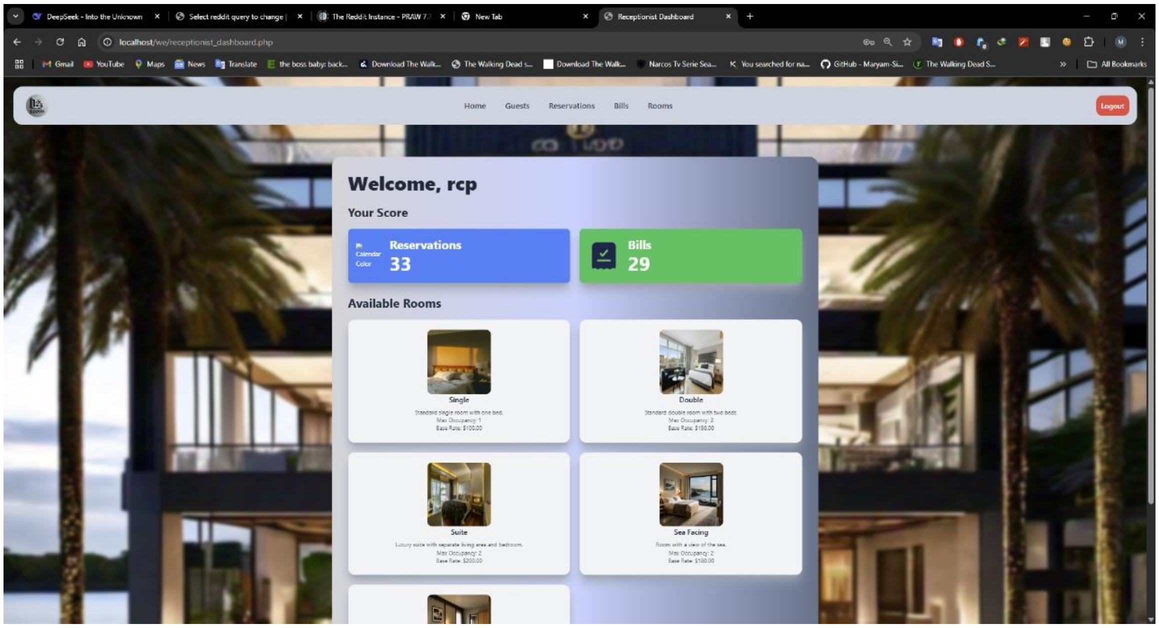
Visuals of Our Project:

Login Screen:



This login screen serves as the entry point to the Hotel Management System. It allows authorized users, such as administrators or staff members, to securely access the system by entering their username and password. Upon submitting valid credentials, users are redirected to the main dashboard where they can perform their respective tasks. If the credentials are incorrect, the system can prevent access and display an error message.

Dashboard Screen:



Welcome, rcp

Your Score

Calendar Color Reservations 33

Bills 29

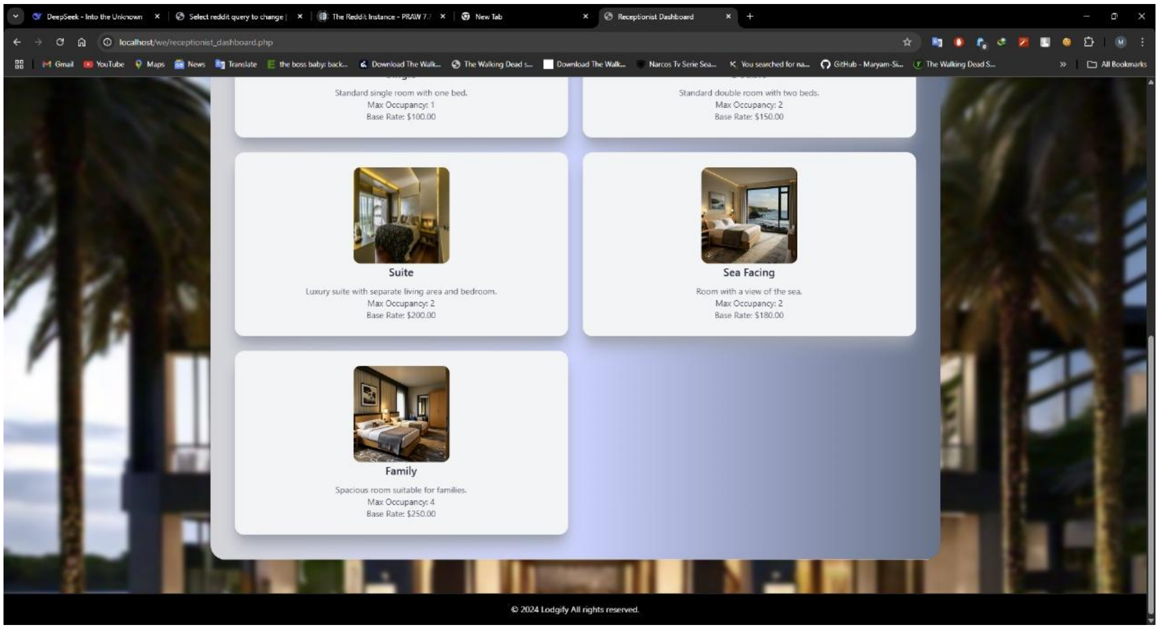
Available Rooms

Single
Standard single room with one bed.
Max Occupancy: 1
Base Rate: $100.00

Double
Standard double room with two beds.
Max Occupancy: 2
Base Rate: $150.00

Suite
Luxury suite with separate living area and bedroom.
Max Occupancy: 2
Base Rate: $200.00

Sea Facing
Room with a view of the sea.
Max Occupancy: 2
Base Rate: $180.00



Standard single room with one bed.
Max Occupancy: 1
Base Rate: $100.00

Standard double room with two beds.
Max Occupancy: 2
Base Rate: $150.00

Suite
Luxury suite with separate living area and bedroom.
Max Occupancy: 2
Base Rate: $200.00

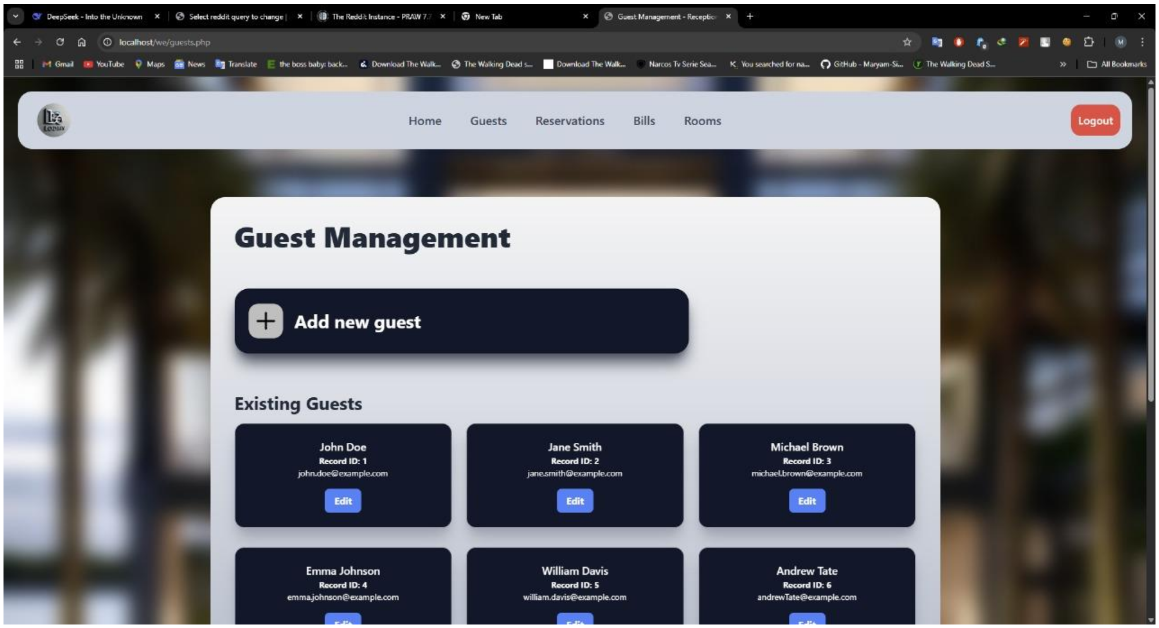
Sea Facing
Room with a view of the sea.
Max Occupancy: 2
Base Rate: $180.00

Family
Spacious room suitable for families.
Max Occupancy: 4
Base Rate: $250.00

© 2024 Lodgify All rights reserved.

This is the main dashboard screen of the Hotel Management System designed for the receptionist. It displays the total number of reservations and bills generated, giving a quick overview of hotel activity. It also shows the list of available room types, such as Single, Double, Suite, Sea Facing, and Family, along with basic information like occupancy and base fare. The receptionist can use this screen to check room availability and guide guests accordingly.

Guest Management Screen:



Guest Management

Existing Guests

John Doe
Record ID: 1
john.doe@example.com
Edit

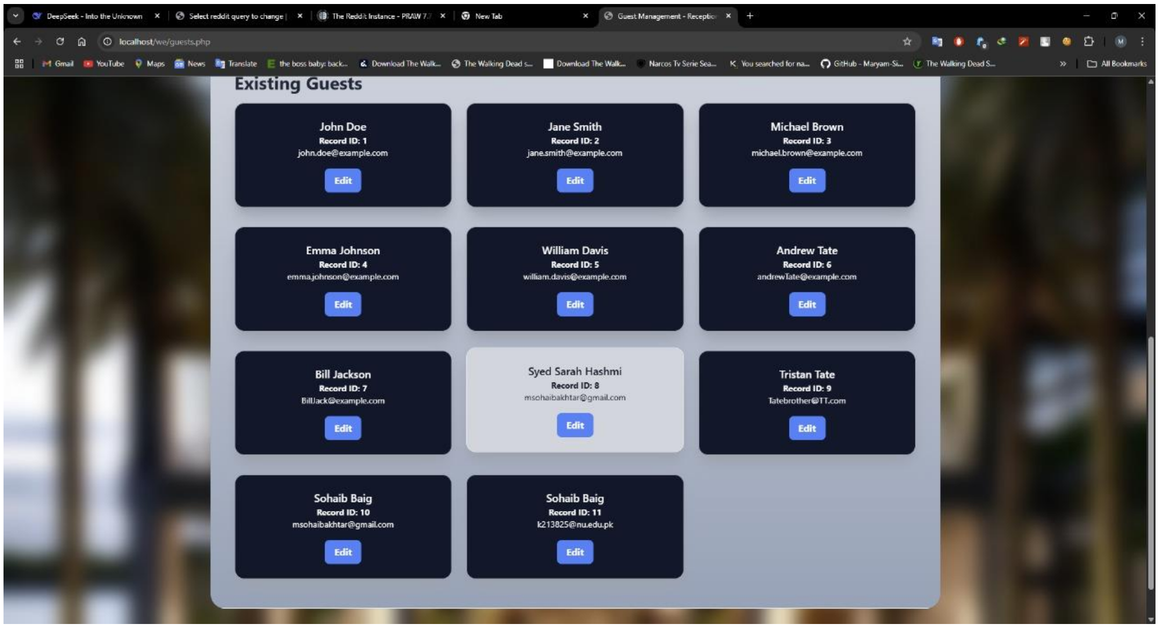
Jane Smith
Record ID: 2
jane.smith@example.com
Edit

Michael Brown
Record ID: 3
michael.brown@example.com
Edit

Emma Johnson
Record ID: 4
emma.johnson@example.com

William Davis
Record ID: 5
william.davis@example.com

Andrew Tate
Record ID: 6
andrewTate@example.com



Existing Guests

John Doe
Record ID: 1
john.doe@example.com
Edit

Jane Smith
Record ID: 2
jane.smith@example.com
Edit

Michael Brown
Record ID: 3
michael.brown@example.com
Edit

Emma Johnson
Record ID: 4
emma.johnson@example.com
Edit

William Davis
Record ID: 5
william.davis@example.com
Edit

Andrew Tate
Record ID: 6
andrewTate@example.com
Edit

Bill Jackson
Record ID: 7
BillJack@example.com
Edit

Syed Sarah Hashmi
Record ID: 8
msohaibakhtar@gmail.com
Edit

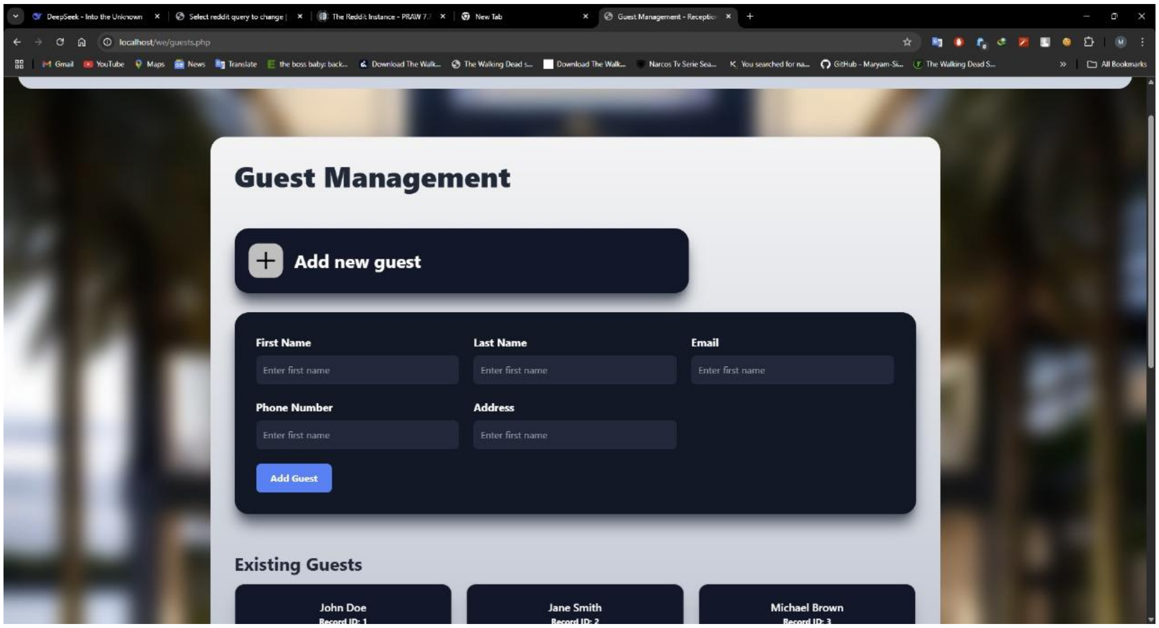
Tristan Tate
Record ID: 9
Tatebrother@TT.com
Edit

Sohaib Baig
Record ID: 10
msohaibakhtar@gmail.com
Edit

Sohaib Baig
Record ID: 11
k213825@nu.edu.pk
Edit

This Guest Management screen allows the receptionist to manage all guest-related records in the Hotel Management System. It provides an option to **add new guests**. **Existing guest records** are displayed below, each with their name, email, and a unique record ID.

Add a New Guest Screen:



Guest Management

First Name	Last Name	Email
Enter first name	Enter first name	Enter first name
Phone Number	Address	
Enter first name	Enter first name	

Existing Guests

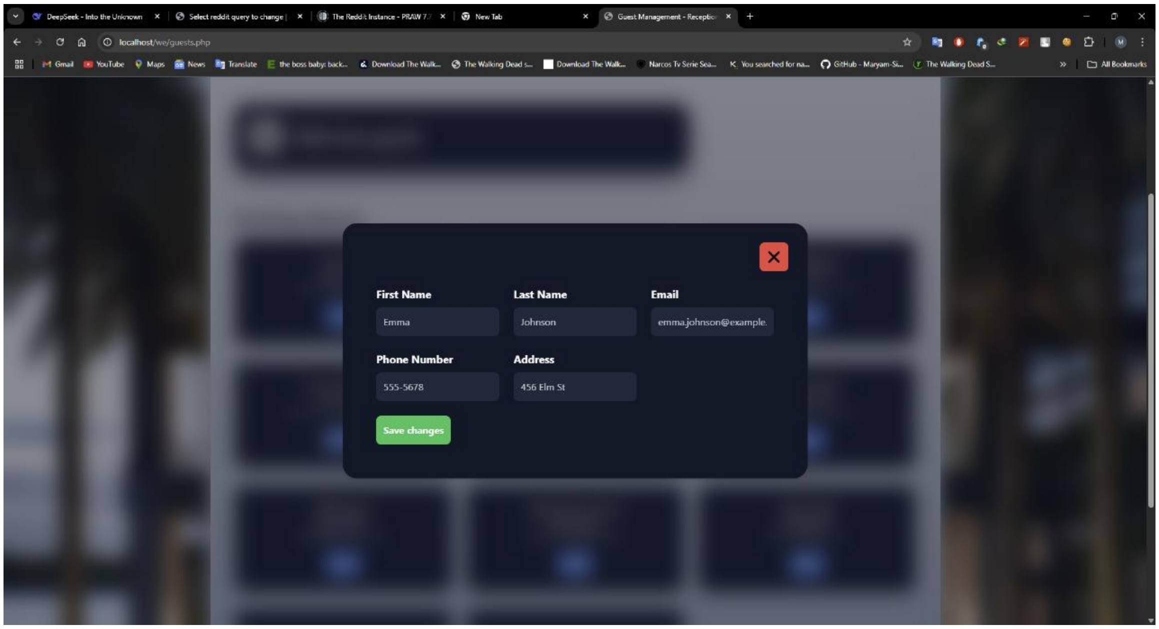
John Doe
Record ID: 1

Jane Smith
Record ID: 2

Michael Brown
Record ID: 3

The **Add New Guest** screen in Hotel Management System allows staff to input and save new guest details such as first name, last name, email, phone number, and address. The form is used to register guests into the system database. After filling in the fields, clicking the "Add Guest" button submits the information. Below the form, a list of existing guests is displayed for quick reference. This helps manage guest records efficiently.

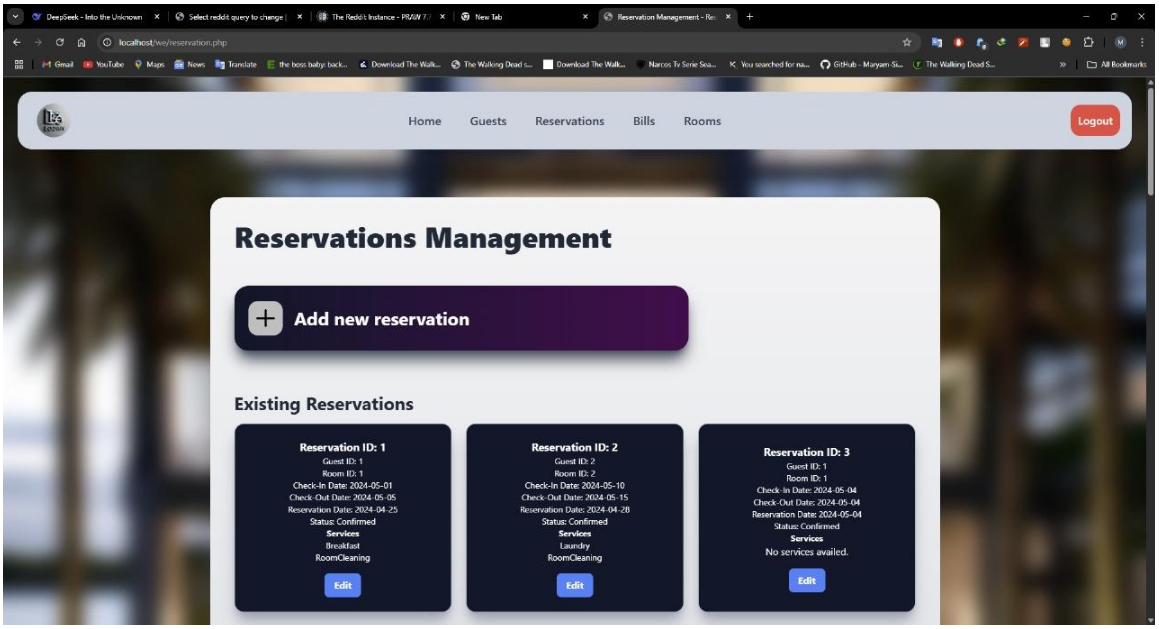
Edit Existing Guest Details:



First Name	Last Name	Email
Emma	Johnson	emma.johnson@example.
Phone Number	Address	
555-5678	456 Elm St	

This screen allows to **edit existing guest details** in the Hotel Management System. It displays editable fields such as First Name, Last Name, Email, Phone Number, and Address, pre-filled with the guest's current information. The user can update any of these fields and click "Save changes" to submit the updates.

Reservation Management Screen:



Reservations Management

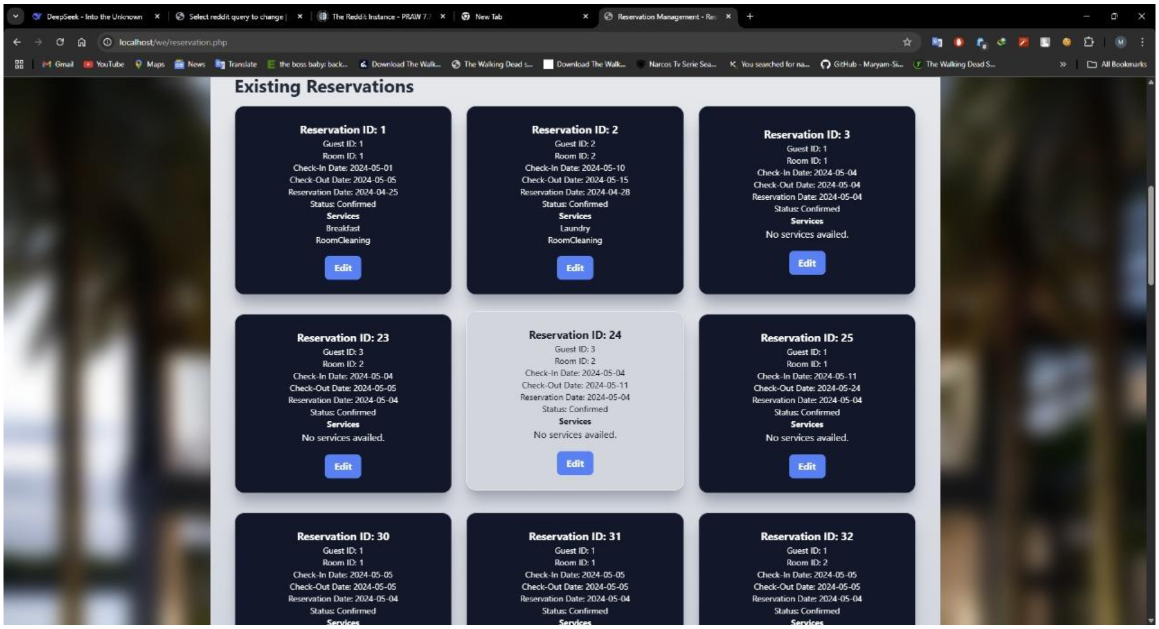
Add new reservation

Existing Reservations

Reservation ID: 1
Guest ID: 1
Room ID: 1
Check-In Date: 2024-05-01
Check-Out Date: 2024-05-05
Reservation Date: 2024-04-25
Status: Confirmed
Services
Breakfast
RoomCleaning
Edit

Reservation ID: 2
Guest ID: 2
Room ID: 2
Check-In Date: 2024-05-10
Check-Out Date: 2024-05-15
Reservation Date: 2024-04-28
Status: Confirmed
Services
Laundry
RoomCleaning
Edit

Reservation ID: 3
Guest ID: 1
Room ID: 1
Check-In Date: 2024-05-04
Check-Out Date: 2024-05-04
Reservation Date: 2024-05-04
Status: Confirmed
Services
No services availed.
Edit



Existing Reservations

Reservation ID: 1
Guest ID: 1
Room ID: 1
Check-In Date: 2024-05-01
Check-Out Date: 2024-05-05
Reservation Date: 2024-04-25
Status: Confirmed
Services
Breakfast
RoomCleaning
Edit

Reservation ID: 2
Guest ID: 2
Room ID: 2
Check-In Date: 2024-05-10
Check-Out Date: 2024-05-15
Reservation Date: 2024-04-28
Status: Confirmed
Services
Laundry
RoomCleaning
Edit

Reservation ID: 3
Guest ID: 1
Room ID: 1
Check-In Date: 2024-05-04
Check-Out Date: 2024-05-04
Reservation Date: 2024-05-04
Status: Confirmed
Services
No services availed.
Edit

Reservation ID: 23
Guest ID: 3
Room ID: 2
Check-In Date: 2024-05-04
Check-Out Date: 2024-05-05
Reservation Date: 2024-05-04
Status: Confirmed
Services
No services availed.
Edit

Reservation ID: 24
Guest ID: 3
Room ID: 2
Check-In Date: 2024-05-04
Check-Out Date: 2024-05-11
Reservation Date: 2024-05-04
Status: Confirmed
Services
No services availed.
Edit

Reservation ID: 25
Guest ID: 1
Room ID: 1
Check-In Date: 2024-05-11
Check-Out Date: 2024-05-24
Reservation Date: 2024-05-04
Status: Confirmed
Services
No services availed.
Edit

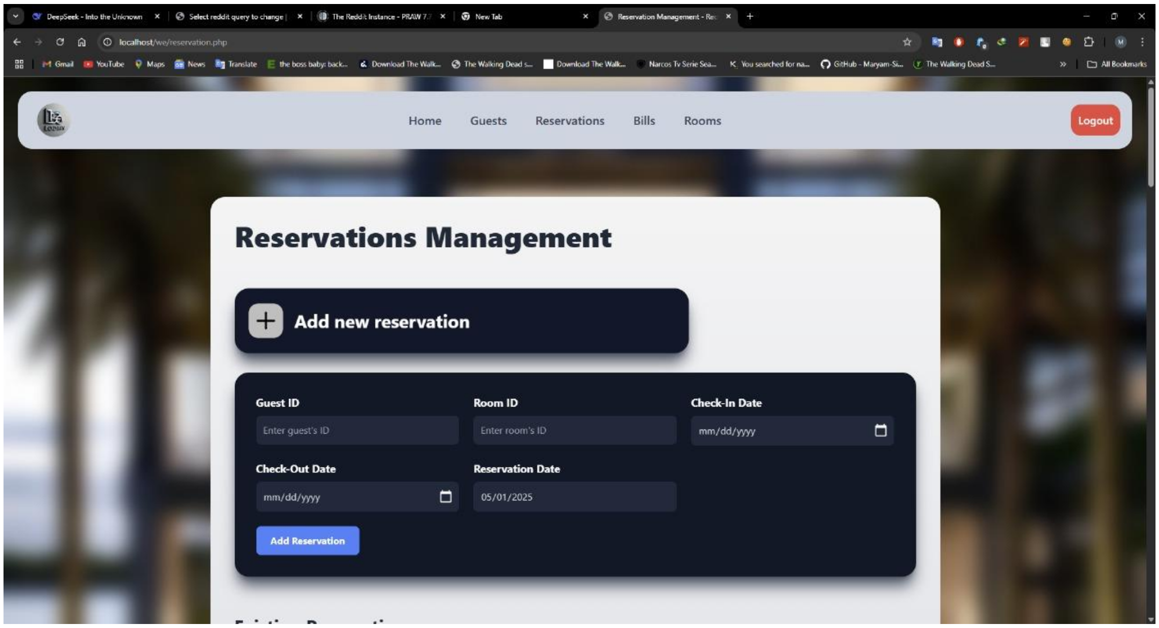
Reservation ID: 30
Guest ID: 1
Room ID: 1
Check-In Date: 2024-05-05
Check-Out Date: 2024-05-05
Reservation Date: 2024-05-04
Status: Confirmed
Services

Reservation ID: 31
Guest ID: 1
Room ID: 1
Check-In Date: 2024-05-05
Check-Out Date: 2024-05-05
Reservation Date: 2024-05-04
Status: Confirmed
Services

Reservation ID: 32
Guest ID: 1
Room ID: 2
Check-In Date: 2024-05-05
Check-Out Date: 2024-05-05
Reservation Date: 2024-05-04
Status: Confirmed
Services

This **Reservations Management** screen includes **Add new reservation** option at the top, allowing staff to input new booking details such as Guest ID, Room ID, and dates. Below it, **existing reservations** are displayed with full details and an "Edit" button for updates, helping manage both new and current bookings in one place.

Add New Reservation Screen:



Reservations Management

Guest ID: Enter guest's ID

Room ID: Enter room's ID

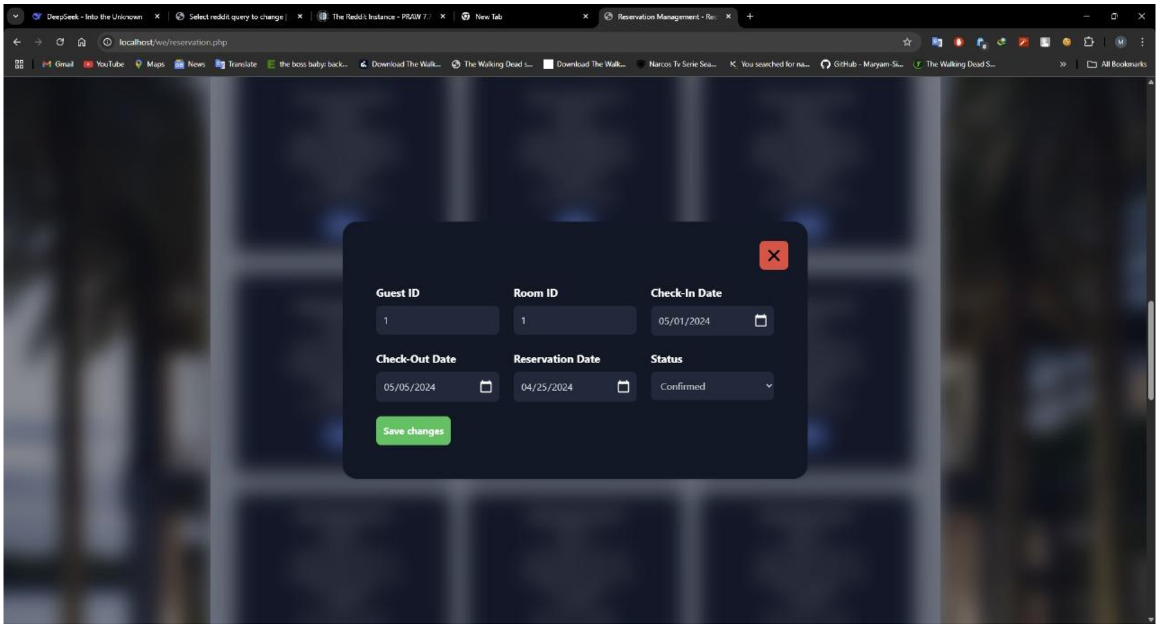
Check-In Date: mm/dd/yyyy

Check-Out Date: mm/dd/yyyy

Reservation Date: 05/01/2025

The **Add new reservation** screen allows hotel staff to create new reservations by entering the Guest ID, Room ID, Check-In and Check-Out dates, and the reservation date. After filling in the details, clicking "Add Reservation" stores the booking in the system. It ensures proper tracking of room bookings and guest stays.

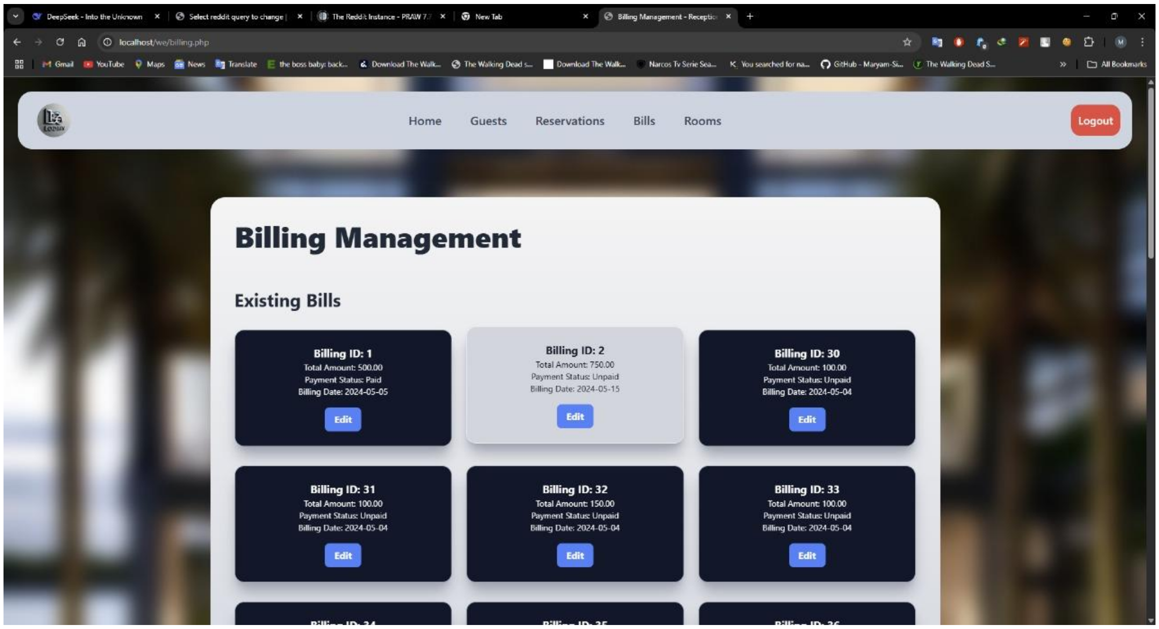
Edit an Existing Reservation:



Guest ID: 1
Room ID: 1
Check-In Date: 05/01/2024
Check-Out Date: 05/05/2024
Reservation Date: 04/25/2024
Status: Confirmed

This screen allows to modify reservation details such as Guest ID, Room ID, Check-In and Check-Out Dates, Reservation Date, and Status. The fields are pre-filled with existing data for easier updating. Once changes are made, the user can click "Save changes" to update the record in the system.

Bill Management Screen:



Billing Management

Existing Bills

Billing ID: 1
Total Amount: 500.00
Payment Status: Paid
Billing Date: 2024-05-05

Billing ID: 2
Total Amount: 750.00
Payment Status: Unpaid
Billing Date: 2024-05-15

Billing ID: 30
Total Amount: 100.00
Payment Status: Unpaid
Billing Date: 2024-05-04

Billing ID: 31
Total Amount: 100.00
Payment Status: Unpaid
Billing Date: 2024-05-04

Billing ID: 32
Total Amount: 150.00
Payment Status: Unpaid
Billing Date: 2024-05-04

Billing ID: 33
Total Amount: 100.00
Payment Status: Unpaid
Billing Date: 2024-05-04

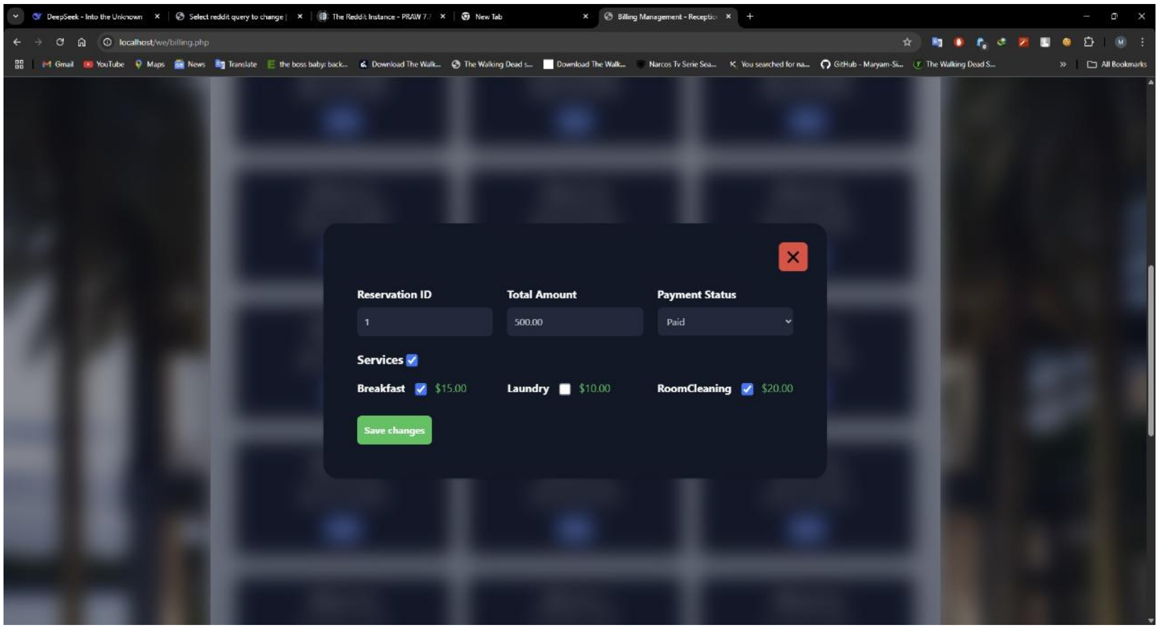
Billing ID: 34

Billing ID: 35

Billing ID: 36

This is the **Bills Management** screen of the Hotel Management System, where a receptionist can view and manage all **existing bills**. Each bill card displays key details such as Billing ID, Total Amount, Payment Status, and Billing Date. Users can click the **"Edit"** button on any card to update the billing information.

Edit Existing Bills:



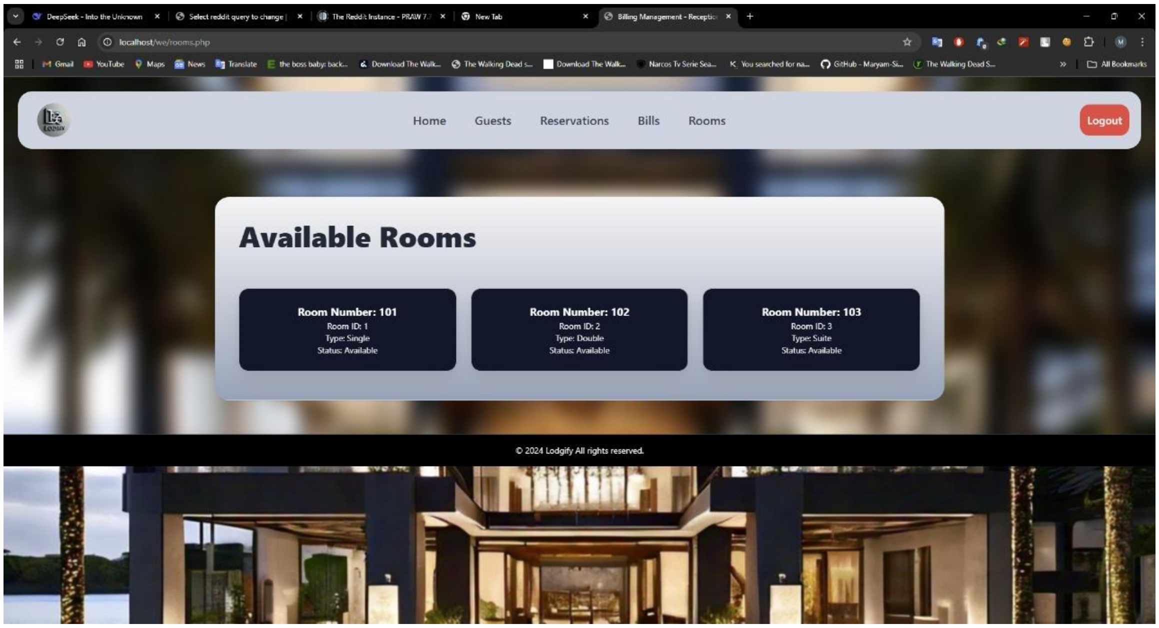
Reservation ID	Total Amount	Payment Status
1	500.00	Paid

Services ☑

Breakfast ☑ $15.00 Laundry □ $10.00 RoomCleaning ☑ $20.00

This screen is the **Edit Existing Bill** interface in the Hotel Management System. It allows a receptionist to update billing details for a reservation by modifying the total amount, payment status, and selected services such as Breakfast, Laundry, and Room Cleaning. The user can check or uncheck services, which are associated with specific charges, and then save the changes. The Reservation ID identifies which booking is being edited. This ensures accurate billing and service tracking per guest.

Available Rooms:



Available Rooms

Room Number: 101
Room ID: 1
Type: Single
Status: Available

Room Number: 102
Room ID: 2
Type: Double
Status: Available

Room Number: 103
Room ID: 3
Type: Suite
Status: Available

© 2024 Lodgify All rights reserved.

This **Available Rooms** screen in the Hotel Management System displays a list of all rooms currently unoccupied and ready for booking. Each room card shows key details including the room number, unique Room ID, room type (Single, Double, Suite), and its availability status. It helps the receptionist or admin quickly view and assign rooms to guests during check-in or reservation processes.