NTPC Summer Training Project

Project Report

2025

Guest House Management System

Supervised By:
Mr Umashankar Dwivedi

Submitted By:
R Neeraj Singh

Abstract

The rapid digitization of workplace services has transformed how organizations manage employee accommodation. Traditional manual processes—often dependent on paper records, phone calls, and in-person verification—can be slow, error-prone, and inconvenient for both staff and guests.

This project report presents the **NTPC Guest House Management System**, a web-based application designed to automate and streamline the allocation, booking, and management of guest house facilities for NTPC employees and visitors. Built using **Flask (Python)** for the backend and **SQLite** for the database, the system enables secure user authentication, real-time room inventory management, automated billing, and comprehensive reporting.

Key features include:

- **Secure Authentication** – Role-based access for authorized personnel
- **Room & Employee Management** – Add, edit, and track records centrally
- **Booking Lifecycle Automation** – From reservation to check-out, with conflict prevention
- **Automated Billing & Reporting** – Instant cost calculation and monthly registers
- **Scalability** – Designed for easy expansion across multiple guest houses

By centralizing all accommodation-related operations into a single, accessible platform, the NTPC Guest House Management System enhances administrative efficiency, reduces operational errors, and improves the overall experience for employees.

INDEX

Introduction

In large organizations like NTPC, guest house facilities play a vital role in accommodating employees, trainees, contractors, and visitors. However, traditional methods of managing these facilities—such as paper-based registers or basic spreadsheets—are often time-consuming, prone to errors, and lack transparency.

The NTPC Guest House Management System addresses these challenges by providing a **secure, user-friendly web platform** to manage bookings, room assignments, and billing. The system empowers administrative staff to manage all aspects of the guest house from a central dashboard while giving employees a smooth and transparent booking experience.

By integrating **real-time availability tracking**, **automated billing**, and **comprehensive reporting**, the system replaces outdated workflows with an efficient, scalable solution. In doing so, it ensures better resource utilization, faster service, and higher satisfaction among guests.

Objectives

The primary objectives of the NTPC Guest House Management System are:

1. **To automate guest house operations** by replacing manual record-keeping with a centralized web application.
2. **To provide secure login and authentication** so only authorized staff can manage bookings and sensitive records.
3. **To track real-time room status** and prevent double-booking or scheduling conflicts.
4. **To generate accurate billing and monthly reports** for revenue analysis and audit purposes.
5. **To ensure scalability and flexibility** so that more facilities or advanced features can be added over time.

Significance

The implementation of this system holds significant value for NTPC:

- **Efficiency:** Automates repetitive administrative tasks, reducing manual workload.
- **Accuracy:** Prevents human errors in booking, billing, and reporting.
- **Transparency:** Makes room availability and booking history visible in real time.
- **Data Security:** Protects sensitive information through authentication and controlled access.
- **Future-readiness:** Provides a foundation for integrating additional modules like online payment or mobile app access.

In short, the system not only improves current operations but also establishes a future-proof digital foundation for NTPC’s accommodation management.

Key Features

1. Secure User Login

- Role-based authentication to ensure only authorized personnel can manage records.
- Session-based security to prevent unauthorized access during active sessions.

2. Employee Management

- Add, view, and update employee details including department and contact information.
- Maintain a historical record of each employee's stays for audit and analysis.

3. Room Management

- Categorize rooms (e.g., Standard, Deluxe) with different pricing.
- Track availability in real time and update room status upon booking or check-out.

4. Booking Management

- Reserve rooms for employees or guests with start and end dates.
- Modify or cancel bookings when required.
- Automatically prevent overlapping reservations.

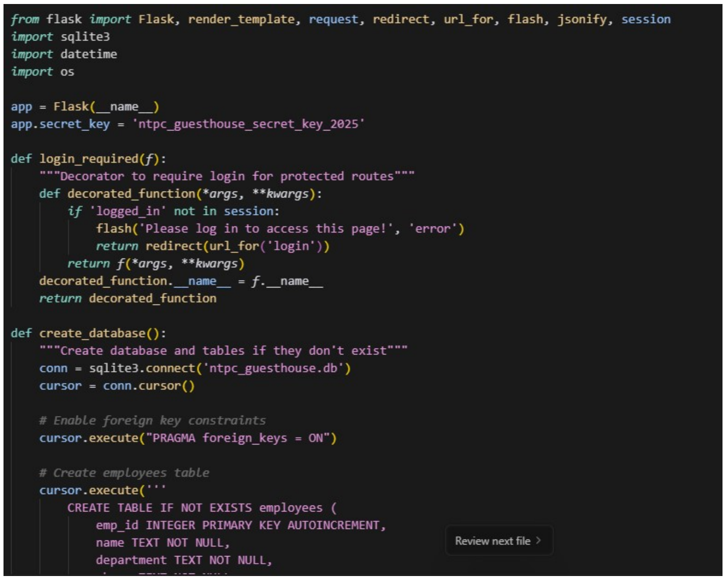
5. Automated Billing

- Calculate total charges based on room category and length of stay.
- Generate detailed invoices for each booking.

6. Monthly Register & Reporting

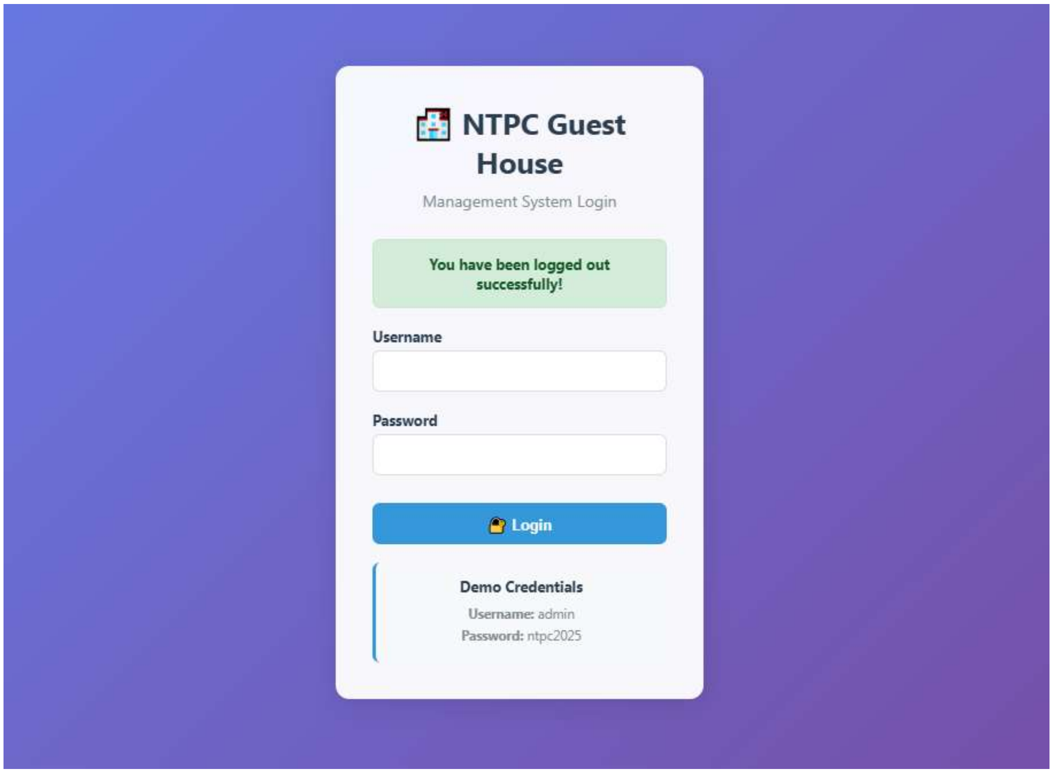
- Produce reports for a given date range with booking details and total revenue.
- Export reports for management review or audits.

#CODE



```
from flask import Flask, render_template, request, redirect, url_for, flash, jsonify, session
import sqlite3
import datetime
import os

app = Flask(__name__)
app.secret_key = 'ntpc_guesthouse_secret_key_2025'

def login_required(f):
    """Decorator to require login for protected routes"""
    def decorated_function(*args, **kwargs):
        if 'logged_in' not in session:
            flash('Please log in to access this page!', 'error')
            return redirect(url_for('login'))
        return f(*args, **kwargs)
    decorated_function.__name__ = f.__name__
    return decorated_function

def create_database():
    """Create database and tables if they don't exist"""
    conn = sqlite3.connect('ntpc_guesthouse.db')
    cursor = conn.cursor()

    # Enable foreign key constraints
    cursor.execute("PRAGMA foreign_keys = ON")

    # Create employees table
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS employees (
            emp_id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT NOT NULL,
            department TEXT NOT NULL,
```

~Sample interface



NTPC Guest House

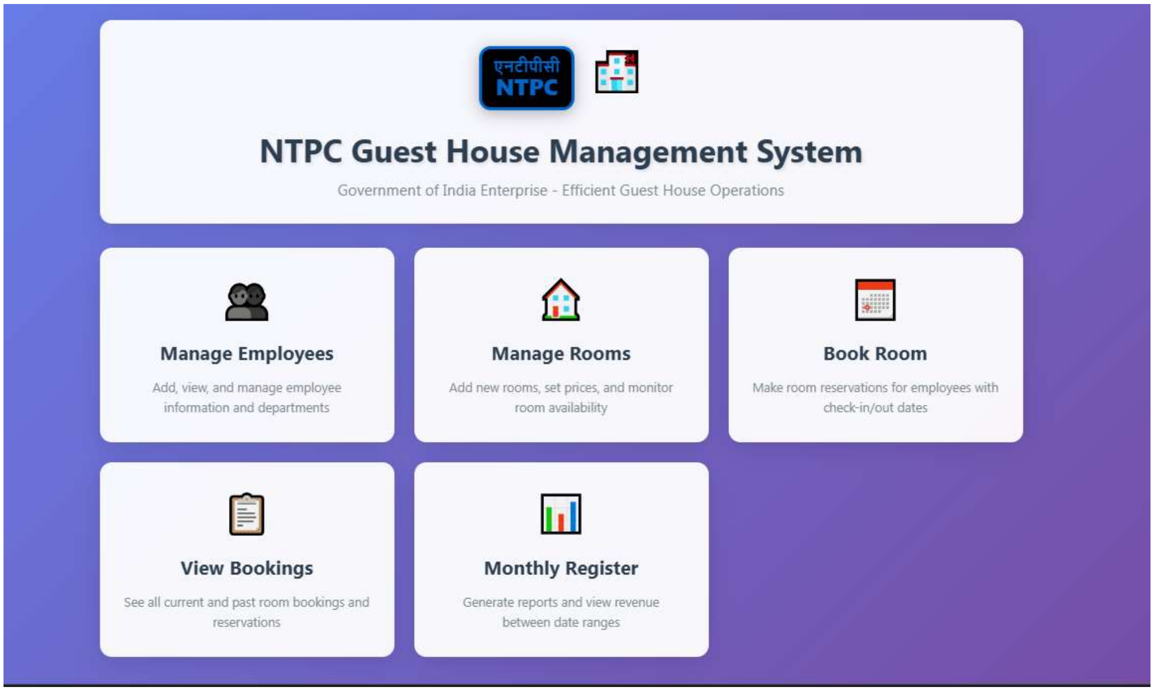
Management System Login

You have been logged out successfully!

Demo Credentials

Username: admin

Password: ntpc2025



NTPC Guest House Management System

Government of India Enterprise - Efficient Guest House Operations

Manage Employees

Add, view, and manage employee information and departments

Manage Rooms

Add new rooms, set prices, and monitor room availability

Book Room

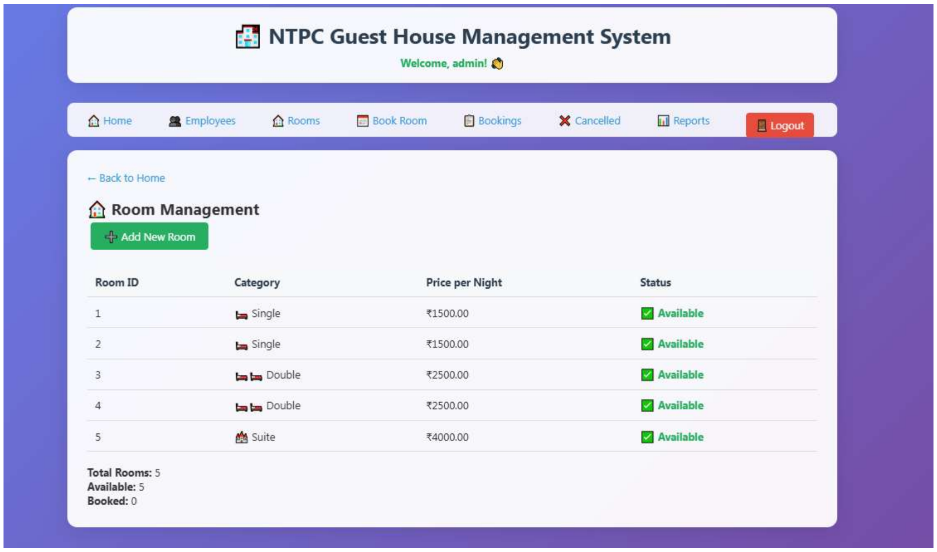
Make room reservations for employees with check-in/out dates

View Bookings

See all current and past room bookings and reservations

Monthly Register

Generate reports and view revenue between date ranges



NTPC Guest House Management System

Welcome, admin!

Room Management

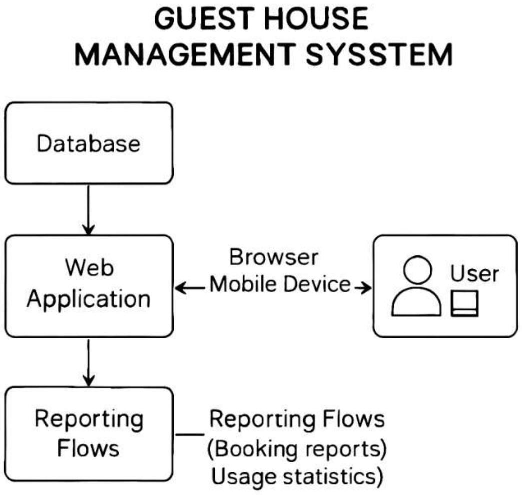
Room ID	Category	Price per Night	Status
1	Single	₹1500.00	✓ Available
2	Single	₹1500.00	✓ Available
3	Double	₹2500.00	✓ Available
4	Double	₹2500.00	✓ Available
5	Suite	₹4000.00	✓ Available

Total Rooms: 5
Available: 5
Booked: 0

System Architecture

The system follows a **client-server architecture**:

- **Frontend:** HTML, CSS, Bootstrap for responsive design.
- **Backend:** Flask (Python) handles requests, processes data, and interacts with the database.
- **Database:** SQLite stores employee, room, and booking data.
- **Flow:**
  1. User logs in via web browser.
  2. Requests are sent to Flask routes.
  3. Database operations update records and retrieve information.
  4. Updated data is displayed in the UI.



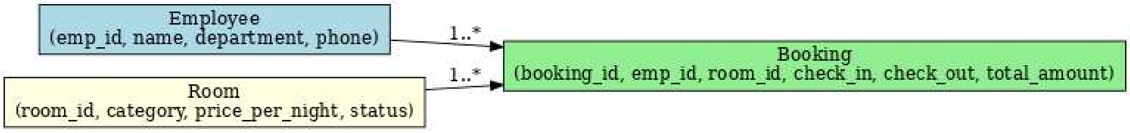
ER Diagram

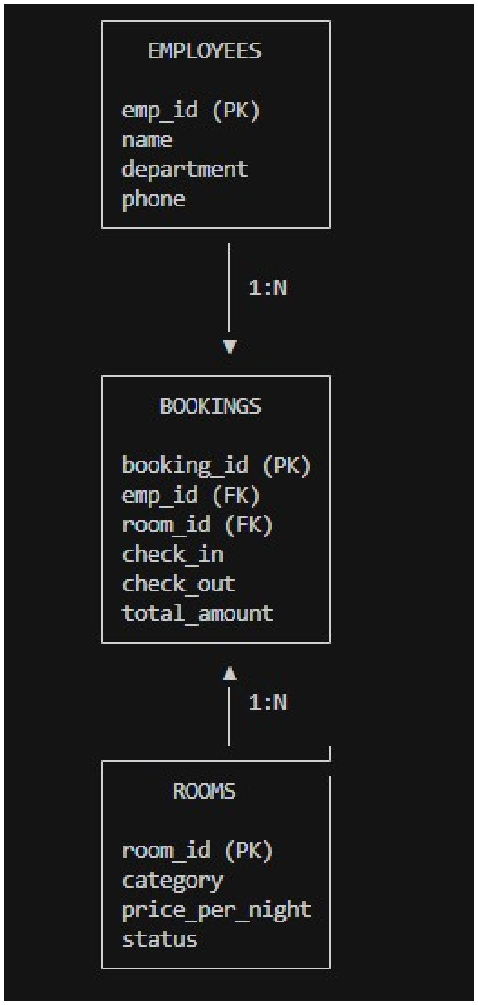
The Entity–Relationship model defines the core tables:

- **Employee:** emp_id, name, department, phone
- **Room:** room_id, category, price_per_night, status
- **Booking:** booking_id, emp_id, room_id, check_in, check_out, total_amount

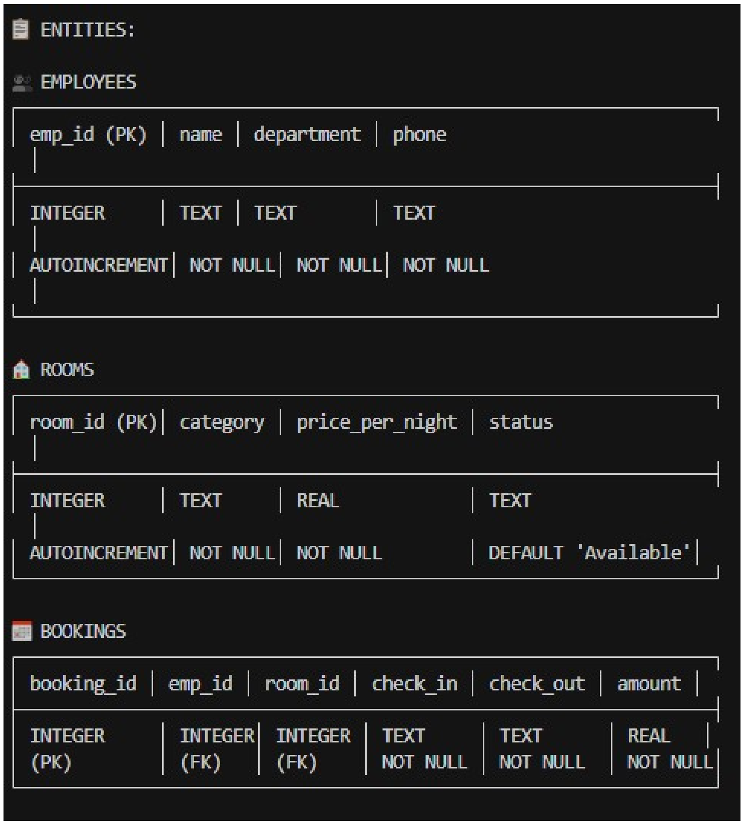
Relationships:

- An Employee can have multiple Bookings.
- A Room can be linked to multiple Bookings over time.


Employee
(emp_id, name, department, phone)
1..*
Booking
(booking_id, emp_id, room_id, check_in, check_out, total_amount)
1..*
Room
(room_id, category, price_per_night, status)



```
┌──────────────────┐
│   EMPLOYEES      │
│                  │
│ emp_id (PK)      │
│ name             │
│ department       │
│ phone            │
└──────────────────┘
          │
          │ 1:N
          │
          ▼
┌──────────────────┐
│   BOOKINGS       │
│                  │
│ booking_id (PK)  │
│ emp_id (FK)      │
│ room_id (FK)     │
│ check_in         │
│ check_out        │
│ total_amount     │
└──────────────────┘
          ▲
          │ 1:N
          │
┌──────────────────┐
│   ROOMS          │
│                  │
│ room_id (PK)     │
│ category         │
│ price_per_night  │
│ status           │
└──────────────────┘
```



```
📋 ENTITIES:

👥 EMPLOYEES
┌───────────────────────────────────────────────────────┐
│ emp_id (PK) │ name │ department │ phone
│
├───────────────────────────────────────────────────────┤
│ INTEGER     │ TEXT │ TEXT       │ TEXT
│
│ AUTOINCREMENT│ NOT NULL│ NOT NULL│ NOT NULL
│
└───────────────────────────────────────────────────────┘

🏠 ROOMS
┌───────────────────────────────────────────────────────┐
│ room_id (PK)│ category │ price_per_night │ status
│
├───────────────────────────────────────────────────────┤
│ INTEGER     │ TEXT     │ REAL            │ TEXT
│
│ AUTOINCREMENT│ NOT NULL│ NOT NULL        │ DEFAULT 'Available'│
└───────────────────────────────────────────────────────┘

📅 BOOKINGS
┌───────────────────────────────────────────────────────┐
│ booking_id │ emp_id │ room_id │ check_in │ check_out │ amount │
├───────────────────────────────────────────────────────┤
│ INTEGER    │ INTEGER│ INTEGER │ TEXT     │ TEXT      │ REAL   │
│ (PK)       │ (FK)   │ (FK)    │ NOT NULL │ NOT NULL  │ NOT NULL│
└───────────────────────────────────────────────────────┘
```

Benefits

- **Streamlined Operations:** Reduces paperwork and manual processes.
- **Improved Guest Experience:** Faster booking and check-in/out process.
- **Better Decision-Making:** Data-driven insights from monthly reports.
- **Cost Savings:** Reduced operational inefficiencies and errors.
- **Scalable Design:** Easily adaptable for more rooms, locations, or features.

Implementation Steps

1. **Requirements Analysis**
   - Gather NTPC-specific needs for room categories, pricing, and reporting.
2. **System Setup**
   - Deploy Flask application and initialize SQLite database.
3. **Training**
   - Educate staff on system usage through hands-on sessions.
4. **Testing**
   - Verify functionality and handle edge cases like overlapping bookings.
5. **Go Live**
   - Launch the system on NTPC's intranet or local network.

Challenges

- **User Adaptation:** Some staff may resist shifting from manual to digital workflows.
- **Data Migration:** Transferring existing booking records to the new database.
- **Technical Maintenance:** Regular database backups and system updates.

Future Scope

Expand to More Locations:

The system can be easily configured to handle multiple NTPC guest houses at different sites by adding basic location/tracking fields to the database and forms.

Bulk Employee & Room Upload:

Provide an option to import employee or room data from Excel/CSV files for faster initial setup and easier annual updates.

Printable and Downloadable Reports:

Add features for direct downloading or printing of monthly booking registers, receipts, and other summary reports in PDF or Excel formats.

Notification Reminders (Internal):

Send simple automated email reminders to admins about check-outs, maintenance due, or high un-booked rates—no external SMS/complex notification services.

Feedback/Rating (Internal Staff):

Allow a simple feedback form for staff to submit issues or suggestions about the guest house, improving service without going public.

Audit and Activity Log:

Maintain a basic log of important admin activities (add, update, delete records) to help with audits and resolve disputes.

User Roles (Admin/Clerk Separation):

Introduce simple role-based access, so only certain users can add/edit rooms or view financial reports.

System Architecture Summary

The NTPC Guest House Management System is designed using a modular, client-server web architecture that emphasizes operational efficiency, data integrity, and user security.

Presentation Layer (Frontend):

*Web-based user interface designed for ease of use by NTPC staff.

*Accessible via modern web browsers within NTPC's secure network.

*Simple, intuitive forms for employee, room, and booking management.

Application Layer (Backend):

*Built with Python (Flask framework).

*Handles business logic for user authentication, booking workflows, room database *management, billing calculations, and report generation.

*Implements session-based security to safeguard sensitive operations.

Data Layer (Database):

*SQLite database for persistent structured storage.

*Three core tables: Employees, Rooms, Bookings, all linked with foreign keys for data integrity.

*Centralized storage enables streamlined queries, aggregation, and audit-friendly recordkeeping.

Security Features:

*Authentication restricts access to authorized personnel only.

*Session management prevents unauthorized actions.

*Data validation ensures consistency and reduces entry errors.

Reporting Module:

*Automated generation of monthly registers and revenue reports.

*Supports administrative analysis, financial review, and audit purposes.

System Interactions:

*All actions performed via the browser are processed by the Flask backend and persisted safely in the SQLite database.

*Admins interact solely through secure web forms; no direct database access is needed