

Hotel Management System

A Survey of Web-Based Room Booking and Hotel Administration Systems

Gayatri Badgujar

Department of Information Technology
Pimpri Chinchwad Polytechnic
Akurdi, Pune, India

Aaryan Pawar

Department of Information Technology
Pimpri Chinchwad Polytechnic
Akurdi, Pune, India

Shivraj Newaskar

Department of Information Technology
Pimpri Chinchwad Polytechnic
Akurdi, Pune, India

Yash Jagtap

Department of Information Technology
Pimpri Chinchwad Polytechnic
Akurdi, Pune, India

Abstract—Many small and medium hotels still keep guest details, room allocation, payments and staff records in registers or loose spreadsheets, which makes retrieval slow and updating error-prone. This paper surveys published work on web-based hotel reservation and hotel information systems and uses it to shape the design of our diploma project, the Hotel Management System. Six peer-reviewed sources are reviewed: three that describe the design and implementation of web-based reservation systems, and three that study the role of information technology in hotels and tourism. The comparison shows that the implemented systems concentrate on online room booking, follow a three-tier browser, server and database structure, and report few measured results, while one of them states that a reservation module alone cannot serve all the needs of a hotel. Based on these findings and on the requirements in our project synopsis, we propose a web-based system with two modules, Admin and User. Customers register, view rooms and services and send booking requests, which the administrator approves or rejects; the administrator also maintains facility, payment, billing and staff salary records. The front end is built with HTML and CSS, and a PHP and MySQL back end running on XAMPP is planned. The paper describes the proposed architecture, workflow, database design, evaluation plan and expected outcomes. No experimental results are claimed, because implementation and testing are still in progress.

Keywords—hotel management system; online room booking; web application; three-tier architecture; PHP; MySQL; literature survey

I. INTRODUCTION

Hotels sell perishable stock: a room that stays empty tonight cannot be sold again tomorrow. Front-office staff therefore need accurate and current information about which rooms are free, which guests are expected, what each guest owes and which employees are on duty. When these details are written in registers or spread across separate files, staff spend time searching for them, the same data is copied into more than one place, and mistakes in bookings and bills become likely.

Research in hospitality and tourism has followed the move from such manual practices to computer- and Internet-based systems for more than two decades. Buhalis and Law [1] reviewed twenty years of eTourism research and described how the Internet changed the way tourism businesses distribute their products and deal with customers. Law, Leung and Buhalis [2]

later reviewed information technology (IT) studies published in 57 hospitality and tourism journals between 2005 and 2007 and concluded that IT had become essential to the competitiveness of these organizations. Sahadev and Islam [3] examined why hotels adopt information and communication technologies (ICTs) and showed that adoption depends both on where a hotel is located and on characteristics of the firm itself.

At the level of individual systems, several authors have built web-based reservation systems for particular hotels. Hu and Gu [4] developed a browser/server (B/S) system offering registration, login, reservation and reservation management. Williams and Micheal [5] designed a reservation management system for a hotel in Lagos using object-oriented analysis, a three-tier architecture and a normalized MySQL database. Vaishnavi, Reddy and Mahenderker [6] implemented an online reservation system with PHP and MySQL that adds a chatbot and e-mail notifications.

Our project, titled *Hotel Management System*, is a web-based application through which a hotel manager can handle room bookings and other hotel activities online. The manager posts the available rooms and services, customers view them and send booking requests, and the administrator approves or disapproves each request. The system is also meant to keep records of customers, room allocation, payments, billing and the hotel's workers. This paper has two aims: to review the work cited above critically, and to use that review to justify and refine the design of our system.

The rest of the paper is organized as follows. Section II states the problem and motivation, and Section III lists the objectives and scope. Section IV reviews the literature, Section V compares it and Section VI identifies the research gap. Sections VII to IX present the proposed methodology, architecture and technologies. Section X explains how the system will be evaluated, Section XI lists the expected outcomes, Section XII discusses advantages and limitations, and Section XIII concludes the paper.

II. PROBLEM STATEMENT AND MOTIVATION

A. The Existing Manual System

The synopsis of this project describes a hotel whose records are maintained by hand. Details of each customer, room allocation, fees and payments, and staff attendance are written on paper and retrieved manually, and information has to be carried between the different sections of the hotel by people. Large amounts of paper are used to record transactions, and no one has central control over the whole process. Three problems follow from this arrangement:

- *Time consumption*: finding a guest's booking, checking whether a room is free or preparing a bill means searching through several records.
- *Difficult updating*: when a booking is changed or cancelled, every record that mentions it must be corrected separately.
- *Inaccurate data*: copying the same details into several registers leads to inconsistencies, for example a room marked free in one register and occupied in another.

Williams and Micheal [5] reported very similar problems at their case-study hotel, which used Excel sheets in place of a database: information could not be stored or retrieved immediately, changes such as cancellations were hard to apply, and accurate reports took a long time to prepare. Because the same symptoms appeared in a hotel that was already partly computerized, the underlying problem seems to be fragmented record keeping, not only the use of paper.

B. Motivation

The literature gives three reasons to move to a web-based system. First, hotels adopt IT to reduce costs, improve operational efficiency and raise service quality [2]. Second, an online booking channel lets a hotel reach customers who are not physically at its front desk; Hu and Gu [4] argue that a hotel without online reservation risks being overlooked by the market. Third, keeping all records in one database removes the duplication that causes the errors described above [5]. For a small hotel with a limited budget, a browser-based system is also attractive because customers and staff need nothing more than a web browser.

Problem statement: to design a web-based Hotel Management System that replaces manual registers with a single database, lets customers view rooms and request bookings online, and lets the administrator approve bookings and maintain customer, room, payment, billing and staff records quickly and accurately.

III. OBJECTIVES AND PROJECT SCOPE

A. Objectives

- 1) To replace the manual process with a computerized one in which hotel records are stored, updated and retrieved through simple, menu-driven pages.
- 2) To let customers register, log in, view available rooms and other hotel services, and send booking requests online.
- 3) To let the administrator approve or disapprove each booking request after checking room availability.

- 4) To maintain customer information, room allocation details, payment details and billing information in one place.
- 5) To keep records of hotel workers and their salary payments.
- 6) To generate reports on the present status of rooms, bookings and payments.

B. Scope

The system covers the two roles defined in the synopsis: the administrator (hotel manager) and the registered user (customer). It is intended for a single hotel and will be developed and tested on a local server. Online payment through a payment gateway, mobile applications, links to external booking websites and multi-branch operation are outside the present scope; some of these are listed as future work in Section XIII.

IV. LITERATURE SURVEY

The six sources reviewed here were identified through Google Scholar and checked against the publisher's page or DOI record. They fall into two groups. The first group studies IT in the hotel and tourism sector and explains why systems such as ours are needed. The second group describes the design and implementation of specific web-based reservation systems and shows how such systems have been built.

A. Information Technology in Hotels and Tourism

Buhalis and Law [1] reviewed eTourism research published over the twenty years up to 2008, drawing mainly on the tourism literature, to analyze how Internet applications had been studied and to anticipate future developments. Their review presents the Internet as a force that changed the structure of the tourism industry, particularly in how businesses distribute their products and interact with consumers. For our project the value of this paper is context rather than technical detail: online booking is now part of how a hotel is expected to operate, not an optional extra.

Law, Leung and Buhalis [2] narrowed this view to studies published between 2005 and 2007 in 57 hospitality and tourism journals, grouping them under consumers, technologies and suppliers. They conclude that IT has become indispensable to the competitiveness, distribution and marketing of these organizations, and that the industry adopts IT to reduce cost, improve operational efficiency and improve service quality and the customer experience. These three motives correspond directly to the problems of the manual system listed in Section II.

Sahadev and Islam [3] looked at the question from the hotel's side. Using a survey of 95 hotels in seven locations in Thailand, they examined how factors related to a hotel's location and to the firm itself affect its propensity to adopt ICT-based facilities, and they proposed a framework for predicting that propensity. Although the study was written mainly for ICT vendors, it reminds system designers that a hotel decides to adopt a system according to its own circumstances. In our interpretation, this makes low cost and ease of use real requirements for a system aimed at small hotels.

B. Web-Based Reservation System Implementations

Hu and Gu [4] developed an online room reservation system on a three-layer B/S architecture, in which users work through a web browser and the web server handles data requests, processing and the generation of dynamic pages. They chose this structure because it lightens the client and simplifies maintenance, since one layer can change without affecting the others. The system ran on Windows 2000 Server with Microsoft SQL Server 2000 and IIS 5.0, using ASP pages built in Dreamweaver MX. Its database was organized into subsets such as room information, customer information, reservations, a consumption sheet for each stay, a customer blacklist and employee information. Customers reserve rooms through an order form, and staff use a reception management interface to confirm, query or delete orders. The paper demonstrates these functions with interface screenshots but does not report any testing or performance measurement.

Williams and Micheal [5] followed a more formal software engineering process. They began with a feasibility study and collected requirements through interviews with hotel staff and a questionnaire. Requirements were modelled as eleven use cases, among them making and cancelling a reservation, adding a customer, changing room prices, checking availability and producing profit and receipt reports, and were divided into functional and non-functional requirements such as usability, reliability, performance and security. The system used a three-tier architecture with HTML and PHP in the presentation tier and a MySQL database served through WAMP, and the database was normalized to third normal form (3NF). Passwords and card numbers were to be stored in encoded form. In their conclusion the authors note that a reservation system cannot operate alone in a hotel: because it concentrates on reservations, it needs accounting and management information systems beside it.

Vaishnavi, Reddy and Mahenderker [6] built an online reservation system with HTML, CSS, JavaScript and Bootstrap on the front end and PHP and MySQL on the back end, running on an XAMPP server. Its database contains user, room, booking, cancellation-log and payment tables. Users can search availability, calculate the cost of a stay, book a room and request cancellation, while an administrator panel manages rooms and bookings. The authors describe two algorithms: one validates booking details and checks availability and occupancy limits (at most two adults and three children per room) to prevent double booking, and the other sends confirmation and cancellation e-mails. A simple chatbot answers common questions. The paper states that the system was tested and handled multiple requests without double bookings, but it gives no quantitative results, and it notes that OTP or e-mail verification is not yet supported.

V. COMPARATIVE ANALYSIS OF EXISTING RESEARCH

Table I summarizes the six sources side by side, and Table II compares the technologies used by the three implemented systems with those planned for our project. Four observations emerge from this comparison.

TABLE I. COMPARATIVE SUMMARY OF THE REVIEWED LITERATURE

Ref. (year)	Type of study	Approach / technology	Key findings or contributions	Limitations (as reported or evident)	Relevance to this project
[1] (2008)	Literature review	Review of 20 years of eTourism research, mainly from tourism journals	Internet has reshaped industry structure and distribution; outlines future research challenges	Review only; no system is designed	Context: online booking is now expected of hotels
[2] (2009)	Literature review	IT studies from 2005–2007 in 57 journals, grouped as consumers, technologies, suppliers	IT is essential for competitiveness; adopted to cut cost, raise efficiency and service quality	Covers a three-year window; no system is designed	Motivates the goals of speed, accuracy and lower cost
[3] (2005)	Survey study	Survey of 95 hotels in seven locations in Thailand; location- and firm-related factors	Framework for predicting a hotel's ICT adoption propensity	Single country; written mainly for ICT marketers	Adoption depends on the hotel's situation, so low cost and simplicity matter (our interpretation)
[4] (2013)	System implementation	Three-layer B/S; ASP on IIS 5.0; SQL Server 2000; Windows 2000 Server	Registration, login, reservation, customer and reservation management; staff confirm or delete orders	Dated proprietary platform; no testing or performance results reported	Supports the three-tier web design and core table set
[5] (2018)	System design and implementation	Object-oriented analysis, 11 use cases, three tiers, HTML and PHP, MySQL on WAMP, 3NF	Structured requirements and a normalized database for a real hotel	Authors state reservation needs accounting and MIS support; no measured results	Model for requirements, database design and normalization
[6] (2025)	System implementation	HTML, CSS, JavaScript, Bootstrap; PHP; MySQL; XAMPP; chatbot; e-mail	Availability and occupancy checks prevent double booking; confirmation and cancellation e-mails	No OTP or e-mail verification; testing described only qualitatively	Closest to our stack; booking-validation logic

1) Common architecture: the three implemented systems [4]–[6] agree on the basic shape of a browser front end, server-side scripts and a relational database arranged in three tiers. The main difference is the technology generation. The system in [4] used a proprietary Microsoft stack from the early 2000s, whereas [5] and [6] used the open-source combination of PHP and MySQL on a local WAMP or XAMPP server. For a student project this combination has the practical advantages of zero licence cost and extensive documentation, which fits the low-cost requirement suggested by [3].

2) Booking confirmation: the systems differ in how a booking becomes final. In [4] customer orders are saved and then confirmed or deleted by reception staff; in [5] reservations are entered by front-desk staff on the customer's behalf; and [6] combines an automatic availability check with confirmation by an administrator. Our synopsis requires the approval pattern, in which a customer's request waits until the administrator approves or disapproves it.

3) Functional scope: all three systems are centred on reservations. The system in [4] adds employee information and a consumption sheet, [5] lists food ordering and profit reports among its requirements, and [6] stores payment details. None of the three describes staff salary records, and only [5] explicitly discusses the need for accounting and management information beyond reservations.

4) Evaluation: the system in [4] is shown through screenshots, [5] defines non-functional requirements without reporting measurements, and [6] describes its testing qualitatively. None of the three reports measured response times, error rates or user satisfaction.

The studies of IT in hospitality [1]–[3] do not describe software, but they supply the reasons for building it and the factors that affect whether a hotel will use it.

TABLE II. TECHNOLOGIES OF IMPLEMENTED SYSTEMS

Aspect	[4]	[5]	[6]	This project
Front end	ASP pages	HTML, PHP	HTML, CSS, JS, Bootstrap	HTML, CSS
Server-side	ASP on IIS 5.0	PHP	PHP	PHP (planned)
Database	SQL Server 2000	MySQL	MySQL	MySQL (planned)
Server package	Windows 2000 Server	WAMP	XAMPP	XAMPP (planned)
Booking made final by	Staff confirm or delete order	Front desk enters booking	Auto check and admin confirmation	Admin approves or rejects request

VI. RESEARCH GAP

A. What the Reviewed Evidence Shows

- The implemented systems [4]–[6] focus on room reservation. Of the three, only [4] stores employee information, and none records staff salary payments.
- The authors of [5] state that a reservation system needs accounting and management information systems alongside it.
- None of [4]–[6] reports measured performance, accuracy or usability results.

- The platform used in [4] is now dated, while [5] and [6] show that PHP and MySQL are sufficient for this class of system.
- Whether a hotel adopts ICT depends on location- and firm-related factors [3].

B. Our Interpretation

From these points we infer that, for a small hotel, a system that combines online booking with the administrative records the hotel already keeps by hand, namely payments, bills and staff salaries, may be more useful than a reservation module on its own. This is our interpretation of the evidence, not a finding of the reviewed papers. Because only six sources were reviewed, other published systems may already combine these functions, and we do not claim that no such system exists.

C. Requirements Carried into This Project

- 1) A single database that holds bookings, payments, bills, service requests, complaints and staff salary records.
- 2) An explicit approval step in which the administrator approves or rejects every booking request after an availability check.
- 3) An evaluation plan with measurable criteria (Section X), so that the project can report evidence and not only screenshots.
- 4) A low-cost, open-source technology stack suited to a small hotel.

VII. PROPOSED SYSTEM AND METHODOLOGY

A. Overview

The proposed system is the computerized version of the existing manual process. It gives quick access to data, records each guest's stay so that the hotel's facilities are used efficiently, and keeps the salary records of the hotel's workers. It is a web application organized into the two modules defined in the synopsis.

B. Modules

1) Admin module: the administrator logs in to a protected dashboard and can view registered users and their payments; add and update information about rooms, facilities and services; view room allocation details and status; approve or disapprove booking requests after checking availability; record payments and generate bills; maintain the list of workers and record their salary payments; reply to complaints; and view reports on the present status of rooms, bookings and payments.

2) User module: a customer first registers and receives a user ID and password, which must be kept secret. After logging in, the customer can view rooms and services, send a room booking request, track its status, view the bill once the booking is approved, request other hotel services and lodge a complaint. As the synopsis states, the user can use a booking only after the administrator has responded.

C. Development Methodology

Fig. 1 shows the six development steps. The synopsis divides design into a primary phase and a secondary phase; we keep that structure and add the steps that come before and after it.

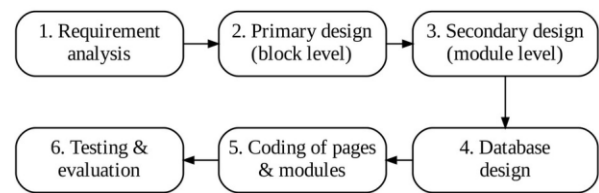


Fig. 1. Development methodology of the proposed system.

1) Requirement analysis: the functions listed in the synopsis are written as functional requirements, and non-functional requirements for usability, reliability and security are added, following the approach of [5].

2) Primary design (block level): the system is divided into blocks so that activities that interact heavily stay in the same block and the flow of information between blocks is kept to a minimum. The blocks are the two modules and the shared database.

3) Secondary design (module level): each block is broken down into small, workable modules such as registration, login, booking request, approval, billing and salary.

4) Database design: tables are derived from the entities of the system and normalized to 3NF, as in [5] (see Section VIII-B).

5) Coding: pages are written in HTML and CSS and connected to PHP scripts that read from and write to the database.

6) Testing and evaluation: each module is tested against prepared test cases, and the complete system is then evaluated using the measures in Section X.

D. Booking Workflow

Fig. 2 shows how a booking moves through the system. A visitor opens the website and views the rooms and services. A new visitor registers, and a registered user logs in. The user selects a room or service and the dates and submits a request, which is stored with the status *Pending*. The administrator reviews pending requests. If the dates are valid and no approved booking overlaps them, the request is approved and a bill is created; otherwise it is rejected and the reason is recorded. The user can then check the status, view the bill or lodge a complaint.

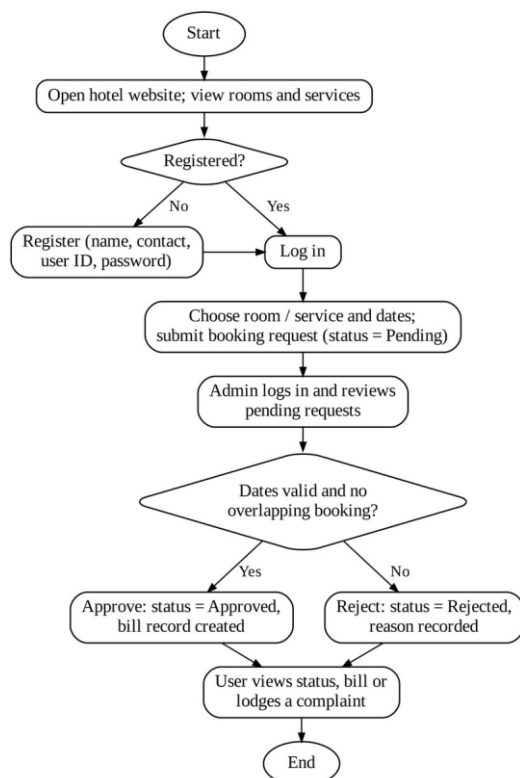


Fig. 2. Workflow of a room booking request in the proposed system.

The availability check before approval is what prevents double booking, which is also the purpose of the validation algorithm in [6]. A new booking b for a room conflicts with an existing approved booking e for the same room when their date ranges overlap, that is, when

$$in(e) < out(b) \text{ and } in(b) < out(e), \quad (1)$$

where $in(\cdot)$ and $out(\cdot)$ denote the check-in and check-out dates of a booking.

If no approved booking satisfies (1) and the check-out date is later than the check-in date, the request is approved and the bill amount is computed as

$$amount = rate \times nights, \quad (2)$$

where $rate$ is the room's nightly rate and $nights$ is the number of nights between check-in and check-out. Charges for services requested during the stay are added to the same bill. The approval, the change of status and the creation of the bill are planned to run inside one database transaction, so that a failure part-way cannot leave a booking approved without a bill.

VIII. PROPOSED SYSTEM ARCHITECTURE

A. Three-Tier Structure

Fig. 3 shows the proposed architecture. Like the systems in [4] and [5], it separates the user interface, the processing logic and the stored data into three tiers, so that a change in one tier does not force changes in the others, and users need only a web browser.

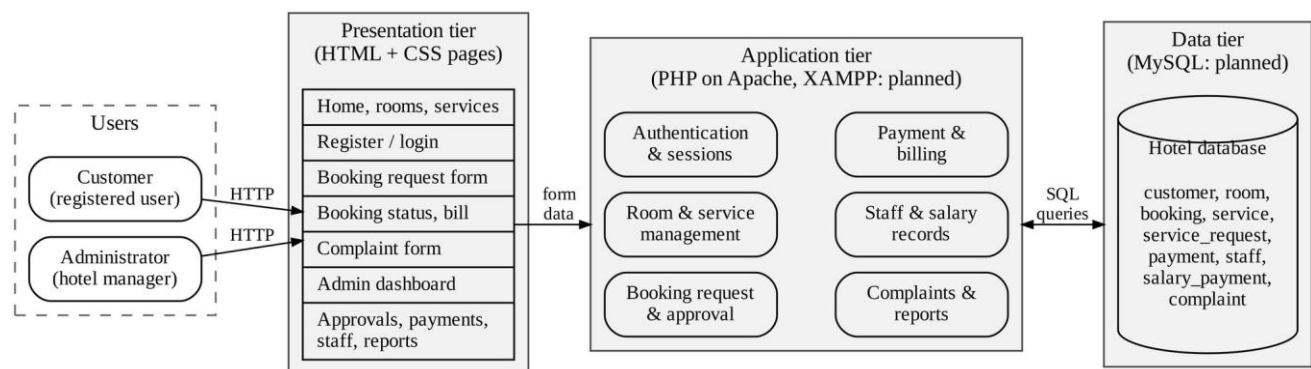


Fig. 3. Proposed three-tier architecture of the Hotel Management System. Components marked "planned" are not yet implemented.

1) Presentation tier: HTML pages styled with CSS, with one set of pages for customers (home, rooms, services, registration, login, booking form, booking status, bill and complaint form) and another for the administrator (dashboard, approvals, payments, staff and reports).

2) Application tier: PHP scripts running on the Apache server supplied with XAMPP. They are grouped into six components: authentication and sessions; room and service management; booking request and approval; payment and billing; staff and salary records; and complaints and reports.

3) Data tier: a MySQL database that holds all records. The PHP scripts send SQL queries to it and present the results as web pages.

B. Database Design

Fig. 4 shows the entity-relationship (ER) diagram of the proposed database, and Table III describes each table. A customer can make many bookings, and a room can appear in many bookings over time. Each approved booking has one bill. Service requests link customers to the services they order, each complaint belongs to one customer, and each salary payment belongs to one staff member.

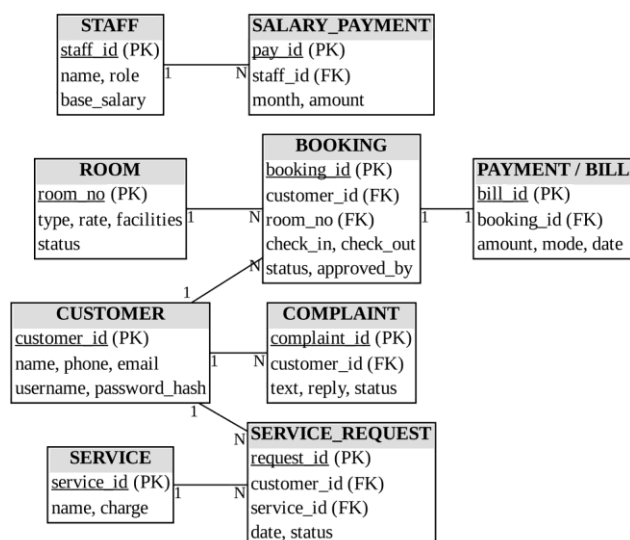


Fig. 4. ER diagram of the proposed database (PK = primary key, FK = foreign key).

The tables are designed to be in 3NF, as recommended in [5]. For example, a room's type and nightly rate are stored once in the ROOM table and not repeated in every booking. The bill stores the amount actually charged, so later changes to room rates do not alter past bills. Passwords are to be stored as hashes generated with PHP's built-in password hashing functions, never as plain text.

TABLE III. PROPOSED DATABASE TABLES

Table	Main fields	Purpose
customer	customer_id, name, phone, email, username, password_hash	Registered users and login details
room	room_no, type, rate, facilities, status	Rooms posted by the administrator
booking	booking_id, customer_id, room_no, check_in, check_out, status, approved_by	Booking requests and their approval status
payment	bill_id, booking_id, amount, mode, date	Bills and payments for approved bookings
service	service_id, name, charge	Other hotel services offered
service_request	request_id, customer_id, service_id, date, status	Services booked by customers
complaint	complaint_id, customer_id, text, reply, status	Complaints and the administrator's replies
staff	staff_id, name, role, base_salary	List of hotel workers
salary_payment	pay_id, staff_id, month, amount	Monthly salary records

IX. IMPLEMENTATION APPROACH AND TECHNOLOGIES

A. Implementation Status

The synopsis includes front-end screens built with HTML and CSS. The server-side logic and the database described in this paper are planned and have not yet been completed. Everything in this section beyond the HTML and CSS pages

should therefore be read as the intended implementation, not a finished one.

B. Choice of Technologies

The synopsis lists HTML and CSS as the front-end software. These languages describe the content and appearance of web pages, but on their own they cannot store records, check a password or keep track of a booking's status. Because the synopsis requires registration, login, administrator approval and stored records, a server-side language and a database are needed. We plan to use PHP with MySQL, the combination used in [5] and [6], because it is free, runs on the existing Windows 10 computer through XAMPP and is widely taught. Table IV summarizes the role of each technology.

TABLE IV. TECHNOLOGIES AND THEIR ROLES

Technology	Role in the system	Status
HTML	Structure of pages and forms (registration, login, booking, admin)	In synopsis
CSS	Layout, colours and styling of all pages	In synopsis
PHP	Server-side logic: sessions, validation, approval, billing, database access	Planned
MySQL	Storage of all records in related tables	Planned
XAMPP (Apache, phpMyAdmin)	Local web server and database administration during development and testing	Planned
Windows 10	Operating system of the development computer	In synopsis

C. Hardware and Software Requirements

The development computer specified in the synopsis has an Intel Core i5-7200U processor and 8 GB of RAM and runs Windows 10. The software consists of HTML and CSS for the front end, the XAMPP package (Apache, PHP, MySQL and phpMyAdmin) for the planned back end, a code editor and any current web browser for testing.

D. Planned Security Measures

All pages other than the public home, rooms and services pages will require login, and PHP sessions will keep track of the logged-in user and role. Administrator pages will check that the session belongs to an administrator. Passwords will be hashed, database queries will use prepared statements to protect against SQL injection, and form input will be validated both in the browser and again on the server. These measures follow the security concerns raised in [5] and [6].

X. EVALUATION METHODS AND PERFORMANCE METRICS

Because the system is not yet complete, this paper reports no results. Instead, this section sets out how the system will be evaluated, so that the final project report can present measured evidence, which the reviewed implementations [4]–[6] did not provide.

A. Functional Testing

Each module will be tested against prepared test cases. Table V lists representative cases; the full set will cover every function in Section VII-B.

TABLE V. REPRESENTATIVE TEST CASES

ID	Test case	Expected result
TC1	Register with valid details	Account created; user can log in
TC2	Log in with a wrong password	Access refused with an error message
TC3	Request a booking with check-out before check-in	Request rejected as invalid
TC4	Approve a request that overlaps an approved booking	Approval blocked; reason recorded
TC5	Approve a valid request	Status Approved; bill equals rate $\times$ nights
TC6	Open an admin page as a customer	Access denied
TC7	Record a salary payment	Payment appears in the staff report

B. Performance Metrics

- *Test-case pass rate*: the percentage of planned test cases that pass.
- *Task completion time*: the time taken to find a booking, check a room's availability and prepare a bill, measured with a stopwatch on the same sample data using the manual register and using the system.
- *Data accuracy*: the number of inconsistencies, such as double-booked rooms or bills that do not match rate and nights, found in a sample of records.
- *Page response time*: the time to load key pages on the local server, measured with the browser's developer tools.
- *User feedback*: ratings from a small group of users, with classmates acting as customers and a staff member acting as administrator, collected through a short Likert-scale questionnaire on ease of use.

XI. EXPECTED OUTCOMES

The following outcomes are expected once implementation is complete. They are goals to be verified through the evaluation in Section X, not measured results.

- A working web application with Admin and User modules running on a local XAMPP server.
- Faster retrieval of booking, payment and staff information than is possible with manual registers.
- No double-booked rooms, because every approval checks for overlapping dates using (1).
- Consistent records held in one normalized database instead of several registers.
- Reports on the present status of rooms, bookings, payments and salaries for the administrator.

XII. ADVANTAGES AND LIMITATIONS

A. Advantages

- Customers and staff need only a web browser to use the system.
- All records are kept in one place, which removes duplicate entries and simplifies updating.

- The administrator keeps control over every booking through the approval step.
- Booking, billing, service, complaint and staff salary records are handled in one system rather than reservations alone.
- The three-tier structure and open-source stack keep cost low and make maintenance easier [4], [5].

B. Limitations

- The system is designed for a single hotel and will run on a local server until it is hosted online.
- There is no payment gateway; payments are recorded by the administrator.
- Manual approval adds a delay compared with the immediate confirmation offered in [6].
- E-mail and SMS notifications are not included in the present scope.
- The planned user evaluation involves a small group, and the literature survey is limited to six sources.

XIII. CONCLUSION AND FUTURE SCOPE

This paper reviewed six published sources on IT in hotels and on web-based hotel reservation systems and used them to design our project, the Hotel Management System. The reviews and the adoption study [1]–[3] explain why hotels move from manual records to online systems, while the implementations [4]–[6] show a consistent three-tier design that has moved from proprietary platforms to PHP and MySQL. The comparison also showed that these systems centre on reservations, that one of them explicitly calls for accounting and management information beside the reservation module, and that none reports measured results.

In response, the proposed system combines online booking requests and administrator approval with payment, billing, service, complaint and staff salary records in a single normalized database. The paper presented its modules, development steps, workflow, architecture, database design, planned technologies and an evaluation plan with measurable criteria. Implementation of the PHP and MySQL back end and the evaluation are the next steps, and their results will be reported in the final project report.

In future, the system can be extended by hosting it online; adding a secure payment gateway; sending booking confirmations by e-mail or SMS, as in [6]; generating printable bills and reports directly from the system; adding analysis of room occupancy over time; supporting more than one branch; and making the pages fully responsive for mobile devices.

ACKNOWLEDGMENT

We thank our project guide and the Department of Information Technology, Pimpri Chinchwad Polytechnic, Akurdi, for their guidance and support during this work.

REFERENCES

- [1] D. Buhalis and R. Law, "Progress in information technology and tourism management: 20 years on and 10 years after the Internet—The state of eTourism research," *Tourism Management*, vol. 29, no. 4, pp. 609–623, Aug. 2008, doi: 10.1016/j.tourman.2008.01.005.

- [2] R. Law, R. Leung, and D. Buhalis, "Information technology applications in hospitality and tourism: A review of publications from 2005 to 2007," *Journal of Travel & Tourism Marketing*, vol. 26, no. 5–6, pp. 599–623, 2009, doi: 10.1080/10548400903163160.
- [3] S. Sahadev and N. Islam, "Why hotels adopt ICTs: A study on the ICT adoption propensity of hotels in Thailand," *International Journal of Contemporary Hospitality Management*, vol. 17, no. 5, pp. 391–401, 2005, doi: 10.1108/09596110510604814.
- [4] P. Hu and D. Gu, "Development and implementation of WEB-based online hotel reservation system," in *Proc. 2nd Int. Symp. Computer, Communication, Control and Automation (ISCCCA-13)*, Advances in Intelligent Systems Research, Paris, France: Atlantis Press, 2013, pp. 235–238, doi: 10.2991/isccca.2013.58.
- [5] K. Williams and A. Micheal, "Design and implementation of reservation management system case study: Grand Ville Hotels," *Journal of Information Technology & Software Engineering*, vol. 8, no. 4, Art. no. 243, 2018, doi: 10.4172/2165-7866.1000243.
- [6] S. Vaishnavi, K. A. Reddy, and T. Mahenderker, "Online hotel reservation system," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 13, no. 4, pp. 1542–1547, Apr. 2025, doi: 10.22214/ijraset.2025.68464.