

Web-Based Boat Booking Management System using Flask Framework

Dr. V. Manimekalai, Associate Professor, Department of Computer Technology, Dr. N.G.P. Arts and Science College, Coimbatore, E-Mail : manimekalaiv@drngpasc.ac.in

K. Saravanan, Student, Department of Computer Technology, Dr. N.G.P. Arts and Science College, Coimbatore, E-Mail : saravananstrzz@gmail.com,



[https://doi.org/ 10.55041/ijst.v2i3.073](https://doi.org/10.55041/ijst.v2i3.073)

Cite this Article: Saravanan, K. (2026). Web-Based Boat Booking Management System using Flask Framework. International Journal of Science, Strategic Management and Technology, 02(03). <https://doi.org/10.55041/ijst.v2i3.073>

License:  This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

1.

ABSTRACT

In many small-scale tourism and water transport services, booking operations are still handled manually. Manual reservation methods often lead to overbooking, poor record maintenance, and lack of transparency in customer management. This paper presents the design and development of a Web-Based Boat Booking Management System developed using the Flask web framework and a relational database management system. The proposed system allows users to register, log in, check boat availability, book boats online, and view booking history. An administrative module is implemented to manage boat details, update schedules, and monitor reservations. The system ensures data accuracy, prevents duplicate bookings, and improves overall operational efficiency. The implementation results show that the system simplifies booking procedures and reduces human errors compared to traditional manual systems.

KEYWORDS

Boat Booking System, Flask Framework, Web Application, Online Reservation System, Database Management System, Tourism Technology

INTRODUCTION

With the rapid growth of digital technology, many industries have shifted from manual processes to automated systems. Online booking platforms are widely used in airlines, railways, and hotels to improve efficiency and customer convenience. However, small-scale boat services often continue to rely on manual reservation methods such as handwritten registers or phone-based bookings.

Manual booking systems create several challenges. They may result in double booking, difficulty in tracking reservations, and poor data management. Additionally, customers do not have real-time access to boat availability information. This lack of automation can affect service quality and customer satisfaction.

To address these problems, this project proposes a Web-Based Boat Booking Management System. The system is developed using Python and the Flask framework for backend processing, while HTML and CSS are used for the frontend interface. A relational database is used to store user information, boat details, and booking records. The objective of this research is to design a simple, secure, and efficient booking system that can be used by small tourism operators.

2. LITERATURE REVIEW

Online reservation systems have become an essential part of modern service industries. Airline and railway booking systems provide real-time availability checking and automated confirmation. These systems reduce manual errors and improve operational transparency.

Several web frameworks such as Django and Flask are commonly used for developing booking applications. Flask is a lightweight Python framework that provides flexibility and easy integration with databases. Research studies indicate that database-driven booking systems improve record maintenance and prevent scheduling conflicts.

Despite these advancements, limited attention has been given to small-scale boat reservation systems. Many local operators lack affordable digital platforms tailored to their needs. Therefore, this project focuses on designing a simple and cost-effective booking management solution specifically for boat services.

3. SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed system follows a client-server architecture. The overall system is divided into three main layers:

3.1 Presentation Layer

The presentation layer consists of the user interface developed using HTML and CSS. It allows users to interact with the system through web pages such as registration, login, booking form, and booking history.

3.2 Application Layer

The application layer is developed using the Flask framework. It handles routing, authentication, booking logic, and validation. When a user submits a booking request, the Flask backend processes the request and verifies boat availability before confirming the booking.

3.3 Database Layer

The database layer stores all relevant information, including:

- User details
- Boat information
- Booking records

- Payment status (if simulated)

The system ensures data consistency and prevents duplicate bookings by validating availability before storing records.

Methodology

1. User registers and logs into the system.
2. User checks available boats for a selected date.
3. User submits booking request.
4. System validates availability and confirms booking.
5. Booking details are stored in the database.
6. Admin manages boats and monitors reservations.

4. IMPLEMENTATION AND RESULTS

The Boat Booking Management System was implemented using Python and Flask framework. The backend logic manages user authentication, session handling, and database interactions. A relational database is used to maintain structured data.

Testing was performed by simulating multiple user interactions. The system successfully handled concurrent booking attempts and prevented double booking through validation checks. The admin panel allowed efficient management of boat schedules and reservation records.

The system improved efficiency compared to manual methods. Booking information was stored securely and could be retrieved easily. The user-friendly interface enhanced accessibility for customers.

4.1 User Registration and Login Module

The system provides secure user registration and login functionality. Users must create an account before accessing booking services. Authentication is handled by the Flask backend to ensure secure session management.

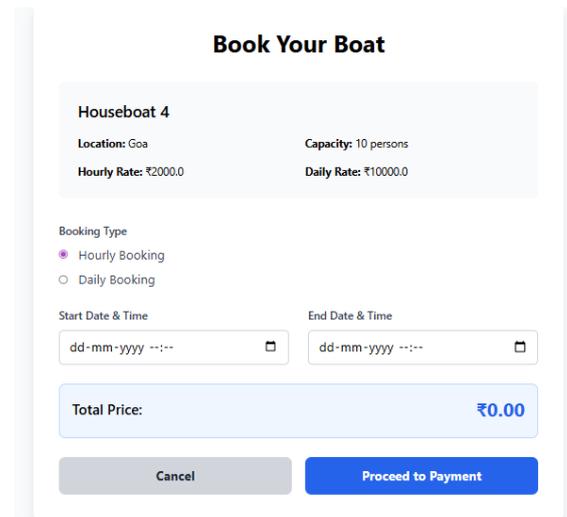
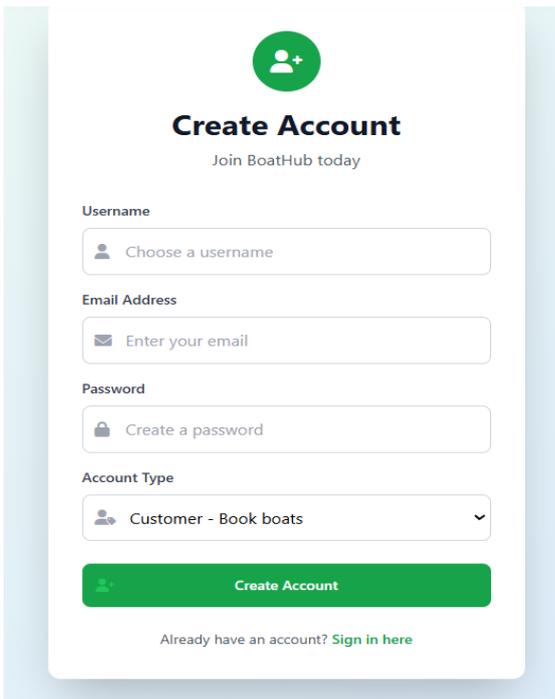


Figure 2: Boat Availability and Booking Page

4.3 Booking History Module

After successful booking, users can view their previous reservations in the booking history section.

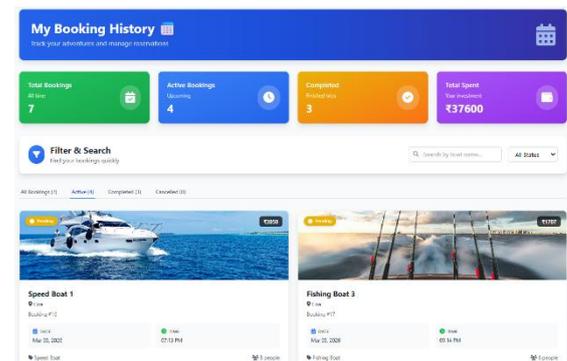
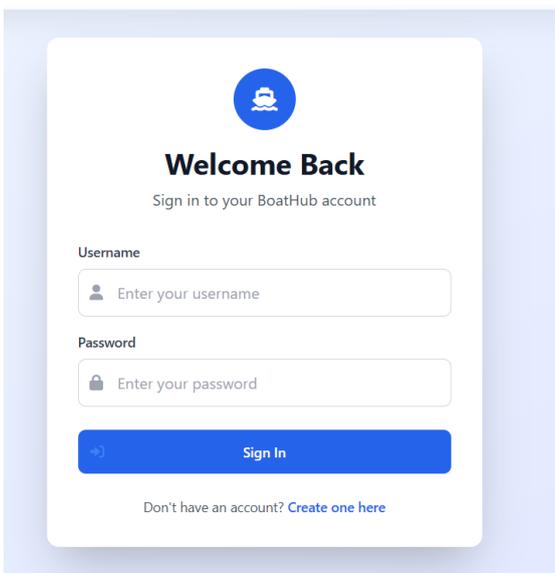
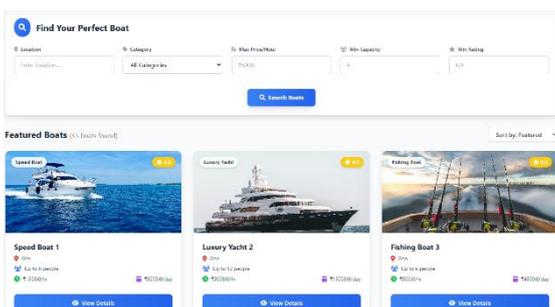


Figure 3: Booking History Display

Figure 1: User Registration and Login Interface

4.2 Boat Availability and Booking Module

Users can check available boats based on selected date and time. The system validates availability before confirming booking to prevent duplicate reservations.



4.4 Admin Management Module

The admin panel allows management of boat details and monitoring of booking records. The admin can add, update, or remove boats and view user bookings.

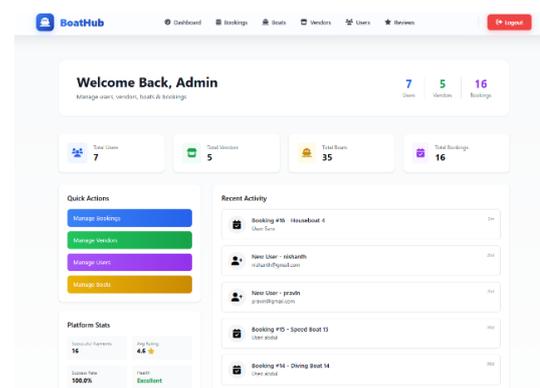


Figure 4: Admin Dashboard Interface

5. DISCUSSION AND FUTURE ENHANCEMENTS

The developed system demonstrates how web technologies can modernize small-scale booking operations. It reduces paperwork, improves data organization, and increases transparency in reservations.

However, the current system includes simulated payment functionality. Future enhancements may include:

- Integration with real-time payment gateways
- Email and SMS notification services
- Role-based access control improvements
- Mobile-responsive design enhancements
- Analytical dashboard for booking statistics

Such improvements can further increase the usability and scalability of the system.

6. CONCLUSION

This paper presented the design and implementation of a Web-Based Boat Booking Management System using Flask framework and a relational database. The system automates the booking process, prevents duplicate reservations, and simplifies administrative tasks. Experimental testing confirms that the system improves booking efficiency and reduces manual errors. The project demonstrates the practical application of web technologies in modernizing tourism-related services.

7. REFERENCES

1. Python Software Foundation, "Python Documentation."
2. Flask Documentation, Pallets Projects.
3. W3C, "HTML Standard Specification."
4. Database System Concepts, McGraw-Hill Publications.