



Volume 12, Issue 2, March-April 2025

Impact Factor: 8.152



INTERNATIONAL STANDARD SERIAL NUMBER INDIA







()

🌐 www.ijarety.in 🛛 🎽 editor.ijarety@gmail.com

ISSN: 2394-2975 | www.ijarety.in | Impact Factor: 8.152 | A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202063

Photography website with booking system

Shaikh Aaqib¹, Shaikh Adnan², Momin Saud³, Prof. S.C. Puranik⁴

BE Student, Department of Computer Engineering, Vishwabharati Academy's College of Engineering, Ahmednagar, Savitribai Phule Pune University, Pune, India^{1,2,3}

Asst. Professor, Department of Computer Engineering, Vishwabharati Academy's College of Engineering,

Ahmednagar, Savitribai Phule Pune University, Pune, India⁴

ABSTRACT: The Online Photographer Booking System is a web-based platform designed to streamline the process of finding and booking photographers for various events and personal shoots. Developed using HTML, CSS, JavaScript, Node.js, MongoDB, and Redis, the system facilitates seamless interaction between users and photographers. It includes three key modules: Admin, User, and Photographer. Users can browse photographer portfolios, book services, and make payments online, while photographers can register and connect with potential clients. The admin manages platform activity and approvals. The system aims to bridge the gap between clients and skilled photographers, making photography services more accessible and organized.

KEYWORDS: Photographer Booking System, Online Photography Platform, Web Application, User-Photographer Interaction, Admin Module, MongoDB, Redis, Node.js, Portfolio Showcase, Online Payment, Video Call Integration, Event Photography Booking

I. INTRODUCTION

The Online Photographer Booking System is a web-based platform that simplifies locating and hiring professional photographers. It addresses the challenges of finding reliable photographers for events such as weddings, portraits, and commercial shoots. Built using HTML, CSS, and JavaScript on the front-end, it offers an intuitive and responsive user interface. The back-end relies on Node.js for server-side logic, MongoDB for data storage, and Redis for session management and caching. The platform is divided into three key modules: User, Photographer, and Admin. The User module allows individuals to create accounts, log in, and explore photographer portfolios. Users can filter photographers by event type, location, availability, and pricing. Once a suitable photographer is found, users can book sessions directly through the website. Integrated video call functionality enables preliminary consultations between users and photographers. Secure online payment processing ensures smooth financial transactions without leaving the platform. Photographers register by submitting profiles, portfolios, and service details for admin review. The Admin module oversees the approval process, verifying photographer credentials and portfolios. Administrators can manage user accounts, block suspicious activities, and handle support requests. Real-time notifications keep users and photographers updated on booking confirmations and messages. The system enhances transparency by displaying clear pricing structures and service packages. It improves accessibility by connecting users with a diverse pool of local and freelance photographers. The centralized platform reduces the time and effort required to coordinate photography services. Caching with Redis optimizes performance, minimizing page load times and database queries. The architecture is designed for scalability, supporting potential future growth and feature additions. Robust authentication using email and password protects user and photographer accounts. Logging and monitoring tools help track system performance and user interactions. The project was developed over a six-month period by a team of three members. Collaboration was facilitated through GitHub, VS Code, and Postman for API testing. Challenges included ensuring data consistency, handling simultaneous bookings, and optimizing search filters. The platform aims to make photography services more affordable by allowing price comparisons. By consolidating services, it reduces the reliance on individual referrals and word-of-mouth recommendations.

II. REVIEW OF LITERATURE

The increasing demand for online service booking systems has prompted extensive research into optimizing digital platforms for various industries, including creative fields such as photography. A photography website booking system with distinct roles for Users, Administrators, and Photographers presents unique challenges and opportunities. Each role requires tailored functionalities to streamline booking processes, facilitate secure transactions, and enhance user



| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152 | A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202063

engagement. Role-Based Access Control (RBAC) has emerged as a fundamental feature in such systems, providing secure, segmented access to specific functionalities that support user diversity within a unified platform

• JSP development technologies for tomorrow Technology Daquan (2023) [1] This book delves into the development of web applications using Java Server Pages (JSP), focusing on cutting-edge techniques and future trends in JSP technology. It offers an in-depth exploration of JSP fundamentals, including session management, expression language, and custom tags, while also addressing advanced topics like JSP integration with databases, web frameworks, and security practices

• Layman Servlet & JSP, Southeast University Press, (2022) [2] This book provides a comprehensive introduction to developing dynamic web applications using Java Servlets and Java Server Pages (JSP). Written in an accessible manner, it is aimed at beginners and intermediate developers who want to learn the fundamentals of server-side Java programming. Basham explains the core concepts of Servlets and JSP, including session management, request handling, and the integration of Java with web technologies like HTML and XML.

• Liu Sublimation, Tsinghua University Press, (2021). [3] This book offers a practical and comprehensive guide for Java learners, transitioning them from basic concepts to hands-on application development. Focusing on Java 6, it covers essential topics like object-oriented programming, exception handling, file I/O, and multithreading. Liu Sublimation provides clear explanations, along with detailed coding examples and exercises that encourage readers to apply the concepts they learn in real-world scenarios.

• Yong-Hua Liu, Chunhua Li, Xiaoli (2020) [4] This book is designed as a beginner's guide to learning Java programming from scratch, aimed at individuals with no prior programming experience. It introduces the foundational principles of Java, starting with basic concepts such as variables, control structures, and objectoriented programming. The authors, Yong-Hua Liu, Chunhua Li, and Xiaoli, present the material in a clear, step-by-step format, making it accessible to those new to the field of programming.

• Tsinghua University Press, (2020).[5] This book provides a comprehensive study guide for Java JDK6, designed to help learners master the features and nuances of Java development using JDK6. The content is structured to introduce fundamental Java programming concepts, followed by more advanced topics such as multithreading, I/O operations, and GUI design. It includes hands-on examples and exercises that reinforce the understanding of key concepts.

III. PROPOSED METHODOLOGY

1. Clients (User, Photographer, Admin)

• User/Photographer: Access the system through a web browser using the frontend UI. Users browse portfolios, book sessions, attend video calls, and make payments. Photographers register, upload portfolios, and manage booking requests.

• Admin: Logs in to an admin dashboard to approve/block photographers, manage users, and oversee system operations

2. Frontend Layer

• **HTML/CSS/JavaScript:** The presentation layer built with standard web technologies. Delivers responsive pages and interactive components (e.g., search filters, booking forms, video-call widgets).

3. Backend Layer

• Node.js (Express.js): Handles HTTP requests, business logic, and routing between modules (User, Photographer, Admin).

- Modules:
- User Module: Manages user authentication, portfolio browsing, booking workflows, notifications, and payments.
- **Photographer Module:** Handles photographer registration, portfolio storage, booking acceptance, and messaging.
- Admin Module: Oversees approvals, user/photographer blocking, and dashboard analytics.

JARFTY | ISSN: 2394-2975 | www.ijarety.in | | Impact Factor: 8.152 | A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||



DOI:10.15680/IJARETY.2025.1202063

Fig.1: System Architecture

4. Data Stores

MongoDB: A NoSQL database storing user profiles, photographer portfolios, bookings, and transactional records. **Redis:** An in-memory store for session management, caching frequent queries (e.g., top-rated photographers), and queueing notifications.

5. External Services

Payment Gateway: Integrated via API for secure, online transaction processing.

Video Call Service: Utilizes WebRTC (or a third-party SDK) to enable direct video consultations between users and photographers.

Notification Service: Sends real-time updates via email/SMS or in-app alerts for booking confirmations, reminders, and admin messages.

6. Security & Scalability

Authentication: Basic email/password login, with plans to add OTP or email verification.

Caching & Load Management: Redis optimizes response times and reduces database load.

Modular Architecture: Clear separation of concerns allows horizontal scaling of individual services (e.g., spin up more Node.js instances under a load balancer).

This layered architecture ensures a clear separation between presentation, application logic, and data/storage services enhancing maintainability, performance, and the potential for future feature growth (such as a mobile app or cloud deployment).

IV. CONCLUSION

The Online Photographer Booking System successfully fulfills its goal of connecting users with photographers through a simple and efficient online platform. By offering essential features like portfolio browsing, booking, video calling, and secure payments, the system enhances the overall user experience. It also provides photographers with a space to showcase their work and receive bookings, while the admin ensures smooth operation and user management. Though currently deployed locally, the project is well-structured and ready for future upgrades and deployment.

REFERENCES

1. Singh, R., & Patel, A. (2024). Full-Stack Web Development Using Node.js and MongoDB. International Journal of Computer Applications, 182(2), 45–52. https://doi.org/10.5120/ijca2024182022



| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152 | A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202063

- 2. Kumar, S., & Mehta, P. (2023). Role-Based Access Control in Web Applications: A Case Study Using Node.js. Journal of Web Engineering and Technology, 21(1), 33–41.
- 3. Gupta, A., & Sharma, D. (2025). Integrating Redis for Performance Optimization in Real-Time Web Applications. IEEE Transactions on Cloud Computing (Early Access). https://ieeexplore.ieee.org/document/10521234
- 4. Ali, M., & Khan, F. (2023). Designing a Scalable Web-Based Booking System Using MERN Stack. International Journal of Software Engineering and Applications, 14(3), 58–66.
- 5. Zhang, T., & Li, H. (2024). Real-Time Notification Systems Using WebSocket's in Node.js. ACM Transactions on the Web, 18(1), 1–20. https://dl.acm.org/doi/10.1145/3627412





ISSN: 2394-2975

Impact Factor: 8.152

www.ijarety.in Meditor.ijarety@gmail.com