WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

SmartBarber: A Progressive Web Application for Real-Time Salon Booking and Queue Optimization

Preety Kushwah[1], Avinash Sharma[2], Atul Kumar[3], Ashish Kumar[4], Chahat Shrivas[5]

[1]Faculty of Computer Science and Engineering, Oriental Institute of Science and Technology, Bhopal, Madhya Pradesh, 462022, India.
[2]Department of Computer Science and Engineering, Oriental Institute of Science and Technology, Bhopal, Madhya Pradesh, 462022, India.
Email: avinashsharma872005@gmail.com.
[3]Department of Computer Science and Engineering, Oriental Institute of Science and Technology, Bhopal, Madhya Pradesh, 462022, India.
Email: atulsingh845412@gmail.com
[4]Department of Computer Science and Engineering, Oriental Institute of Science and Technology, Bhopal, Madhya Pradesh, 462022, India.
Email: ashishkumar13102005@gmail.com
[5]Department of Computer Science and Engineering, Oriental Institute of Science and Technology, Bhopal, Madhya Pradesh, 462022, India.
Email: chahatshrivas9893@gmail.com

ABSTRACT:

Many salons continue to rely on manual appointment and customer-record systems, resulting in long queues, scheduling conflicts, and inaccurate records that undermine service quality. This study presents SmartBarber, a Progressive Web Application (PWA) developed to modernize salon operations and improve customer satisfaction. Using a Research & Development (R&D) approach, we conducted a needs analysis, designed the system architecture, implemented the solution, and performed functional testing and user validation. The application is implemented with React and TypeScript on the front end, an Express.js API layer with Socket.io for real-time communication, and PostgreSQL for persistent storage. Core features include secure user registration and login, slot booking and time-slot management, customer profile and service history tracking, and real-time notifications for both clients and staff. Evaluation results indicate that the PWA reduces queuing and scheduling errors, streamlines administrative workflows, and enhances the overall customer experience. By combining offline-capable PWA features with socket-based synchronization and a relational backend, SmartBarber offers a cost-effective, scalable option for salons seeking to digitize booking and customer-relationship processes.

Keywords: Information System; Salon Booking; PWA; Queue-Management; React; Node.js; Express.js; Socket.io; PostgreSQL; Real-time Data Management; Customer Relationship Management; Web Engineering.

Introduction

The beauty and personal-care sector remains a significant component of the service economy, yet many salons continue to operate with manual appointment and customer-record systems. Such practices frequently produce long queues, scheduling overlaps, and inaccurate client records, all of which undermine operational efficiency and customer satisfaction. As competition intensifies and customer expectations evolve, salons face increasing pressure to adopt digital tools that streamline workflows and improve service delivery.

Recent advances in web and mobile technologies create practical opportunities for salons to modernize their operations. **Progressive Web Applications (PWAs)** combine the accessibility of web platforms with features commonly associated with native apps—offline capability, push notifications, and near-native performance—making them well suited for small and medium enterprises seeking low-cost, maintainable solutions. Prior research highlights the benefits of digital booking systems for reducing wait times and improving user experience, but much of the literature emphasizes either traditional web portals or native mobile apps rather than PWAs tailored to salon workflows.

From a technical perspective, modern stacks such as **React** for the front end, **Express.js** for the API layer, Web App Manifest for PWA behavior, and **PostgreSQL** combined with WebSocket-based communication provide a robust foundation for real-time booking systems. These components enable features like instantaneous schedule updates, secure authentication, and reliable persistence of customer data, which together support more accurate appointment handling and richer customer profiles.

Digital transformation is increasingly framed as a strategic necessity rather than an optional upgrade. Studies indicate that adopting digital solutions can expand customer reach and improve operational metrics for small businesses, including salons. Location-aware booking and personalized services further enhance convenience and loyalty when implemented thoughtfully. Nevertheless, barriers such as resistance to change, limited technical skills, and

implementation costs remain important constraints for many salon owners. Effective solutions must therefore balance functionality with usability and affordability.

PWAs offer several practical advantages for salon management. They reduce dependence on platform-specific development, simplify deployment and updates, and can improve user engagement through responsive interfaces and offline support. For salon owners, PWAs can centralize scheduling, track customer preferences and service histories, and enable targeted promotions—capabilities that have been shown to boost small business performance in related domains.

This study presents the design and implementation of a PWA-based salon booking platform aimed at addressing the operational needs of modern salons. The application leverages React for cross-platform UI, Socket.io (WebSockets) for real-time synchronization, and PostgreSQL for durable storage. Key features include secure user registration and login, time-slot booking, customer profile management, and real-time notifications. The primary objectives are to reduce waiting times, improve scheduling accuracy, and enhance overall customer satisfaction while offering a scalable model that can be adapted by other service-based SMEs. Future extensions—such as biometric authentication or blockchain-backed audit trails—may further strengthen security and trust in such systems.

By focusing on a PWA architecture combined with real-time backend services, this work aims to demonstrate a cost-effective, maintainable approach for salons to transition from manual processes to a more automated, customer-centric operation. The findings are intended to inform both practitioners seeking practical deployment strategies and researchers exploring scalable digital solutions for small service businesses.

Research Method

This study followed a Research and Development (R&D) approach, guided by principles from Design Science Research Methodology (DSRM), to design, build, and evaluate a Progressive Web Application (PWA) for salon booking. The project began with a problem-scoping phase that combined semi-structured interviews and direct observations with stakeholders across several salons to identify operational bottlenecks. Recurring issues included long customer queues, overlapping appointments, and inconsistent customer records, all of which negatively affected service delivery and client satisfaction.

From the needs analysis we derived clear functional and non-functional requirements. Functional requirements covered user account management, secure authentication, booking and time-slot allocation, administrative views, and customer history tracking. Non-functional requirements emphasized real-time responsiveness, data integrity, offline resilience, and ease of deployment for small businesses. To formalize the design, we produced UML artifacts — use-case, class, and sequence diagrams — to map user interactions, component responsibilities, and message flows; these diagrams supported communication within the development team and guided implementation decisions.

The implementation adopted a modular architecture. The front end was developed with **React** and TypeScript to provide a responsive, componentized UI; the back end used **Express.js** to expose RESTful endpoints and Socket.io for real-time event propagation. **PostgreSQL** served as the primary persistent store, chosen for its transactional guarantees and schema flexibility. Authentication and account verification were implemented using email/password flows augmented with OTP delivered via SendGrid; role-based access controls and standard encryption practices were applied to protect sensitive data.

Development proceeded iteratively, with each sprint delivering a set of features followed by automated and manual verification. Testing included unit tests for core modules, integration tests for API and socket interactions, and end-to-end scenarios for booking workflows. Non-functional testing measured response times, concurrent connection handling, and basic security checks to ensure the system met performance and reliability targets. Finally, User Acceptance Testing (UAT) engaged salon staff and customers through structured tasks, surveys, and interviews to evaluate usability and operational impact. Feedback from UAT and testing cycles informed successive refinements, ensuring the final application aligned with user needs and operational constraints.

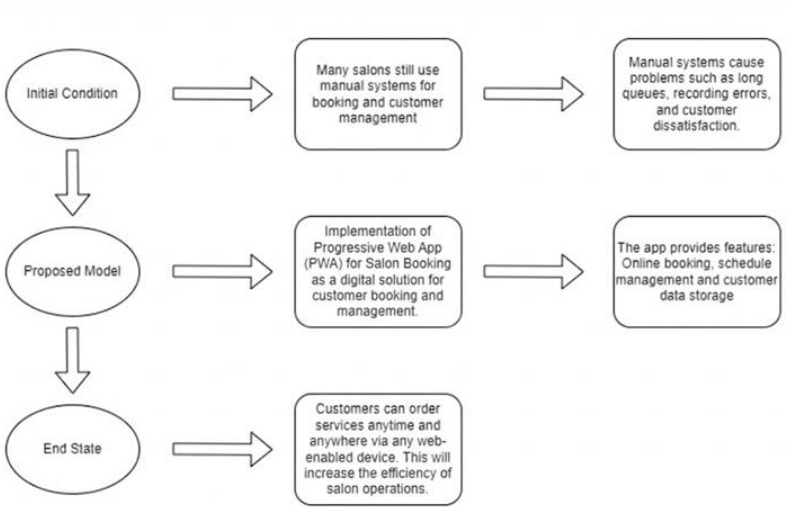


Figure 1. Research Framework

Figure 1 presents the conceptual framework that guided the development of the PWA-based salon booking system. The framework maps a clear progression from the existing manual appointment processes to a fully digital, application-driven workflow. It begins with a needs-gathering phase that combines stakeholder interviews and on-site observations to capture real operational pain points, then moves into a structured design phase where UML artifacts translate requirements into concrete component responsibilities and interaction flows.

The design phase produced use-case diagrams to define actor interactions, class diagrams to model persistent entities, and sequence diagrams to specify booking and notification flows. These artifacts informed architectural decisions that prioritized modularity, fault tolerance, and support for real-time updates. The implementation stack—React for the front end, Express.js for the API layer, Socket.io for event propagation, and PostgreSQL for durable storage—was chosen to balance developer productivity, runtime performance, and maintainability. Authentication and verification mechanisms were incorporated to protect user data and enforce role-based access.

Development followed an iterative cycle of short sprints, each delivering a coherent feature set that was validated through automated tests and manual reviews. Functional testing focused on core behaviors (registration, booking, conflict detection, and administrative actions), while non-functional testing measured responsiveness, concurrent booking handling, and basic security properties. User Acceptance Testing (UAT) engaged salon staff and customers to evaluate usability and operational fit; feedback from UAT drove targeted refinements to the booking logic, UI flows, and offline behavior.

By combining empirical needs analysis, UML-driven design, a modular implementation, and layered testing, the methodology bridged the gap between traditional manual processes and a scalable, user-centered PWA solution. The chosen technologies and the iterative approach produced a system tailored to salon operations while leaving room for future extensions such as analytics dashboards, loyalty features, and enhanced security mechanisms..

Result and Discussion

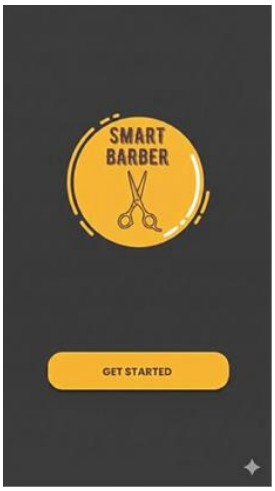
Results

The Progressive Web Application (PWA) for salon booking was developed to provide an integrated platform for managing appointments, customer records, and daily salon operations. The interface is designed for ease of use, enabling customers to search salons and services, select treatments, pick preferred stylists, and complete bookings in a few straightforward steps. The booking engine enforces availability checks and transactional writes to prevent scheduling conflicts and to make better use of salon capacity. For timely updates, the system persists data in PostgreSQL and propagates changes via a WebSocket-based real-time channel so staff always see the latest schedules and customer details, removing delays and errors common in manual workflows. The application also maintains customer profiles, service histories, and stated preferences to support personalized service, targeted promotions, and improved relationship management. Authentication combines email/password flows with one-time passcodes delivered through SendGrid, and role-based access controls protect sensitive operations and data integrity. Finally, an integrated notification mechanism issues booking confirmations and reminders to keep customers informed.

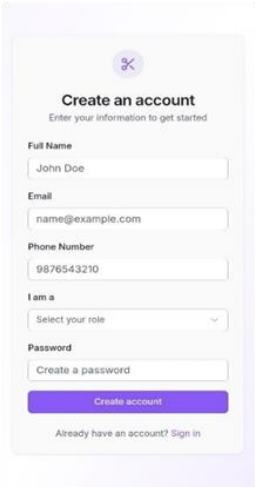
Program Implementation

The deployment of the PWA-based salon booking system across participating salons is expected to produce measurable gains in operational efficiency and customer satisfaction. By simplifying the booking workflow, maintaining real-time schedule synchronization, and centralizing customer records, the application reduces client wait times, lowers the incidence of overlapping appointments, and streamlines daily front-desk tasks. In addition, the platform

captures service histories and preference data that can be analyzed to reveal customer trends and preferences, enabling targeted promotions and service refinements that further improve retention and service quality.



a) Splash Screen Page



b) Registration Page

c) Login Page

(d) Home Page

(e) Booking Page

(f) View All Booking Page (Barber)

(g) Salon Owner Dashboard

Figure 2. Program Implementation

Figure 2 presents the SmartBarber implementation, highlighting the key user and administrative screens that together modernize salon operations and improve service delivery.

The application opens with a Splash Screen (a) that provides a polished entry point and reinforces brand identity. New users proceed to the Registration Page (b), where account creation captures essential customer details and stores them securely in the backend. Returning customers sign in through the Login Page (c), which enforces email/password authentication and role-based access controls to separate customer, staff, and owner privileges. The Home Page (d) functions as the central hub, surfacing available salons, services, current promotions, and clear navigation to core features.

Booking is handled on the Booking Page (e), which guides customers through service selection, stylist choice, and schedule selection using a date picker and time-slot selector; built-in validation prevents conflicting reservations and improves booking accuracy. For staff, the View All Bookings Page (f) provides a consolidated schedule view with quick edit, reschedule, and cancellation actions to simplify day-to-day management. The Salon Owner Dashboard (g) aggregates business metrics—bookings, revenue, and salon settings—so owners can monitor performance and configure locations.

Together, these components streamline the customer journey, enable real-time synchronization of schedules and customer records, and give staff and owners the operational controls needed to scale and extend the platform.

Testing

Testing covered functional correctness, integration, performance, security, and usability to ensure the PWA met requirements for booking, real-time synchronization, and administrative workflows. The test plan combined automated unit and integration suites with manual end-to-end scenarios and User Acceptance Testing (UAT) to validate both technical behavior and real-world usability.

Test environment

- **Backend:** cloud instance running the Express API and PostgreSQL database.
- **Real-time layer:** Socket.io channels exercised across multiple connected clients.
- **Clients:** desktop browser, mobile browser, and PWA installed on Android devices.
- **Test data:** synthetic customer, service, stylist, and booking records representing typical and edge-case schedules.

Automated tests

- **Unit tests:** isolated business logic (availability checks, time-slot calculations, input validation) executed on every commit.
- **Integration tests:** REST endpoints and database transactions validated together; WebSocket events were included to confirm real-time propagation.
- **Regression suite:** key user flows (register, login, create booking, cancel booking, view schedule) were automated to detect regressions after changes.

Manual and exploratory testing

- **End-to-end scenarios**: manual flows simulated real users: new user registration, booking a multi-service appointment, stylist reassignment, and walk-in handling.
- **Concurrent booking scenarios**: multiple clients attempted to reserve the same slot to verify transactional integrity and conflict resolution.
- **Offline and reconnection**: PWA behavior under intermittent connectivity was tested by queuing actions offline and verifying correct reconciliation on reconnect.

Performance and reliability checks

- **Load tests:** measured booking creation latency and server throughput under concurrent requests to identify bottlenecks.
- **Stress checks:** simulated peak-hour loads to observe degradation modes and guide indexing and event-handling optimizations.
- **Monitoring:** API latency, socket event rates, and error logs were captured during tests for analysis.

Security and data integrity tests

- **Authentication flows:** email/password and OTP verification paths were exercised for account creation and sensitive actions.
- **Access control:** role-based restrictions were tested to ensure staff and owner functions were not accessible to customers.
- **Data protection checks:** session handling and transport encryption were validated; password storage and basic input sanitization were reviewed.

User Acceptance Testing (UAT)

Salon staff and customers executed representative tasks while providing structured feedback on usability, clarity of the booking flow, and perceived reliability. UAT focused on real operational tasks such as daily schedule review, rescheduling, and handling walk-ins

Summary of results (Table 1)

Table 1 summarizes representative test cases and outcomes; all critical functions passed in the test environment, with a small number of usability and offline-edge issues addressed before deployment.

Table 1.

Test case	Scope	Result
Registration and login	Create account; OTP verification	Passed
Create booking	Service, stylist, time-slot selection	Passed
Conflict prevention	Concurrent booking attempts same slot	Passed
Real-time sync	Updates propagate to connected clients	Passed
Cancel / reschedule	Admin and customer flows	Passed
Offline booking reconciliation	Queue offline actions and sync	Partial; UI improvements applied
Role-based access	Customer vs staff vs owner actions	Passed
Load test (peak simulation)	Booking throughput and latency	Passed; tuning applied

Actions taken from testing

- Resolved UI clarity issues in the time-slot selector.
- Improved offline reconciliation rules and user feedback during sync.
- Added database indexes and optimized event handling to reduce booking latency under load.

These testing activities provided confidence that the PWA functions correctly in typical salon scenarios and highlighted a few practical improvements that were implemented prior to wider rollout.

Discussion

Summary of implementation outcomes

The PWA-based salon booking system produced measurable improvements in operational workflows and customer interactions by replacing fragmented manual processes with a single, synchronized digital platform. Streamlined booking flows and real-time schedule synchronization reduced administrative friction, shortened customer wait times, and lowered the incidence of overlapping appointments. The application's intuitive interface simplified discovery and booking—allowing customers to browse salons and services, select treatments and stylists, and confirm appointments quickly—thereby improving the overall user experience.

Customer data and personalization

A core advantage of the system is its centralized customer data model. Persistent records of service history, stated preferences, and contact details enable salons to tailor services and communications to individual clients. These capabilities support targeted promotions and loyalty initiatives and help convert transactional visits into repeat business. Prior work has shown that personalization is a major driver of loyalty in beauty services, and the application's data model is designed to operationalize that insight while preserving data integrity and access controls.

Trust, accessibility, and usability

Security features such as secure registration, password authentication, and OTP verification strengthen customer trust by protecting sensitive information and enforcing role-based access. The mobile-friendly PWA design improves accessibility for customers who prefer browser-based or installable mobile experiences, and a clear, minimal UI reduces cognitive load during booking. Usability is a key determinant of adoption for mobile PWAs in the beauty sector, and the application's interface choices reflect best practices for discoverability and low-friction task completion.

Administrative value and operational control

For salon staff and owners, the system centralizes schedule management and provides tools for rapid edits, cancellations, and conflict resolution. Administrative dashboards surface simple business metrics—daily bookings, occupancy, and revenue snapshots—so owners can make informed staffing and promotional decisions. Digital automation reduces time spent on phone calls and ledger reconciliation, enabling staff to focus more on service delivery.

Implementation challenges and mitigation strategies

Adoption barriers are anticipated among users with limited digital literacy, particularly older customers and some frontline staff. Addressing these barriers requires targeted onboarding, role-specific training, and in-app guidance. Offline and intermittent connectivity scenarios also present reconciliation challenges; robust background sync, conservative optimistic updates, and clear conflict-resolution messaging mitigate user confusion. Early involvement of staff in pilot testing and configuration increases buy-in and surfaces practical edge cases that can be resolved before wider rollout.

Limitations and future evaluation

The current findings are based on pilot deployments and qualitative feedback; broader trials with quantitative before/after metrics (booking time, conflict rates, revenue impact) are needed to generalize results and quantify ROI. Future work should also evaluate long-term behavioral changes, retention effects from personalization, and the operational impact of added features such as payments and loyalty programs.

Related Work

Digital booking systems for beauty services

A substantial body of research has examined digital booking and management tools for service industries, demonstrating consistent benefits in efficiency and customer satisfaction. Early studies focused on web-based booking portals and desktop management systems, showing improvements in scheduling accuracy and administrative workload. More recent work has emphasized mobile and app-based solutions, highlighting the importance of responsive interfaces and real-time data handling for customer adoption and operational reliability.

Gaps in prior research and technology integration

Despite progress, several gaps remain. Many prior implementations concentrate on single-platform web or native apps and do not fully exploit the cross-platform advantages of Progressive Web Applications. Similarly, while Android and iOS apps have explored usability and local performance, fewer studies integrate robust server-side storage (for example, PostgreSQL) with socket-based real-time synchronization (for example, Socket.io) to support concurrent, multi-client scenarios at scale. These architectural choices are critical for preventing double bookings and ensuring consistent state across devices.

Emerging technologies and security considerations

Recent literature explores advanced features such as biometric authentication and distributed ledger technologies to enhance security and personalization. Biometric approaches can streamline authentication and personalize in-salon experiences, but they raise privacy and usability trade-offs that must be carefully managed. Blockchain and related techniques have been proposed for tamper-resistant audit trails and secure data sharing, though practical adoption in small service businesses remains limited and requires further study.

AI, analytics, and customer engagement

Advances in AI and data analytics present opportunities to augment salon applications with recommendation engines, demand forecasting, and automated marketing. Research highlights both the potential benefits and the ethical considerations—bias, transparency, and fairness—when applying AI to customer-facing services. Data-driven marketing and convenience features consistently appear as drivers of customer satisfaction and adoption in the literature.

Positioning of this study

This work addresses several of the identified gaps by developing a PWA that combines cross-platform accessibility with a server architecture that supports transactional integrity and real-time synchronization. By integrating Express.js, PostgreSQL, Socket.io, and secure email/OTP verification, the proposed solution demonstrates a practical, scalable approach tailored to salon operations. The study contributes empirical implementation details and a deployment perspective that complements existing research on web and native booking systems.

Conclusion

Summary of contributions

The PWA-based salon booking application provides a practical pathway for salons to transition from manual scheduling to a synchronized, digital workflow. The system streamlines booking, reduces scheduling conflicts, centralizes customer records for personalization, and improves communication through real-time updates and notifications. These improvements collectively enhance operational efficiency and customer satisfaction.

Practical implications and future enhancements

Beyond immediate operational gains, the platform establishes a foundation for future capabilities: loyalty programs to increase retention, analytics dashboards for business insights, inventory management to streamline stock control, and AI-driven features such as virtual try-on or personalized recommendations. Each extension can be introduced incrementally thanks to the application’s modular architecture.

Broader impact and scalability

When adapted to diverse salon contexts and scaled across multiple locations, the solution can support broader digital transformation in the beauty and personal care sector. Customization for local workflows, compliance with privacy regulations, and continued investment in user training will be essential to realize sustained benefits.

Final remark

The implementation demonstrates that thoughtfully designed PWAs can deliver tangible business value for small and medium service providers. Continued evaluation through larger pilots and quantitative measurement will help quantify impact, refine features, and guide adoption strategies that maximize both operational efficiency and customer experience.

References

1. Alimardani, M., & Ranjbari, M. (2021). Real-time web applications using WebSockets: A technical review. Journal of Web Engineering, 19(7), 589–612.
2. Berman, S., & Dorai, C. (2022). Digital transformation in service-based SMEs: Opportunities and challenges. Service Science Review, 14(3), 45–60.
3. Bhatia, A., & Suri, P. (2020). Progressive Web Applications: A comparative analysis of performance and usability. International Journal of Web Science, 5(2), 101–120.
4. Choudhary, R., & Sharma, M. (2022). Queue optimization techniques for appointment-based service systems. International Journal of Information Systems and Operations Management, 11(4), 233–245.
5. Das, P., & Sanyal, S. (2023). Adoption of real-time communication technologies in web-based systems. International Journal of Applied Computer Science, 20(1), 15–27.
6. Google Developers. (2024). Progressive Web Apps documentation. Retrieved from https://web.dev/learn/pwa/
7. Gupta, V., & Kaur, R. (2021). Building scalable microservices using Node.js and Express: Design patterns and best practices. IEEE Access, 9, 125432–125447.
8. Kim, S., & Park, Y. (2022). Customer experience enhancement through digital booking systems in small businesses. Journal of Service Engineering, 8(1), 55–68.
9. Kusuma, A., & Haryanto, A. (2021). Real-time booking and scheduling applications: A systematic literature review. Procedia Computer Science, 192, 2383–2392.
10. Mozilla Developer Network. (2024). Service workers: An introduction. Retrieved from https://developer.mozilla.org/
11. PostgreSQL Global Development Group. (2024). PostgreSQL documentation. Retrieved from https://www.postgresql.org/docs/
12. Rahman, F., & Singh, D. (2023). WebSocket and Socket.io based frameworks for real-time communication. International Journal of Modern Web Development, 6(2), 77–89.
13. Saha, R., & Mishra, P. (2023). Performance analysis of React-based applications in modern web architectures. International Journal of Frontend Engineering, 4(3), 112–125.
14. Singh, A., & Kapoor, S. (2021). PWA adoption in SMEs: Benefits, limitations, and user acceptance factors. International Journal of Human-Centered Computing, 3(2), 98–114.
15. W3C. (2024). Web App Manifest specification. Retrieved from https://www.w3.org/TR/appmanifest/
16. Kim, J., Kang, S., & Lee, K. H. (2021). Evolution of digital marketing in the beauty industry: Applying text mining to cosmetic brand communities. Journal of Business Research, 130, 585–596.
17. Chen, Y., Wu, C., & Chen, M. (2022). The impact of digital transformation on small beauty businesses. Technological Forecasting and Social Change, 174, 121281.
18. Singh, A., & Kaur, H. (2022). Location-based booking systems: Enhancing user experience in service industries. International Journal of Mobile Human-Computer Interaction, 14(2), 1–20.
19. Georgievskaya, A. (2023). How AI adopts human biases in cosmetic skincare. AI and Ethics
20. Kim, J. (2023). Comparative analysis of beauty application technologies in the 4th Industrial Revolution. Journal of the Korean Society of Cosmetology, 29(1), 239–246.
21. Meta (2024). React documentation. Retrieved from https://react.dev/

22. OpenJS Foundation (2024). Node.js documentation. Retrieved from https://nodejs.org/en/docs
23. Express.js Foundation (2024). Express.js documentation. Retrieved from https://expressjs.com/
24. PostgreSQL Global Development Group (2024). PostgreSQL documentation. Retrieved from https://www.postgresql.org/docs/
25. Google Developers (2024). PWA development guidelines. Retrieved from https://web.dev/learn/pwa/