IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

Centralized Complaint Management System

M. Dharshini[1], V. Ruban Kumar[2], A. Nirendrasethupathi[3]
[1]Assistant Professor, Dept. of Information Technology
Students, Dept. of Information Technology2,3
K.L.N. College of Engineering, Sivaganga, India
Dharshinibtecit@gmail.com, rubankumar2572004@gmail.com, nirendrasethupathi@gmail.com.

Abstract: *The SmartComplaint platform is designed to address the growing challenges organizations face in managing large volumes of complaints from customers and employees. In many organizations, complaint handling is inefficient, unstructured, and lacks proper tracking, leading to delays and poor user experience. This system provides a centralized solution that allows multiple organizations to manage complaints efficiently within a single platform while ensuring complete data separation. It introduces a structured role-based approach where different users have specific responsibilities and controlled access.*

The platform ensures authenticity of complaints through flexible verification methods tailored to each organization's requirements. It also includes a secure employee approval process to prevent unauthorized access. A well-defined complaint lifecycle enables smooth tracking from creation to resolution, ensuring transparency and accountability. Real-time notifications keep all stakeholders informed about important updates, reducing communication gaps.

Additionally, the system supports scalable usage through plan-based access, making it suitable for organizations of different sizes. The solution improves operational efficiency, enhances communication, and ensures faster complaint resolution. Overall, SmartComplaint provides a reliable, secure, and scalable foundation for modern complaint management across diverse industries.

Keywords: Complaint Management System, Role-Based Access Control, Multi-Tenant Architecture, Data Security, Complaint Lifecycle, User Authentication, Real-Time Notifications, Workflow Automation, Scalable Platform, Issue Tracking System, Predictive Analytics, Process Optimization

I. INTRODUCTION

In today's fast-growing digital world, organizations receive a large number of complaints from customers and employees every day. Managing these complaints efficiently and transparently has become a major challenge. Traditional systems are often slow, unorganized, and unable to meet modern expectations. Smart Complaint is designed to simplify and streamline the complaint handling process through a centralized platform. It allows organizations to manage, track, and resolve complaints in a structured and timebound manner. The system ensures better accountability and improved communication between users and management. Overall, SmartComplaint helps organizations enhance service quality and build stronger trust with their users.

II. RELATED WORK

Several approaches have been proposed for complaint management systems. Traditional methods rely on manual processes or basic ticketing tools, making them simple but inefficient for handling large volumes of complaints. Rule-based systems introduce predefined workflows, improving structure but lacking flexibility for complex organizational needs. Web-based and cloud-based platforms provide centralized access and better tracking, but often struggle with data isolation in multi-organization environments. Recent solutions incorporate role-based access control and notification mechanisms to enhance security and communication; however, they frequently lack comprehensive verification processes and real-time updates. While these systems improve complaint handling efficiency, there remains

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

a need for an integrated solution that ensures secure authentication, scalable architecture, complete lifecycle management, and transparency, which forms the motivation for this work.

III. METHODOLOGY

[1] Data Collection and Preparation

The data used in this system is collected from multiple organizations to represent real-world complaint handling scenarios. The dataset includes complaint details such as complaint type, description, priority level, submission time, user information, and status updates. Each record represents an individual complaint registered within the platform.Key parameters influencing complaint processing include complaint category, urgency level, department assigned, response time, and resolution status. These features are essential for tracking, managing, and analyzing complaints efficiently.

To ensure data quality, preprocessing steps such as removal of incomplete records, normalization of textual inputs, and encoding of categorical attributes (e.g., complaint type, priority, and department) were applied. The data is structured to support efficient storage, retrieval, and processing within the system. Additionally, simulated data is generated to represent different complaint scenarios such as high-volume submissions, delayed responses, and resolved cases. This helps in testing system performance and scalability under various conditions.

[2] Feature Engineering and Complaint Modelling

Feature engineering is used to enhance the effectiveness of complaint management and tracking. Important features such as complaint priority, response time, and resolution duration are derived to better understand system performance. Categorical attributes like complaint type, department, and user role are encoded to ensure compatibility with system logic and analytics modules. Derived features such as complaint aging, average resolution time, and escalation frequency help in identifying inefficiencies and improving workflow management. A structured complaint model is designed to define the lifecycle of a complaint, including stages such as submission, verification, assignment, processing, and resolution. This ensures consistency and transparency in complaint handling.

[3] Complaint Processing and System Implementation

The system is implemented using a role-based architecture to manage different types of users such as administrators, employees, and customers. Each role is assigned specific permissions to ensure secure and controlled access. A structured workflow is followed where complaints are created, verified, assigned to relevant departments, and tracked until resolution. Secure authentication mechanisms are used to validate users, and an approval process is implemented for employee registration to prevent unauthorized access. The system ensures efficient complaint handling by automating status updates and maintaining a complete history of actions performed on each complaint.

[4] Classification and Optimization Strategy

Complaints are categorized based on priority and status to improve management efficiency. They are classified into:

- Pending Complaints: Newly registered or unassigned complaints
- In-Progress Complaints: Complaints currently being handled
- Resolved Complaints: Successfully addressed complaints

This classification helps in identifying delays and bottlenecks in the system. Based on analysis, optimization strategies such as prioritization, reassignment, and escalation mechanisms are applied to improve resolution time. Recommendations are generated to enhance system efficiency, including workload balancing among employees and prioritizing high-urgency complaints.

[5] Dashboard Visualization and System Monitoring

To improve usability, the system includes an interactive dashboard that provides a clear overview of complaint statistics. The dashboard displays metrics such as total complaints, pending cases, resolved cases, and average

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

resolution time. Real-time notifications are integrated to keep users informed about status updates, assignments, and resolutions. This reduces communication gaps and improves responsiveness. A monitoring module allows administrators to analyze system performance and identify trends in complaint handling. The visualization components ensure transparency, support informed decision-making, and enhance overall system efficiency.

IV. SYSTEM ARCHITECTURE

The proposed system follows a multi-tier client–server architecture designed for efficient complaint management and tracking. The architecture consists of three main layers: (1) the web-based user interface, which acts as the presentation layer for complaint submission, tracking, and visualization; (2) the backend application server, which handles business logic, authentication, and workflow management; and (3) the database layer, which stores complaint records, user information, and system data securely.

Communication between the client and server is carried out using RESTful APIs with secure protocols, ensuring reliable and efficient data exchange. The workflow begins when a user submits a complaint through the web interface. The backend server processes the request, verifies user authenticity, and stores the complaint in the database. Based on predefined workflows, the complaint is assigned to the appropriate department or employee.

The system implements a structured complaint lifecycle, where each complaint progresses through stages such as submission, verification, assignment, processing, and resolution. Role-based access control ensures that only authorized users can view or modify specific data. Real-time notifications are generated to keep users informed about status updates and actions taken.

The architecture also supports multi-organization (multi-tenant) functionality, ensuring complete data isolation between organizations while allowing centralized management within a single platform.

For scalability and deployment, the backend can be containerized and hosted on cloud infrastructure, enabling the system to handle increasing numbers of users and complaints efficiently. The modular design ensures maintainability, flexibility, and adaptability to different organizational requirements.

V. RESULTS AND DISCUSSION

The proposed SmartComplaint system was evaluated using simulated and real-time complaint data collected from multiple organizational scenarios. The system performance was analyzed based on parameters such as complaint resolution time, system responsiveness, and workflow efficiency. The evaluation demonstrates that the system effectively manages and tracks complaints across different stages, ensuring transparency and accountability. The dashboard visualization presents an overview of complaint statistics such as total complaints, pending cases, and resolved cases, as shown in Fig. 1.

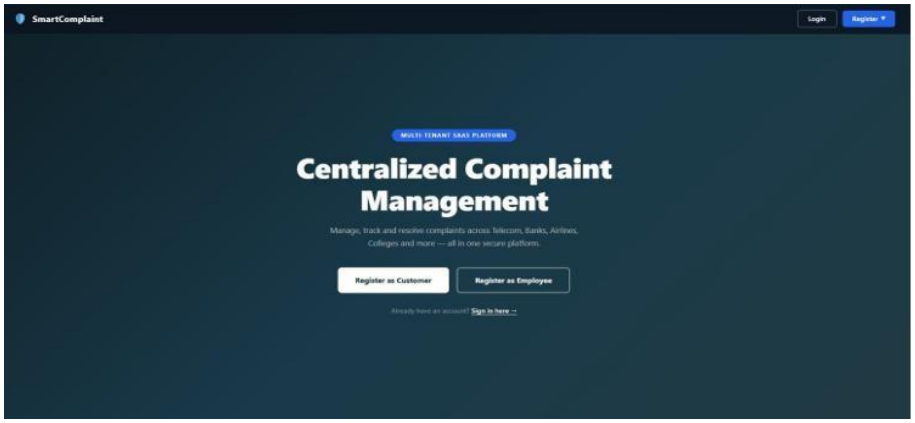

Figure 1 : Home Page

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

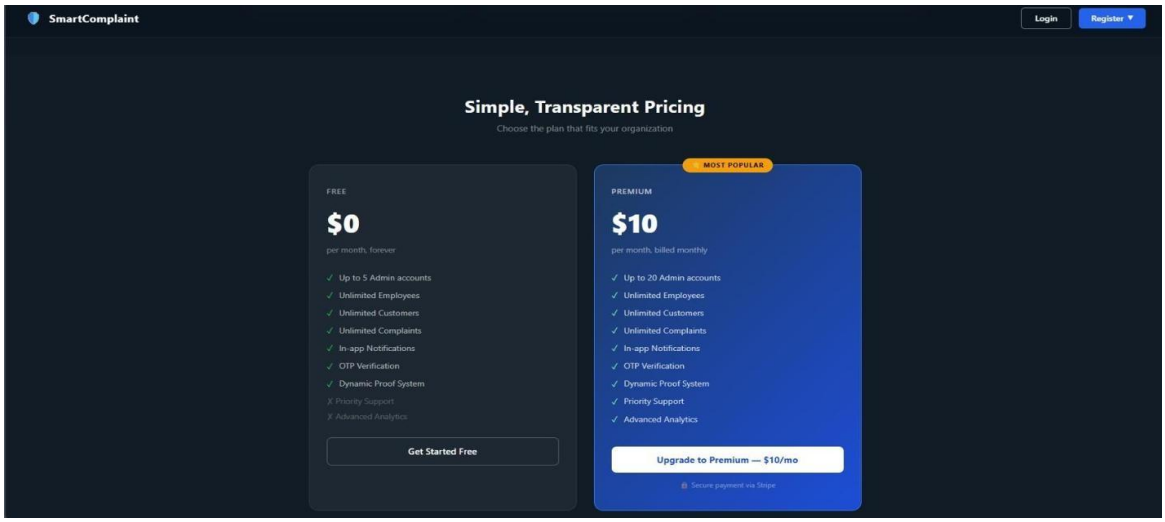

Figure 2 – Landing Page

The landing page provides an overview of the SmartComplaint platform with options for user registration and login. It highlights key features such as centralized complaint management, secure access, and multi-organization support. The pricing section enables organizations to choose suitable plans based on their requirements.

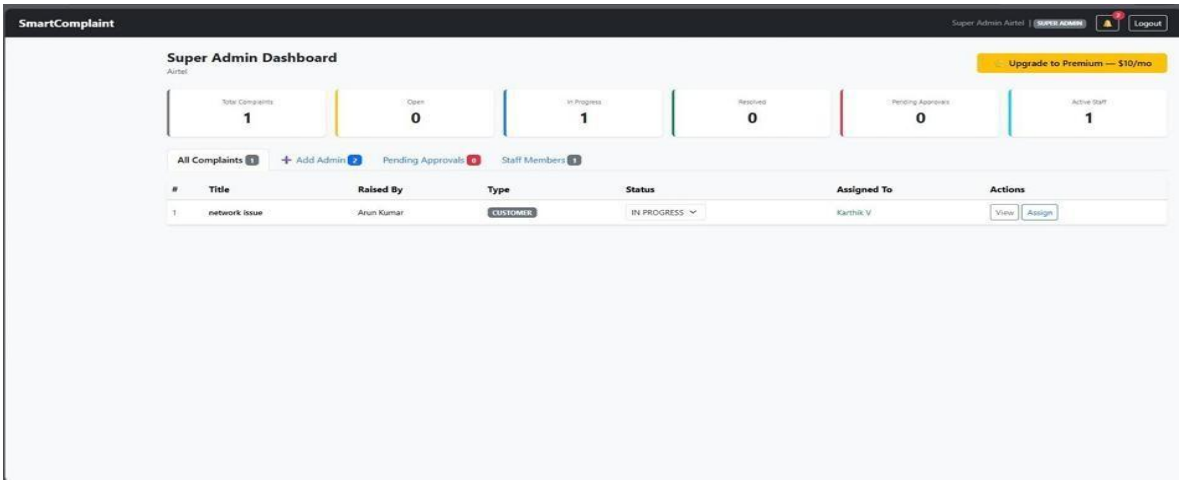

Figure 3 – Super Admin Dashboard

The Super Admin Dashboard allows complete control over the system, including managing organizations, monitoring complaints, and approving employees. It provides a high-level summary of system activities, enabling efficient administration and decision-making.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

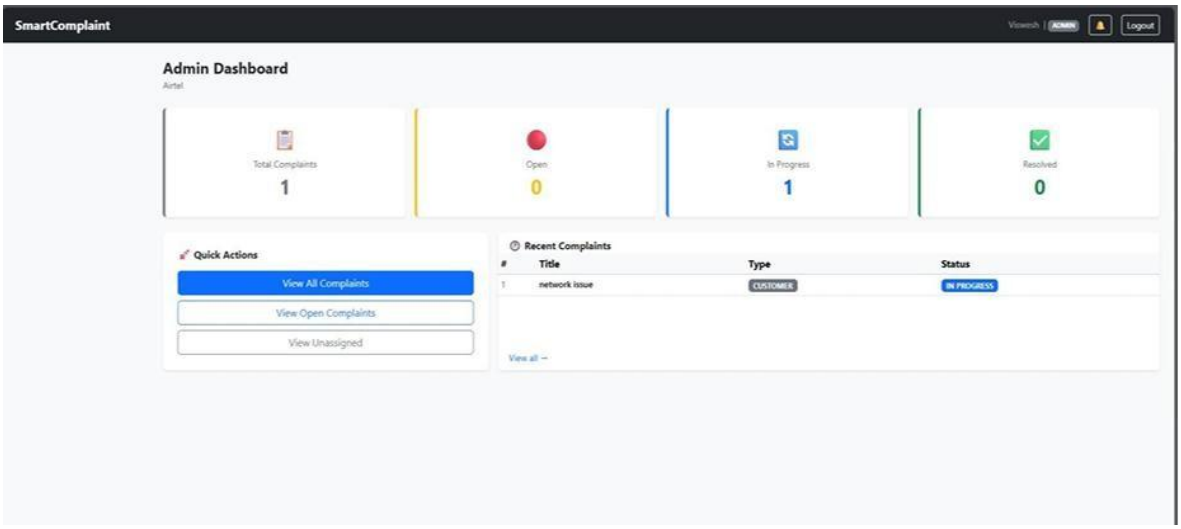

Figure 4 – Admin Dashboard

The Admin Dashboard provides insights into complaints within a specific organization. It displays key metrics such as total complaints, open issues, and resolved cases. Admins can assign complaints to employees and track progress effectively.

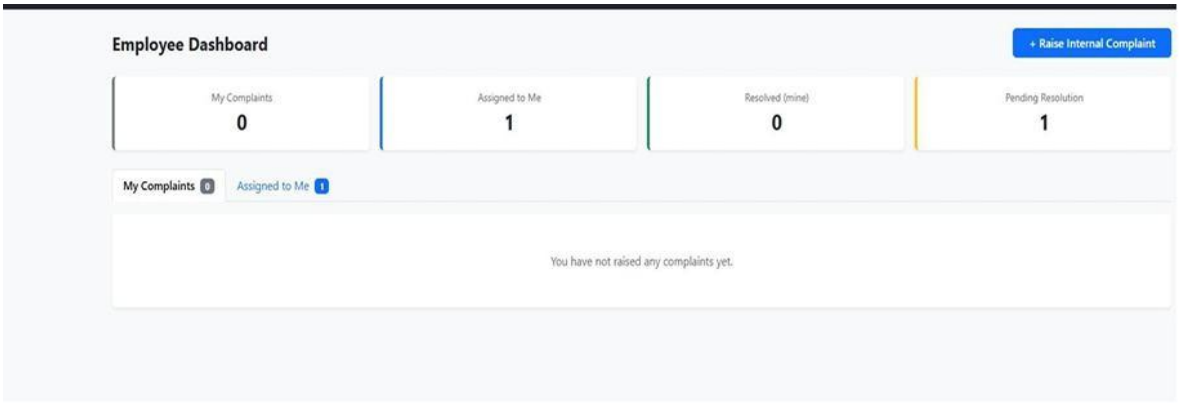

Figure 5 – Employee Dashboard

The Employee Dashboard helps employees manage assigned complaints. It shows the number of tasks assigned, resolved, and pending. Employees can update complaint status and ensure timely resolution.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

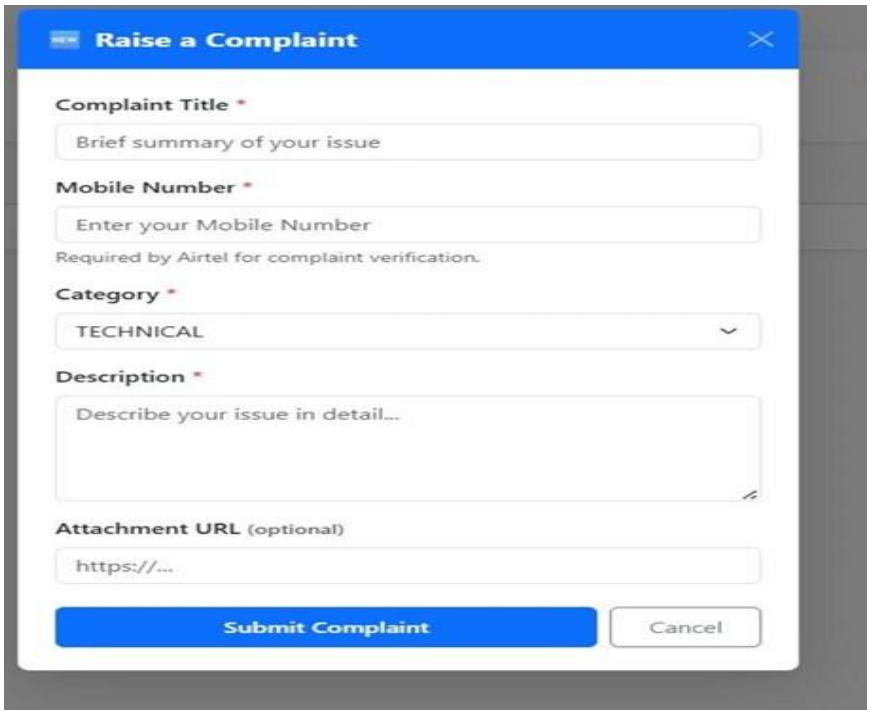

Figure 6 – Complaint Form

The complaint form allows users to submit complaints by entering details such as title, category, description, and contact information. This structured input ensures proper classification and efficient processing of complaints.

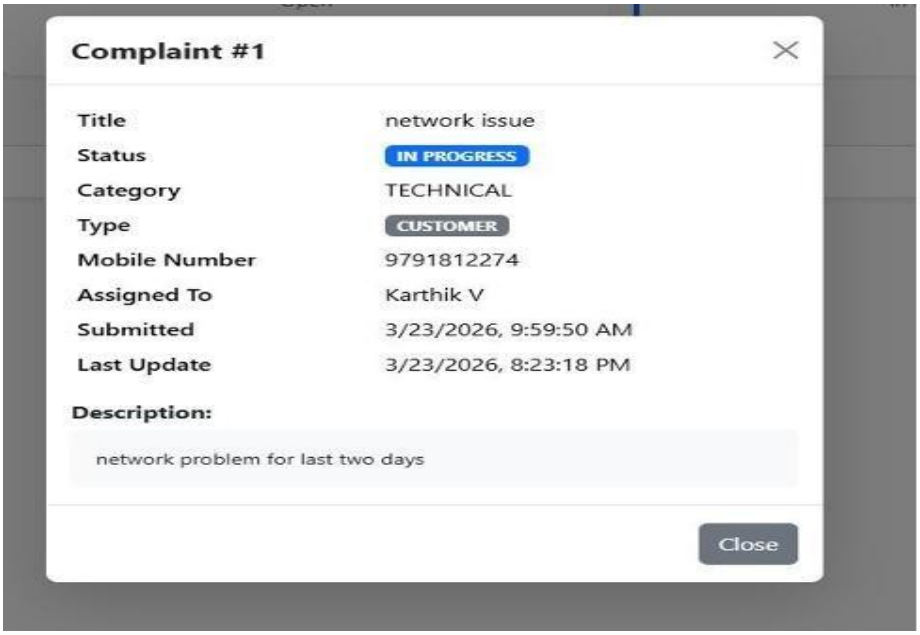

Figure 7 – Customer Dashboard and Complaint View

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 5, April 2026 Impact Factor: 8.2

This view displays detailed information about a specific complaint, including status, category, assigned employee, and timestamps. It helps users track the progress of their complaints and ensures transparency in the resolution process.

The above results demonstrate that the SmartComplaint system effectively streamlines complaint management through structured workflows, role-based access, and real-time tracking. The user-friendly interface and dashboards provide clear insights into complaint status and system performance. Overall, the system improves efficiency, reduces response time, and enhances user satisfaction.

ACKNOWLEDGMENT

The authors would like to thank Ms.M.Dharshini,M.E.,(CSE) , K.L.N. College of Engineering, for her continuous guidance and support throughout this project. The authors also acknowledge the use of Google Colab's GPU resources and the Roboflow platform for dataset curation and annotation management.

REFERENCES

[1] Bezemer, C. P., & Zaidman, A. (2010). Multi-Tenant SaaS Applications: Maintenance Dream or Nightmare? Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), pp. 88–92.

[2] Takabi, H., Joshi, J. B. D., & Ahn, G. J. (2010). Security and Privacy Challenges in Cloud Computing Environments.
IEEE Security & Privacy, 8(6), 24–31. https://doi.org/10.1109/MSP.2010.186

[3] Niranjanamurthy, M., & Chahar, D. (2013). The Study of E-Commerce and M Commerce Advantages, Limitations and Security. International Journal of Advanced Research in Computer and Communication Engineering, 2(6), 2360–2

[4] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role- BasedAccess Control https://doi.org/10.1109/2.485845 Models.IEEEComputer,29(2),38–47.

[5] Singh, A., & Wesson, J. (2009). Evaluation Criteria for Assessing the Usability of ERP Systems. Proceedings of the 2009 Annual Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT), pp. 87 95.

[6] Johnson, R., Hoeller, J., Donald, K., Sampaleanu, C., Harrop, R., Risberg, T., & Arendsen, A. (2022). Spring Framework Reference Documentation — Version 6.x. VMware / Spring.io.

[7] Stripe Inc. (2024). Stripe API Reference — PaymentIntents, Webhooks, and Stripe.js. https://stripe.com/docs/api/payment_intents and https://stripe.com/docs/js

[8] JSON Web Token (JWT). (2015). RFC 7519: JSON Web Token. Internet Engineering Task Force (IETF). https://datatracker.ietf.org/doc/html/rfc7519

[9] Facebook Inc. / Meta. (2024). React.js Documentation — Components, Hooks, and Routing. https://react.dev/ and https://reactrouter.com/