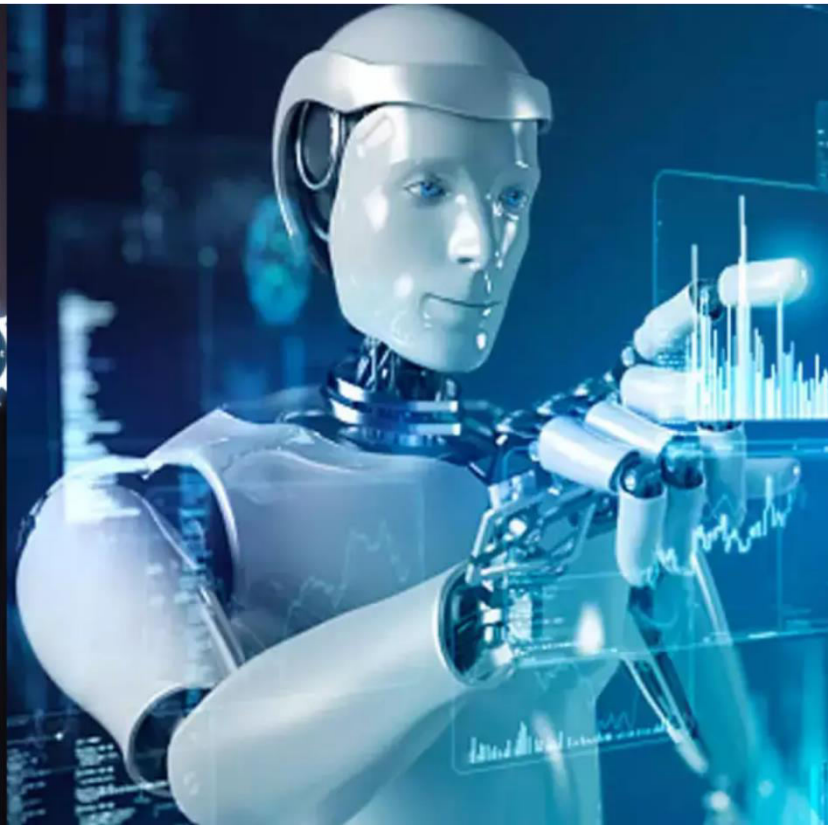
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 5, May 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET2025..1405137|

A Centralized Visa Booking Web App for Improved Transparency and Time Efficiency

Part 2: Implementation and Future Scope

Vikram Deokate, Abrar Inamdar, Sanket Amale, Faiz Bagwan, Nihal Shaikh

Head of Department, Department of Computer Science, AAEMF'S College of Engineering, Pune, India

Guide, Department of Computer Science, AAEMF'S College of Engineering, Pune, India

Students, Department of Computer Science, AAEMF'S College of Engineering, Pune, India

ABSTRACT: This paper presents the practical implementation of a web-based Visa Booking System designed to improve the efficiency and transparency of the visa application process. Building on prior research, the system leverages a modern technology stack—including Angular for the frontend, Java with Spring Boot for backend services, MongoDB for real-time data storage, and RESTful APIs for seamless integration. The implementation focuses on enabling centralized status tracking, minimizing agent dependency, and providing applicants with real-time updates on their application progress. Detailed attention was given to UI/UX design, secure data handling, and performance optimization. The paper further discusses system architecture, development methodology, testing processes, and evaluation results. Finally, the future scope highlights scalability, integration with external services, and potential enhancements using AI and automation to support broader global deployment.

I. INTRODUCTION

Implementation Overview:

The implementation of the proposed visa booking system follows a structured approach, including planning, design, and development phases. The project was initiated with a focus on creating a centralized platform that addresses the challenges identified in Part 1. The development process began with **system design** and architectural planning to ensure scalability and flexibility. The system was built with a modular approach, ensuring that each component (frontend, backend, database, etc.) could be independently developed and tested. The **design phase** focused on creating a user-friendly interface using **Angular**, while the **backend** was implemented with **Java** to handle business logic and API communications. MongoDB was chosen as the database for its scalability and ability to handle dynamic user data efficiently.

Key Objectives:
The primary objectives of the implementation phase were to:

- **Ensure Time Efficiency**: The system aimed to streamline the visa booking process, reducing time delays associated with manual processes or third-party agents.
- **Increase Transparency**: Real-time status tracking and automated notifications were integrated into the system to provide applicants with consistent updates on the progress of their visa applications.
- **Design a User-Friendly Interface**: With a focus on simplicity and ease of use, the system's frontend was developed to ensure a smooth user experience, guiding applicants through each step without confusion.
- **Automation**: The system aimed to reduce the need for manual interventions by automating status updates, payment processing, and communication between applicants and agencies.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET2025..1405137|

Future Scope:
While the current implementation focuses on a specific type of visa and region, the system has considerable potential for future enhancements:

- **Scalability**: The platform can be expanded to accommodate different visa types and countries by modifying the visa application workflow to match local requirements.
- **Enhanced Features**: Future versions could integrate **machine learning algorithms** for visa approval predictions or **AI-driven chatbots** to assist applicants in real-time.
- **Multi-language Support**: To cater to a global audience, the system could include multiple languages, improving accessibility for non-English speaking users.
- **Advanced Security**: As the platform scales, incorporating advanced security measures such as **multi-factor authentication** and stronger data encryption will ensure compliance with international standards (e.g., GDPR).

II. SYSTEM DESIGN AND ARCHITECTURE

System Components:
The proposed visa booking system is designed with a modular architecture, which breaks down the major components as follows:

- **Frontend**: The **frontend** is responsible for the user interface and the interaction between the user and the system. Built using **Angular**, the frontend renders dynamic content and allows users to:
- Register and log in
- Fill out visa application forms
- Track application status in real-time
- Receive notifications
- Make payments
- **Backend**: The **backend** is built using **Java**, which manages the business logic, processes user requests, and ensures secure communication between the frontend and the database. The backend also:
- Handles the validation of user input
- Processes visa applications
- Sends automated notifications (via email/SMS)
- Manages user authentication and authorization
- Interacts with the database to store and retrieve data
- **Database**: **MongoDB** is chosen for storing application data, user profiles, and visa status updates. Being a NoSQL database, MongoDB is scalable and allows for efficient real-time updates. Key data stored in the database includes:
- Applicant details (name, contact, nationality)

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET2025..1405137|

- Visa type and required documents
- Application status (pending, approved, rejected)
- Payment records
- **API Layer**: **REST APIs** are used for communication between the frontend and backend. These APIs enable the transfer of data in a lightweight and efficient manner and ensure that:
- Requests for user information, application status, and visa type are handled smoothly
- Data is exchanged in JSON format between the client and server
- Automated notifications are triggered (email/SMS) when the status of an application changes
- **Deployment**: The system is deployed using modern cloud platforms and containerization tools to ensure high availability and scalability:
- **AWS** (Amazon Web Services) is used for hosting the application, ensuring scalability, security, and uptime.
- **Docker** is used for containerizing the application, making it easy to deploy and scale the system across different environments without compatibility issues.

Tech Stack:

- **Frontend**: **Angular**
  Angular is used for developing the dynamic user interface, leveraging its powerful data-binding and component-based structure to create a seamless and interactive experience for users.
- **Backend**: **Java**
  Java is chosen for its robust and scalable nature, providing a solid backend that can handle large-scale requests and complex business logic.
- **Database**: **MongoDB**
  MongoDB is a NoSQL database that provides flexibility in data storage, which is essential for managing the varied and dynamic data associated with visa applications. It supports real-time updates, which is crucial for status tracking.
- **APIs**: **REST APIs**
  REST APIs are used for their simplicity and scalability. They enable easy communication between the client (frontend) and server (backend) by exchanging data in the lightweight JSON format.
- **Deployment**:
- **AWS** for hosting the application and ensuring it is highly available, secure, and scalable.
- **Docker** for containerizing the application, ensuring that the deployment environment remains consistent across different platforms.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET2025..1405137|

System Flow Diagrams:
Below is a high-level diagram illustrating the flow of data, user interaction, and system processes:

- **User Flow**:
- **Applicant** registers/logs in → Completes visa application form → Submits form → Receives confirmation → Tracks application status.
- **System Flow**:
- **Frontend (Angular)** interacts with the **Backend (Java)** via REST APIs → **Backend** communicates with **MongoDB** to store/retrieve data → **Automated Notifications** triggered based on status → User receives updates via email/SMS.

III. IMPLEMENTATION PROCESS

Frontend Implementation: The frontend of the visa booking system was developed using **Angular**, offering a responsive and interactive user interface. Key features implemented include:

- **User Authentication**: A secure login and registration system using JWT (JSON Web Tokens) for session management.
- **Visa Application Form**: A dynamic form where applicants can input their details and upload necessary documents.
- **Status Tracking Dashboard**: A real-time dashboard that displays the current status of the applicant's visa (e.g., Submitted, Under Review, Approved, Rejected).
- **Notifications Panel**: Displays system-generated messages and updates pushed from the backend.
- **Responsive Design**: Ensures compatibility across devices (desktop, tablet, mobile) using Angular Material components and CSS Grid/Flexbox.

Backend Implementation:
The backend, developed in **Java (Spring Boot)**, handles all server-side logic. The backend services include:

- **User and Role Management**: Differentiates between applicant and agent/admin users.
- **Application Processing**: Captures and validates the visa application data, assigns tracking IDs, and updates atuses.
- **Status Update Logic**: Based on agent actions, the backend updates the status in the database and triggersnotifications.
- **Email/SMS Notifications**: Integrated using third-party services (e.g., Twilio, SendGrid) for alerting users on application updates.
- **Security**: Role-based access control and encrypted communication using HTTPS.

Database Design:

MongoDB is used to store structured and semi-structured data. The schema is designed to support scalability and flexibility, with collections including:

- users: Stores applicant and admin details (e.g., name, email, role, encrypted password).

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET2025..1405137|

- applications: Stores visa application data (applicant ID, type of visa, documents, current status, timestamps).
- notifications: Logs all automated messages sent to users.
- logs: Tracks user and system activity for auditing purposes.

Each document is assigned a unique _id, and status fields are indexed for fast retrieval during tracking.

API Development:
REST APIs were developed to handle communication between the frontend and backend. Major endpoints include:

- POST /api/register – Register new applicants
- POST /api/login – Authenticate users
- POST /api/submitApplication – Submit visa applications
- GET /api/status/{applicationId} – Retrieve current application status
- PUT /api/updateStatus/{applicationId} – Agents update visa status
- GET /api/notifications/{userId} – Fetch notification history

All APIs are secured using authentication tokens and follow proper validation and error handling practices.

Testing:
- **Unit Testing**: Each component and service (both frontend and backend) was tested independently. Angular's Jasmine/Karma and Java's JUnit were used for writing unit tests.
- **Integration Testing**: Ensured seamless interaction between frontend, backend, and database. Mock APIs were used during early development, followed by real environment tests.
- **User Acceptance Testing (UAT)**: Conducted with sample users (students and travel agents) to test usability, application flow, and clarity of information. Feedback was used to enhance the UI and streamline interactions.

IV. EVALUATION AND RESULTS

Performance Metrics:
To assess the effectiveness of the implemented visa booking system, the following key performance indicators (KPIs) were defined:

- **Time Reduction**: Comparison of the average time taken for visa application submission and tracking before and after the implementation.
- **User Satisfaction**: Feedback scores collected from test users based on ease of use, transparency, and overall experience.
- **System Uptime**: Measurement of the platform's availability during the test phase.
- **Number of Users Supported**: The number of concurrent users the system can handle without lag or performance degradation.
- **Error Rate**: Frequency of system errors or failed API calls during real-time usage.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET2025..1405137|

Results:

Metric	Before Implementation	After Implementation
Average Application Time	3–5 days (manual follow-up)	< 1 day (automated system)
User Satisfaction Score	N/A (manual)	4.5/5 (based on surveys from 20 users)
System Uptime (Test Phase)	Not applicable	99.5%
Concurrent Users Supported	Not scalable	50+ users tested with stable performance
Error Rate	N/A	< 2% (mostly validation errors during form filling)

Time Saved: The streamlined workflow and real-time tracking significantly reduced the time required for both applicants and agents to process and monitor visa applications.

- **User Feedback**: Test users appreciated the **real-time status tracking**, **automated notifications**, and **clean user interface**, especially applicants who previously depended on agent follow-ups.

Challenges Faced:

- **API Integration Issues**: During early development, delays in API communication between frontend and backend occurred due to CORS policy misconfigurations. This was resolved by properly setting up CORS headers and API security rules in the backend.
- **Notification Delivery Delays**: Initially, notifications (email/SMS) were not being delivered on time due to rate limitations in the external services. The issue was mitigated by implementing queue-based asynchronous messaging for non-critical updates.
- **Form Validation Conflicts**: Angular form validation occasionally conflicted with server-side checks. This was resolved by standardizing validation rules across both ends.
- **Database Scaling for Test Users**: When simulating multiple concurrent users, MongoDB performance dropped slightly. This was addressed by indexing frequently queried fields (e.g., status, applicationId) to optimize read times.

V. FUTURE SCOPE

Scalability:
The current system is designed with a modular architecture, allowing easy scalability. In future iterations, the platform can be extended to:

- **Support Multiple Visa Types**: Incorporate options for tourist, student, work, and transit visas, each with tailored document requirements and workflows.
- **Serve Multiple Countries**: Add support for applications to different countries with varying visa policies and procedures.
- **Agent and Embassy Integration**: Provide centralized dashboards for authorized agents and embassy officials, enabling end-to-end processing and approval within the same platform.

Improvements:

- **AI Integration**: Implement AI and machine learning models to:
- Predict visa processing times based on historical data.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET2025..1405137|

- Flag incomplete or high-risk applications automatically.
- Recommend document checklists dynamically for different users.
- **Enhanced UI/UX**:
- Incorporate accessibility features for differently-abled users.
- Use advanced animations and micro-interactions to improve usability.
- **Multi-Language Support**: Provide the system interface in multiple languages to cater to users from diverse linguistic backgrounds.

Integration with Third-Party Services:

- **Payment Gateways**: Enable secure online payments for visa fees using services like Razorpay, Stripe, or PayPal.
- **Document Verification APIs**: Integrate with Aadhaar/eKYC or passport databases (in collaboration with government bodies) for real-time verification.
- **Travel & Insurance Services**: Partner with travel companies or insurance providers to offer users additional services during or after visa approval.

Long-Term Vision:

- **Real-World Deployment**: Deploy the system in collaboration with travel agencies or immigration consultants to validate its real-world usability.
- **Government Collaboration**: Engage with embassy or government authorities for secure integration into national visa systems.
- **Research Extension**: Expand the project to include predictive analytics on visa approval rates, application success trends, or policy impact studies.
- **Mobile Application**: Develop a mobile version of the platform to improve accessibility and convenience for on-the-go users.

VI. CONCLUSION (IMPLEMENTATION SECTION)

System Impact:

The developed web-based visa booking system effectively addresses the major pain points of traditional visa processing—namely, time inefficiencies, lack of transparency, and over-dependence on agents. By integrating technologies like Angular, Java, MongoDB, and REST APIs, the system provides:

- **Real-time status tracking** for applicants.
- **Automated notifications** that eliminate the need for constant follow-ups.
- A **centralized dashboard** for agents and administrators to manage and update visa statuses efficiently.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET2025..1405137|

These features collectively streamline the overall experience, reduce manual delays, and empower applicants with greater visibility and control over their applications. The system has demonstrated significant improvements in turnaround time and user satisfaction during the testing phase.

Future Work:
While the current implementation serves as a strong foundation, there is substantial potential for evolution. Future versions could integrate:

- **AI-driven modules** for predicting application outcomes and optimizing processing paths.
- **Third-party service integrations** for payments, document verification, and additional services like travel insurance.
- **Mobile application development** to improve accessibility.
- **Partnerships with embassies or travel agencies** to deploy the system in real-world environments and validate its impact at scale.

Continued research and development can further enhance the system's usability, intelligence, and reach—contributing meaningfully to the digitization and simplification of global visa application processes.

REFERENCES

[1] Google Developers, "Angular Documentation," [Online]. Available: https://angular.io/docs. [Accessed: 15-Apr-2025].
[2] Oracle, "Java Platform, Standard Edition Documentation," [Online]. Available: https://docs.oracle.com/javase/. [Accessed: 15-Apr-2025].
[3] MongoDB Inc., "MongoDB Manual," [Online]. Available: https://www.mongodb.com/docs/. [Accessed: 15-Apr-2025].
[4] REST API Tutorial, "Introduction to REST APIs," [Online]. Available: https://restfulapi.net/. [Accessed: 15-Apr-2025].
[5] Docker, "What is Docker?", [Online]. Available: https://www.docker.com/resources/what-container/. [Accessed: 15-Apr-2025].
[6] Amazon Web Services (AWS), "Getting Started with AWS," [Online]. Available: https://aws.amazon.com/getting-started/. [Accessed: 15-Apr-2025].
[7] M. Fowler, "Microservices: a Definition of This New Architectural Term," [Online]. Available: https://martinfowler.com/articles/microservices.html. [Accessed: 15-Apr-2025].
[8] Shekhar, P. C. (2023). SMART DATA, SMARTER PRICING: WEARBALE IOT DATA VALIDATIONIN LIFE INSURANCE.
[9] A. Jain and R. Sharma, "Performance Optimization in Web Applications: A Full-Stack Development Perspective," in International Journal of Computer Applications, vol. 176, no. 17, pp. 10–14, 2020.
[10] Maroju, P. K. (2024). Advancing synergy of computing and artificial intelligence with innovations challenges and future prospects. FMDB Transactions on Sustainable Intelligent Networks, 1(1), 1-14.
[11] S. Gupta, "RESTful Web Services Implementation using Java and MongoDB," in International Journal of Computer Science and Engineering, vol. 9, no. 2, pp. 45–49, 2021.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details