NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

REPORT ON

15 WEEKS OF INTERNSHIP

Carried out at

CLOVER BAY TECHNOLOGIES

Submitted to

NMAM INSTITUTE OF TECHNOLOGY, NITTE

(An Autonomous Institution under VTU, Belagavi)

In partial fulfilment of the requirements for the award of the

Degree of Bachelor of Engineering in
Information Science & Engineering

by

GAURAV SHENOY K
4NM21IS216

Under the guidance of

Mr. Manjunath BS
(CTO)

NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

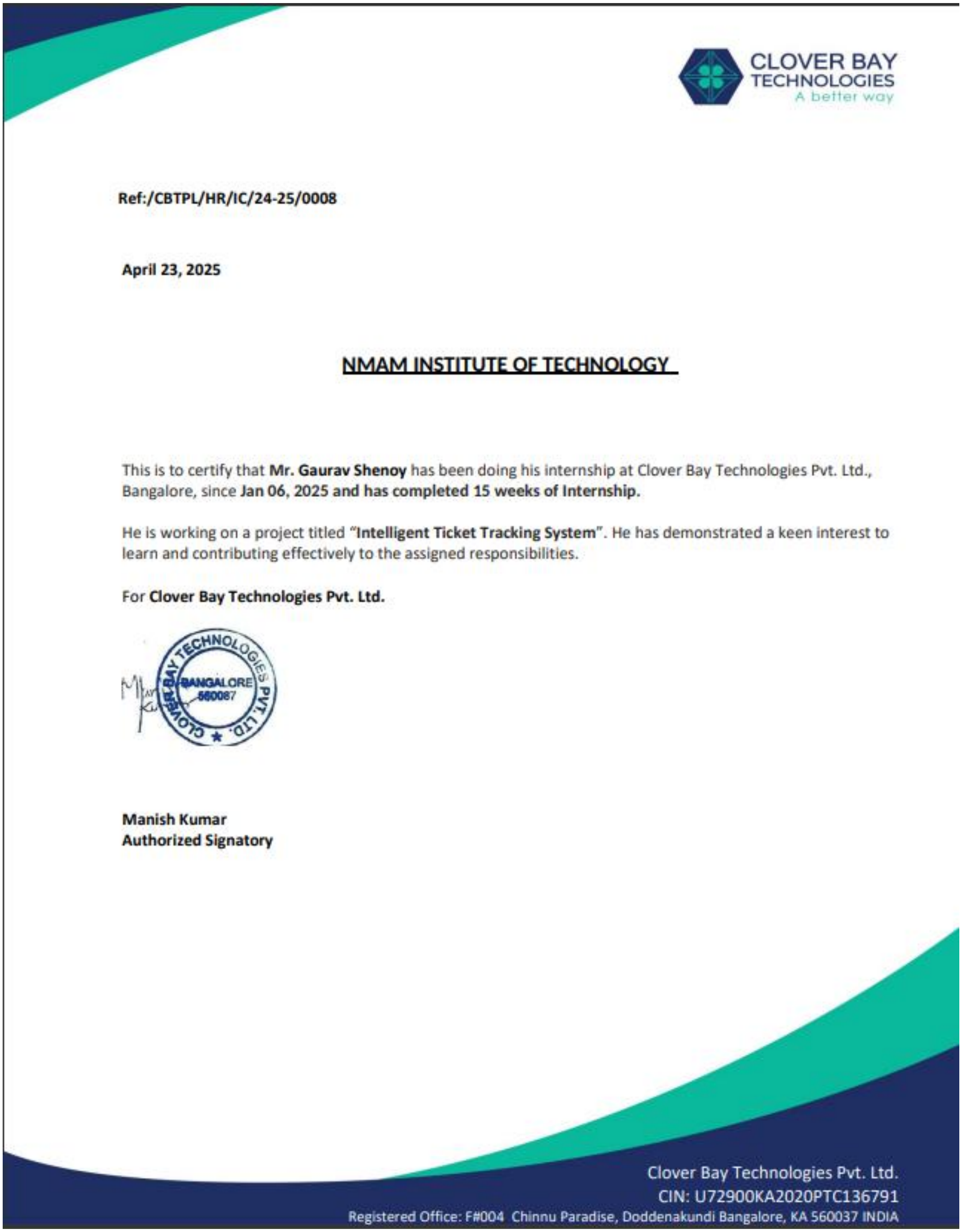
CERTIFICATE

This is to certify that the "Internship report" submitted by Gaurav Shenoy K USN 4NM21IS216 *of VIII semester B.E., a Bonafide student of NMAM Institute of Technology, Nitte, has undergone 15 weeks of internship at Clover Bay technologies during January 6 2025 to April 19 2025 fulfilling the requirements for the award of degree of Bachelor of Engineering in* ***Information Science & Engineering*** *at NMAM Institute of Technology, Nitte*.

Name and Signature of Mentor

Signature of HOD

CERTIFICATE



CLOVER BAY
TECHNOLOGIES
A better way

Ref:/CBTPL/HR/IC/24-25/0008

April 23, 2025

NMAM INSTITUTE OF TECHNOLOGY

This is to certify that **Mr. Gaurav Shenoy** has been doing his internship at Clover Bay Technologies Pvt. Ltd., Bangalore, since **Jan 06, 2025 and has completed 15 weeks of Internship.**

He is working on a project titled "**Intelligent Ticket Tracking System**". He has demonstrated a keen interest to learn and contributing effectively to the assigned responsibilities.

For **Clover Bay Technologies Pvt. Ltd.**

CLOVER BAY TECHNOLOGIES PVT. LTD.
BANGALORE
560087

Manish Kumar
Authorized Signatory

Clover Bay Technologies Pvt. Ltd.
CIN: U72900KA2020PTC136791
Registered Office: F#004 Chinnu Paradise, Doddenakundi Bangalore, KA 560037 INDIA

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Clover Bay Technologies for providing me with the opportunity to pursue my internship and gain practical exposure to the industry. I am especially thankful to Mr. Manjunath B S, Chief Technology Officer of Clover Bay Technologies, for his constant guidance, valuable insights, and encouragement throughout the internship period.

I also extend my heartfelt thanks to the entire team at Clover Bay Technologies for their support, cooperation, and for creating a positive learning environment. Their assistance and feedback have been instrumental in the successful completion of this project.

I also extend my sincere thanks to Dr. Ashwini B, Head of the Department of Information Science & Engineering, NMAMIT, for her support and for overseeing the academic aspects of my internship.

My gratitude also goes to Dr. Niranjan N. Chiplunkar, Principal of NMAM Institute of Technology, Nitte, for providing an environment that fosters learning and growth.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The Ticketing System is a web-based application designed to simplify and automate the process of managing customer queries, technical issues, and service requests within an organization. During my internship, I developed this project to address the common challenges faced by support teams, such as scattered communication, lack of centralized tracking, and delayed resolution of reported problems.

This system enables users to create tickets by submitting their issues through an intuitive frontend interface. Each ticket is assigned a unique identifier and can be tracked throughout its lifecycle, from creation to resolution. Administrators can view, assign, update, and close tickets, ensuring clear task ownership and accountability at every stage. The system also provides real-time status updates to users, helping reduce follow-up efforts and improving user satisfaction.

The project adopts a full-stack development approach, utilizing React and TypeScript for the frontend to ensure a smooth and responsive user experience, while Java Spring Boot powers the backend services, handling business logic, API routing, and secure data persistence. This separation of concerns ensures scalability and maintainability of the codebase.

Through the implementation of this Ticketing System, organizations can streamline their support processes, minimize communication gaps, and maintain a historical record of all raised and resolved issues for future reference. The system is designed with flexibility in mind, allowing easy adaptation to various business domains beyond technical support, such as human resources, facility management, and client service desks.

The Ticketing System ultimately enhances organizational efficiency by ensuring a structured and transparent workflow for issue tracking and resolution.

Chapter 1: Introduction

Clover Bay Technologies, is a rapidly growing digital transformation company focused on the retail, e-commerce, and supply chain sectors. The firm partners with businesses to modernize operations through AI-driven, cloud-based, and agile solutions tailored to client needs. Known for its customer-centric approach, Clover Bay builds scalable platforms that improve both operational efficiency and user experience. One of its notable achievements includes developing India’s second-largest e-commerce platform from the ground up. With a strong emphasis on engineering excellence and strategic vision, the company positions itself as a trusted technology partner for digital innovation.

The Ticketing System developed during this internship addresses common challenges faced by support teams, such as scattered communication, lack of centralized tracking, and delayed resolution of issues. It enables users to create tickets through an intuitive frontend interface, with each ticket assigned a unique identifier for tracking throughout its lifecycle. Administrators can view, assign, update, and close tickets, ensuring clear task ownership and accountability at every stage. Real-time status updates and notifications keep users informed, reducing follow-up efforts and improving user satisfaction.

The system adopts a full-stack development approach, utilizing React and TypeScript for the frontend to ensure a smooth and responsive user experience, while Java Spring Boot powers the backend services, handling business logic, API routing, and secure data persistence. This separation of concerns ensures scalability and maintainability of the codebase.

Through the implementation of this Ticketing System, organizations can streamline their support processes, minimize communication gaps, and maintain a historical record of all raised and resolved issues for future reference. The system is designed with flexibility in mind, allowing easy adaptation to various business domains beyond technical support, such as human resources, facility management, and client service desks. Ultimately, the Ticketing System enhances organizational efficiency by ensuring a structured and transparent workflow for issue tracking and resolution.

Chapter 2: Details Of the Training Undergone

2.1 TRAINING PROCESS UNDERGONE

Over the 15-week internship, Week 1 to Week 5 I underwent a comprehensive learning experience beginning with foundational corporate training. I gained insights into information security management, the company's code of conduct, leadership, and key software domains. I engaged in self-learning on Java Spring Boot, design patterns, JavaScript, and business continuity practices. Initial weeks also involved knowledge transfers and project discussions, including domain assignments and architecture planning for a ticketing application.

From Week 6 to Week 15, As your training progressed, I transitioned into practical development tasks. I contributed to designing the schema, building services, and developing both backend and frontend components of a ticketing system. I worked on application security using Key cloak and conducted integration of frontend and backend systems. In the final phase, you focused on testing using tools like Sonar, Jacoco, and Check style, ensuring quality and integrity of the application, rounding off a full-stack development experience.

2.2 PROBLEM STATEMENT

Organizations often face difficulties in managing customer queries and internal issues due to the lack of a centralized and structured system. Manual handling of support requests through emails or direct communication leads to delays, missed issues, lack of accountability, and poor tracking. This creates inefficiencies in resolving problems and maintaining service quality. A reliable ticketing system is essential to streamline issue reporting, automate status tracking, and ensure timely resolution of all raised requests.

2.3 OBJECTIVES

1. Ticket Lifecycle Management:
   Design and implement a platform to streamline ticket creation, assignment, status updates, and closure, ensuring clear tracking throughout the lifecycle.

2. Role and Privilege-Based Access Control:
   Implement secure role-based access to ensure that users, support agents, and administrators can only perform actions permitted for their assigned roles.

3. Multi-Tenant Support:
   Architect the system to support multiple tenants, enabling independent ticket data and configurations for different organizations or departments.

4. Real-Time Status Updates and Notifications:
   Ensure real-time updates for ticket status changes and trigger email notifications to inform users and agents promptly about important actions.

5. User-Friendly Interface:
   Provide a clean, responsive, and intuitive user interface that enhances user experience across desktops, tablets, and mobile devices.

6. Audit Trail and Analytics:
   Maintain historical records of all ticket activities and enable performance monitoring to identify trends, bottlenecks, and areas for process improvement.

2.4 METHODOLOGY

The development of the Ticketing System followed a structured and step-by-step approach to ensure effective handling of user-reported issues. Each stage in the workflow was carefully designed to support smooth ticket creation, assignment, and resolution while maintaining system transparency and user satisfaction. The methodology below outlines the core functional flow of the system.

1. **Issue Reporting:** The process begins when users encounter a technical issue or service requirement. They can report the issue either by sending an email to the IT support team or by using the in-built ticket creation form within the application. This ensures that every request is properly logged from the start and initiates a traceable workflow.

2. **Ticket Generation:** Once an issue is reported, the system automatically generates a support ticket. This ticket includes key information such as a brief description of the problem, the ticket category (e.g., software, hardware, access), ticket type (incident, request), urgency level, and a timestamp. These details enable structured problem handling and help prioritize critical issues.

3. **Ticket Classification:** The submitted ticket is validated to ensure all mandatory fields are completed correctly. Based on the issue category, the system classifies the ticket, which facilitates efficient routing to the appropriate support team or personnel. Classification also helps in analysing the nature and frequency of reported issues for future insights.

4. **Ticket Assignment:** After classification, the system either automatically or manually assigns the ticket to a specific support agent or team. This assignment is based on pre-defined rules, agent availability, and expertise area, ensuring the right personnel handle the issue for quicker resolution.

5. **Notification Dispatch:** As soon as a ticket is assigned, the respective support agent receives a real-time notification through email or in-app alerts. This ensures prompt acknowledgment and minimizes delays in addressing the issue. The requester may also receive confirmation regarding the ticket assignment and expected timelines.

6. **Response and Status Update:** The assigned support agent reviews the ticket details, interacts with the user if further clarification is needed, and works towards resolving the issue. Throughout the process, they can update the ticket status to reflect progress (e.g., Open, In Progress, Resolved, Closed). This transparent workflow keeps users informed at all times.

7. **Role-Based Access Management:** To ensure data privacy and secure operations, the system employs role-based access controls. Admin users have full control over managing user roles and defining who can view, create, update, or delete tickets. This structure prevents unauthorized access and ensures that each user interacts only with the features permitted for their role.

2.5 IMPLEMENTATION

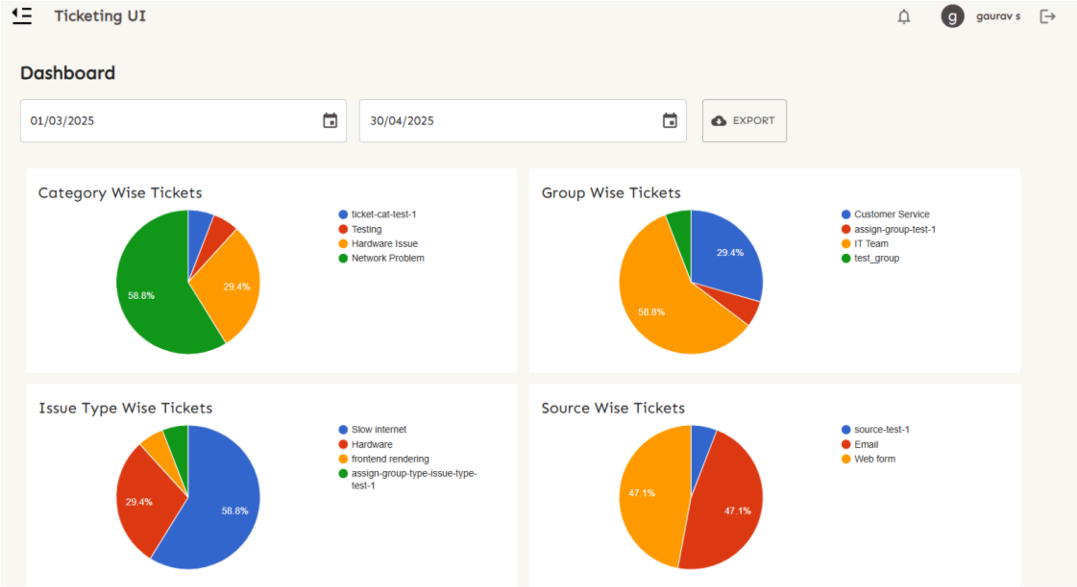


Fig 2.1. Dashboard for It support to see the Ticket Details

This ticketing dashboard provides a visual summary of support tickets within a selected date range. It includes four pie charts showing ticket distribution by category, support group, issue type, and source. Users can filter dates and export reports easily. The dashboard helps identify common issues, team workloads, and ticket origins. Its clean layout enables quick insights for efficient support management.

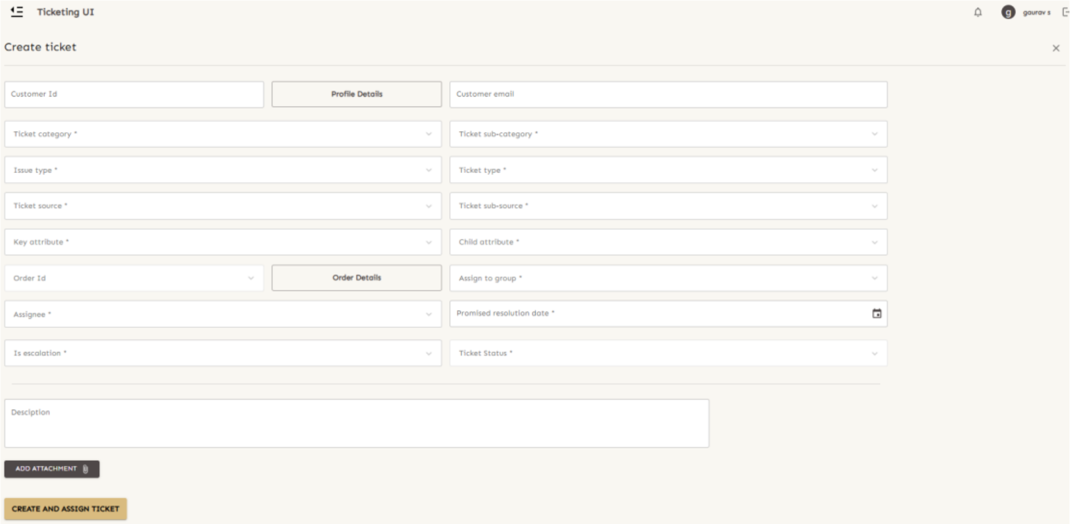
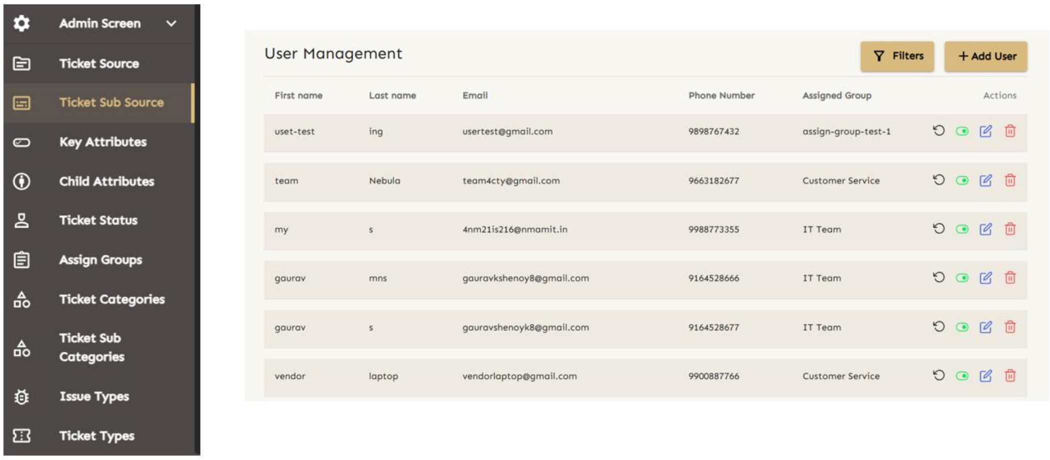


Fig 2.2. Ticket creation and Assign Ticket

This is a “Create Ticket” interface from a Ticketing UI system. It includes fields for customer details, ticket classification, issue type, and source information. Users can assign the ticket to groups or individuals and set a resolution date. There’s an option to add order details and mark if the issue is escalated. A description box and attachment option are provided before submitting the ticket.

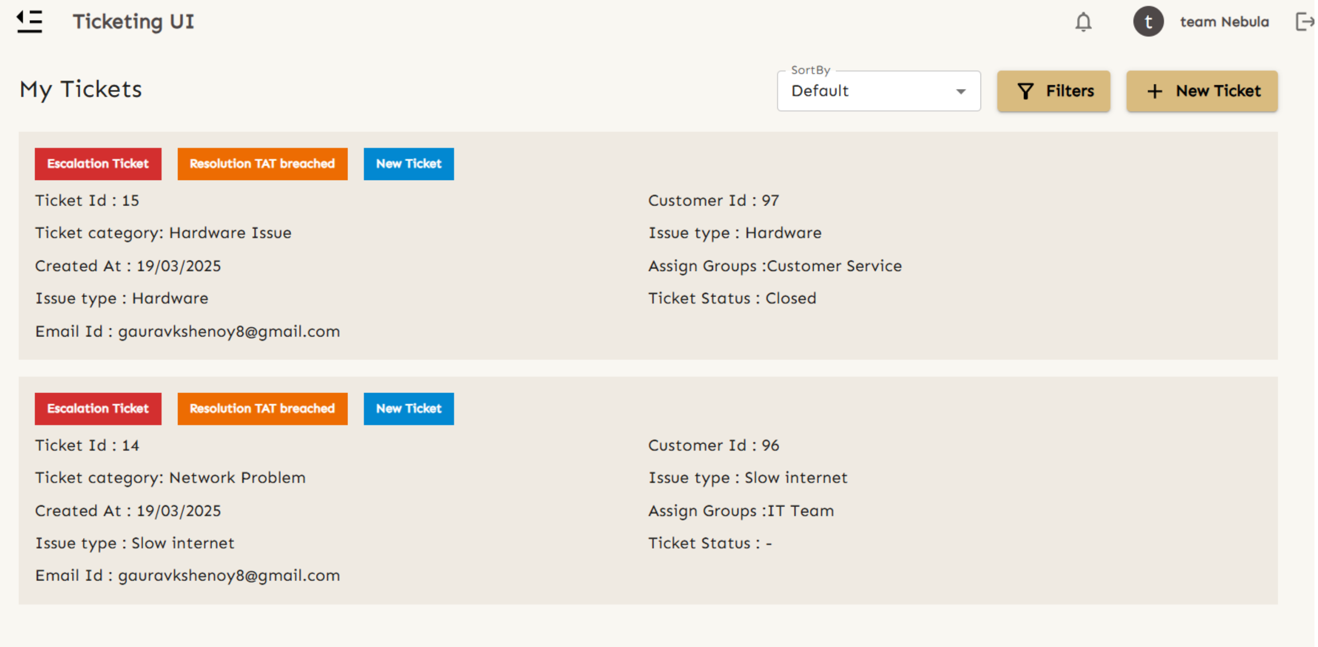


User Management

First name	Last name	Email	Phone Number	Assigned Group	Actions
uset-test	ing	usertest@gmail.com	9898767432	assign-group-test-1	↺ ◉ ✎ 🗑
team	Nebula	team4cty@gmail.com	9663182677	Customer Service	↺ ◉ ✎ 🗑
my	s	4nm21is216@nmamit.in	9988773355	IT Team	↺ ◉ ✎ 🗑
gaurav	mns	gauravkshenoy8@gmail.com	9164528666	IT Team	↺ ◉ ✎ 🗑
gaurav	s	gauravshenoyk8@gmail.com	9164528677	IT Team	↺ ◉ ✎ 🗑
vendor	laptop	vendorlaptop@gmail.com	9900887766	Customer Service	↺ ◉ ✎ 🗑

Fig 2.3. Admin Screen Today New Properties to the ticket and User Management

This image shows a “User Management” interface from an admin dashboard. The left panel features navigation options like Ticket Source, Ticket Status, and Issue Types. The main panel lists users with details such as name, email, phone number, and assigned group. Each user row includes action icons for editing, deleting, and other operations. Buttons for filtering users and adding new users are located at the top right of the main panel.



My Tickets

Escalation Ticket | Resolution TAT breached | New Ticket

Ticket Id : 15
Ticket category: Hardware Issue
Created At : 19/03/2025
Issue type : Hardware
Email Id : gauravkshenoy8@gmail.com

Customer Id : 97
Issue type : Hardware
Assign Groups :Customer Service
Ticket Status : Closed

Escalation Ticket | Resolution TAT breached | New Ticket

Ticket Id : 14
Ticket category: Network Problem
Created At : 19/03/2025
Issue type : Slow internet
Email Id : gauravkshenoy8@gmail.com

Customer Id : 96
Issue type : Slow internet
Assign Groups :IT Team
Ticket Status : -

Fig 2.4. Assigned User screen

The image shows a “Ticketing UI” interface listing support tickets. Each ticket displays its ID, category, issue type, customer details, assigned group, and status. Tags like “Escalation Ticket” and “Resolution TAT breached” indicate urgent issues. The tickets can be sorted and filtered, with an option to create a new ticket. The user interface is simple and neatly organized for quick access to ticket information.

2.6 RESULT

The implementation of the Ticket Management System has successfully addressed the core objectives outlined in the project scope. A robust platform was developed to manage the entire ticket lifecycle—from issue reporting to resolution—ensuring streamlined operations and enhanced service delivery.

The system supports ticket creation, classification, assignment, and closure, with clear visibility into each stage of the process. Role-based access controls have been effectively enforced, ensuring users, support staff, and administrators operate within clearly defined permissions, maintaining system security and integrity.

A multi-tenant architecture was introduced, enabling isolated environments for different organizations or departments. This ensures data privacy and allows for independent configurations, making the solution scalable and adaptable across diverse operational needs.

Real-time status updates and automated notifications are integrated to keep all stakeholders informed at each critical step of the ticket lifecycle. These features enhance responsiveness and ensure prompt attention to user issues.

The user interface was designed with a focus on simplicity and responsiveness, delivering a consistent and intuitive experience across devices—including desktops, tablets, and smartphones.

Additionally, an audit trail and analytics module were implemented, capturing comprehensive logs of ticket activities. This supports performance tracking and provides actionable insights into support trends and operational bottlenecks, contributing to continuous improvement.

Overall, the system provides a structured, secure, and scalable solution that enhances support operations, improves user satisfaction, and enables data-driven decision-making.

Chapter 3: CONCLUSION

The Ticketing System project was designed and developed to streamline the process of handling technical issues and service requests within an organization. By implementing a structured and automated approach, the system ensures that every user-reported issue is properly documented, assigned, and resolved in an organized manner. The platform not only helps IT support teams manage incoming requests more efficiently but also improves overall communication between users and support staff through timely notifications and status updates.

Role-based access control was introduced to maintain security and ensure that users, support agents, and administrators only interact with the components relevant to their responsibilities. The system's multi-tenant support also enables it to scale and serve multiple teams or organizations independently, without data overlap or conflicts.

Additionally, the integration of email notifications, real-time ticket status updates, and intuitive UI design enhances the user experience, making the platform more reliable and easier to use. The project successfully meets its objective of improving ticket visibility, tracking, and resolution efficiency. Overall, this system creates a productive environment for support operations and lays a strong foundation for future enhancements like reporting dashboards, performance analytics, and AI-driven issue classification.

Chapter 4: REFERENCES

1. Documentation - Keycloak
2. https://docs.spring.io/spring-boot/redirect.html
3. https://developer.mozilla.org/en-US/Web/JavaScript
4. https://refactoring.guru/design-patterns/java
5. https://react.dev/learn
6. https://www.typescripttutorial.net/
7. https://www.tutorialspoint.com/microservice_architecture/index.html
8. Stack overflow