Buyer-Talk

AN INTERNSHIP REPORT

Submitted by

Makadiya Raj P.

210120116056

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

Information Technology Department

Gandhinagar Institute of Technology

Gandhinagar

Gujarat Technological University, Ahmedabad

May, 2024-25

Gandhinagar Institute of Technology

Moti Bhoyan Road, Gandhinagar, Gujarat

(Affiliated with GTU)

"WHERE SUCCESS IS A TRADITION"

INFORMATION TECHNOLOGY DEPARTMENT

CERTIFICATE

This is to certify that the work of Internship entitled **"MERN Stack"** has been carried out by ***Makadiya Raj Prafulbhai (210120116056)*** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in ***Information Technology, 8th*** Semester in the ***Gandhinagar Institute of Technology***, Moti-Bhoyan, Gandhinagar, Gujarat, during the academic year 2024-2025 and his work is satisfactory. This student has successfully completed all the activity under my guidance related to Internship for 8th semester.

Internal Guide
Mr. Tushar Pokar

External Guide
Mr. Rahul Kirpekar

Head of the Department
Dr. Mohit Bhadla

PMMS Generated Certificate

GROWNITED

Date: - 17/04/2024

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Raj Prafulbhai Makadiya**, a student of Gandhinagar Institute Of Technology has successfully completed his/her internship in the field of MERN STACK from 20th January 2025 to 19th April 2025 12 Weeks under the guidance **Rahul Kirpekar**.

His internship activities include Project Understanding & Requirement Analysis, Technical Documentation, Learning New Technologies, Development Activities, Testing & Quality Assurance, Soft Skills & Reporting.

During the internship period, he/she was exposed to various processes and was found to be diligent, hardworking, and inquisitive.

We wish him/her all the best in future endeavors.

For, Grownited Private Limited.

Rahul Kirpekar

(Authorised Signature)

301-305, 3rd Floor, Surbhi Complex, Nr. Municipal Market, Chimanlal Girdharlal Road, Navrangpura,Ahmedabad Gujarat- 380009, 7874014621, hr@grownited.com

Gandhinagar Institute of Technology

Moti Bhoyan Road, Gandhinagar ,Gujarat

(Affiliated with GTU)

"WHERE SUCCESS IS A TRADITION"

DECLARATION

We hereby declare that the Internship report submitted along with the Internship entitled BuyerTalk in MERN Stack submitted in partial fulfillment for the degree of Bachelor of Engineering in Information Technology to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me at Grownited Private Limited under the supervision of Mr. Rahul Kirpekar / Mr. Tushar Pokar and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student
Makadiya Raj Prafulbhai

Sign of Student

Acknowledgement

I have taken efforts in this Internship. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Mr. Tushar Pokar & Mr. Rahul Kirpekar** for their guidance and constant supervision as well as for providing necessary information regarding the Internship. I take this opportunity to thank all my friends and colleagues who started me out on the topic and provided extremely useful review feedback and for their all-time support and help in each and every aspect of the course of my project preparation. I am grateful to my college Gandhinagar Institute of Technology, for providing me all required resources and good working environment.

I would like to express my gratitude towards Head of Department, **Dr. Mohit Bhadla** for her kind co-operation and encouragement which help me in this Internship.

Thank You
Makadiya Raj

Abstract

BuyerTalk is a digital platform built with the MERN stack (MongoDB, Express.js, React, Node.js) that connects consumers, businesses, and regulatory bodies to foster transparent communication, product reviews, feedback sharing, and complaint resolution. It empowers users to raise concerns, receive prompt responses, and stay informed about their rights and consumer laws. With features like complaint registration, business response management, real-time updates, and role-based access for consumers, businesses, regulators, and admins, BuyerTalk ensures a structured and secure environment. Its user-friendly interface, strong authentication, and data security measures aim to enhance consumer protection, promote business accountability, and build trust in a fair, responsive consumer-business ecosystem.

Internship Content

List Of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Introduction of Company

Grownited Private Limited is a forward-thinking technology company specializing in delivering innovative software solutions. Established with a vision to empower businesses and promote youth entrepreneurship, the company focuses on transforming ideas into reality using modern technologies with expertise in custom software development, web and mobile applications, and enterprise solutions, Grownited is dedicated to providing high-quality, scalable, and secure digital products. The company's commitment to innovation and customer satisfaction has positioned it as a trusted partner for businesses across various industries.

1.2 Aim and Objectives

Aim: To develop and deliver innovative, reliable, and user-centric software solutions that empower businesses and enhance user experiences using modern technology.

Objective:

1. **Provide Innovative Solutions**– Develop cutting-edge applications that solve real-world problems efficiently.
2. **Promote Digital Transformation** Assist businesses in transitioning to digital platforms for improved operations and customer engagement.
3. **Ensure Customer Satisfaction**– Deliver high-quality, customized solutions that meet client requirements.
4. **Encourage Youth Entrepreneurship**– Support and mentor young entrepreneurs in transforming their ideas into successful digital products.
5. **Focus on Continuous Improvement**– Adapt to evolving technology trends to offer competitive and scalable solutions.
6. **Maintain Quality and Security** – Ensure robust security measures and follow industry standards in all projects.

1.3 Introduction of Organization

Grownited Private Limited is an emerging technology company dedicated to providing innovative digital solutions by bridging the gap between ideas and execution. Specializing in customized software development, web and mobile application creation, and enterprise solutions, the company empowers organizations with tailored technology. With a strong emphasis on quality, innovation, and customer satisfaction, Grownited has earned a reputation as a reliable technology partner. Backed by a team of experienced professionals, the company continuously embraces the latest advancements to deliver scalable, secure, and user-friendly solutions across various industries.

CHAPTER 2: INTRODUCTION TO INTERNSHIP

2.1 Internship Summary

During my internship at Grownited Private Limited, I worked as a MERN Stack Developer on the Buyer Talk project, a platform for consumer discussions, reviews, feedback, and complaint resolution. My responsibilities included developing user interfaces using React.js, creating RESTful APIs with Express.js and Node.js, and managing data with MongoDB.

This experience enhanced my technical skills in full-stack development and provided practical exposure to project management, collaboration, and problem-solving in a professional setting.

2.2 Purpose and Objective

The primary purpose of my internship at Grownited Private Limited is to gain practical experience in software development using the MERN stack while contributing to the development of the BuyerTalk platform. This opportunity allows me to apply my academic knowledge to a real-world project, enhancing my technical and problem-solving skills. By working on a platform that facilitates transparent communication between consumers, businesses, and regulatory bodies, I aim to understand how technology can be used to solve practical problems and create a positive impact.

The key objectives of my internship include developing a strong proficiency in MongoDB, Express.js, React, and Node.js through hands-on implementation. Additionally, I aim to understand the complete software development lifecycle, from planning and designing to coding, testing, and deployment. This experience will help me improve my collaborative skills by working closely with a team of professionals and applying best practices in software engineering. Furthermore, I seek to contribute effectively to the BuyerTalk project by building user-friendly interfaces, ensuring seamless data management, and optimizing system performance. Through this experience, I aspire to become a more competent and industry-ready software developer.

2.3 Tools & Technologies and Language Intro

During my internship at Grownited Private Limited, I worked with a variety of modern tools and technologies to develop a web-based application, BuyerTalk. These tools and frameworks played a crucial role in building a dynamic, responsive, and efficient platform. The primary technologies used in the project included MongoDB, Exprerss.js, React.js, and Node.js along with supporting tools for development, testing, and deployment.

1. VS Code – The primary code editor used for writing and debugging code.
2. Git & GitHub – Version control tools for managing project repositories and collaboration.
3. HTML & CSS – Core technologies for structuring and styling the web application, with Tailwind CSS providing enhanced design capabilities.
4. JavaScript – Used for building dynamic and interactive frontend components, mainly with React.js and Next.js.
5. React.js – A popular JavaScript library for building user interfaces with reusable components.
6. Node.js – Node.js is a JavaScript runtime environment for server-side scripting with an asynchronous, event-driven architecture, while HTML and CSS build the UI structure and styling.
7. Bootstrap CSS – A utility-first CSS framework used for designing a responsive and visually appealing frontend.
8. MongoDB – A flexible and scalable NoSQL document database for storing JSON-like data, offering efficient querying, indexing, and aggregation.
9. Express.js – A minimalist and flexible Node.js web framework for building APIs and web applications, offering rapid development with routing, middleware, and templating functionalities.
10. Postman – A tool for testing API requests and responses.

2.4 Internship Planning

Phase 1: Understanding and Requirement Analysis (Week 1-2)

- Introduction to the project and its objectives.
- Understanding the problem statement and gathering requirements.
- Researching similar platforms and analyzing user needs.
- Exploring the MERN stack and project management tools.Learning project management methodologies and understanding task breakdown.

Phase 2: Design and Planning (Week 3-4)

- Designing the system architecture and database schema using MongoDB.
- Creating wireframes and UI mockups for frontend components using Figma or similar tools.
- Planning the API structure and defining endpoints.
- Finalizing the project timeline and task distribution.

Phase 3: Frontend Development (Weeks 5-7)

- Developing the user interface using React.js.
- Implementing reusable components for efficient UI management.
- Ensuring responsive design for cross-device compatibility.
- Integrating form validation and user authentication features.
- Testing the frontend design on multiple devices for consistency.

Phase 4: Backend Development (Weeks 8-9)

- Building the backend using Node.js and Express.js.
- Creating RESTful APIs for CRUD operations.
- Implementing authentication and authorization using JWT.
- Connecting the application to MongoDB for data management.

Phase 5 Integration and Testing (Weeks 10-11)

- Integrating frontend and backend systems.
- Performing unit testing, API testing using Postman, and end-to-end testing.
- Debugging errors and optimizing code.

- Ensuring proper data flow and error handling..

Phase 6: Deployment and Documentation (Weeks 12-13)

- Deploying the application to a suitable cloud platform.
- Conducting performance testing and fixing bugs.
- Preparing technical documentation, including API documentation and user manuals.

Phase 7: Final Review and Presentation (Weeks 14)

- Conducting a final project review.
- Gathering feedback from mentors and making necessary improvements.
- Preparing the internship report and project presentation.
- Demonstrating the completed project to stakeholders.

CHAPTER 3: INTERNSHIP IMPLEMENTATION

3.1 Internship Weekly Task

Week 1 (20/01/2025 - 24/01/2025)

- Project Allocation and Workflow Understanding.
- Introduction to the project and its objectives.
- Overview of project architecture, key modules, and functionalities.
- Familiarized with development tools and project management methodologies.

Week 2 (27/01/2025 - 31/01/2025)

- React.js Basics and Project Requirements.
- Learned React.js fundamentals: components, props, state management, and lifecycle methods.
- Understood reusable components for efficient UI development.
- Analyzed project requirements for frontend implementation.

Week 3 (03/02/2025 - 07/02/2025)

- Gained hands-on experience with React Hooks: `useState`, `useEffect`, and `useContext`.
- Applied state management and handled side effects using hooks.
- Developed dynamic and responsive components.

Week 4 (10/02/2025 - 14/02/2025)

- Completion of Login, Signup, and Landing Page
- Implemented Login, Signup, and Landing Pages using React.js.
- Integrated APIs and applied form validation.
- Ensured a seamless user experience.

Week 5 (17/02/2025 - 21/02/2025)

- UI Development and Error Resolution
- Created Sidebar, Navbar, and other UI elements.
- Ensured responsive design and design guideline compliance.
- Collaborated for code review and resolved issues.

Week 6 (Date: 24/02/2025 to 28/02/2025)

- Backend Learning with MongoDB
- Learned MongoDB database management using collections and documents.
- Implemented CRUD (Create, Read, Update, Delete) operations.
- Applied database concepts to manage project data.

Week 7 (Date: 03/03/2025 to 07/03/2025)

- API Development and Testing
- Developed APIs using Node.js and MongoDB.
- Conducted API testing using Postman.
- Ensured correct functionality, handled edge cases, and validated API responses.

Week 8 (Date: 10/03/2025 to 13/03/2025)

- Integration of User and Service Provider Modules**
- Integrated backend APIs with frontend modules.
- Established seamless data communication.
- Implemented authentication and data management features.

Week 9 (Date: 17/03/2025 to 21/03/2025)

- Code Review, Project Testing, and Error Resolution
- Participated in a detailed code review session.
- Applied feedback for code quality improvement.
- Conducted testing to identify and resolve bugs.
- Achieved significant project progress with more than half of the project completed.

Week 10 (Date: 24/03/2025 to 28/03/2025)

- Learned and implemented Context API in React for efficient global state management.
- Eliminated prop drilling by sharing state across multiple components seamlessly.
- Applied Context API to enhance communication between User and Service Provider modules.
- Resolved project-related doubts through team discussions and **o**nline resources, improving problem-solving skills.

Week 11 (Date: 31/03/2025 to 04/04/2025)

- Revisited and strengthened core React concepts like props and state for dynamic UI handling.
- Understood limitations of prop/state management in large-scale applications.
- Learned and implemented Redux for centralized state management and predictable state flow.
- Conducted testing to identify and resolve bugs.

Week 12 (Date: 07/04/2025 to 11/04/2025)

- Code Review, Project Testing, and Error Resolution**
- Participated in a detailed code review session.
- Applied feedback for code quality improvement.
- Integrated Redux into the project to efficiently manage shared data across components.

Week 13 (Date: 14/04/2025 to 18/04/2025)

- Created the final internship report detailing all project phases and learnings.
- Designed a professional PowerPoint presentation to showcase the project highlights.
- Compiled the final documentation, including code structure, features, and technical stack used.

3.2 Flowchart / Pseudo code of the task

```
START

// User Registration and Login
FUNCTION RegisterUser(userDetails)
    IF userDetails are valid THEN
        Save userDetails to Database
        Display "Registration Successful"
    ELSE
        Display "Invalid Registration Details"
    END IF
END FUNCTION

FUNCTION LoginUser(email, password)
    IF email and password match Database THEN
        Display "Login Successful"
        RETURN UserRole
    ELSE
        Display "Invalid Email or Password"
    END IF
END FUNCTION

// Dashboard Based on User Role
FUNCTION DisplayDashboard(UserRole)
    SWITCH (UserRole)
        CASE "Consumer":
            Display ConsumerDashboard()
        CASE "Business":
            Display BusinessDashboard()
        CASE "Regulator":
            Display RegulatorDashboard()
        CASE "Admin":
            Display AdminDashboard()
        DEFAULT:
            Display "Invalid Role"
    END SWITCH
END FUNCTION
```

Fig 3.2.1 Pseudo code of Registration, Login and User

```
// Consumer Functionalities
FUNCTION ConsumerDashboard()
    DISPLAY "1. File Complaint"
    DISPLAY "2. Write Review"
    DISPLAY "3. View Complaint Status"
    DISPLAY "4. View Product Reviews"
    CHOICE = GET User Input

    SWITCH (CHOICE)
        CASE 1:
            FileComplaint()
        CASE 2:
            WriteReview()
        CASE 3:
            ViewComplaintStatus()
        CASE 4:
            ViewProductReviews()
        DEFAULT:
            DISPLAY "Invalid Option"
    END SWITCH
END FUNCTION

FUNCTION FileComplaint()
    GET ComplaintDetails from User
    Save to Database
    DISPLAY "Complaint Submitted Successfully"
END FUNCTION

FUNCTION WriteReview()
    GET ReviewDetails from User
    Save to Database
    DISPLAY "Review Submitted Successfully"
END FUNCTION
```

Fig 3.2.2 Pesudo code of Consumer Functionalities

```
// Business Functionalities
FUNCTION BusinessDashboard()
    DISPLAY "1. View Complaints"
    DISPLAY "2. Respond to Complaints"
    DISPLAY "3. Manage Products"
    CHOICE = GET User Input

    SWITCH (CHOICE)
        CASE 1:
            ViewComplaints()
        CASE 2:
            RespondToComplaint()
        CASE 3:
            ManageProducts()
        DEFAULT:
            DISPLAY "Invalid Option"
    END SWITCH
END FUNCTION

// Regulator Functionalities
FUNCTION RegulatorDashboard()
    DISPLAY "1. Monitor Complaints"
    DISPLAY "2. Provide Resolution"
    CHOICE = GET User Input

    SWITCH (CHOICE)
        CASE 1:
            MonitorComplaints()
        CASE 2:
            ProvideResolution()
        DEFAULT:
            DISPLAY "Invalid Option"
    END SWITCH
END FUNCTION
```

Fig 3.2.3 Pesudo Code of Business and Regulator Functionalities

```
// Admin Functionalities
FUNCTION AdminDashboard()
    DISPLAY "1. Manage Users"
    DISPLAY "2. Manage Complaints"
    DISPLAY "3. Generate Reports"
    CHOICE = GET User Input

    SWITCH (CHOICE)
        CASE 1:
            ManageUsers()
        CASE 2:
            ManageComplaints()
        CASE 3:
            GenerateReports()
        DEFAULT:
            DISPLAY "Invalid Option"
    END SWITCH
END FUNCTION

END
```

Fig 3.2.4 Pesudo Code of Admin Functionalities

3.3 Roles & Responsibilities

An internship in MERN stack development typically involves a range of roles and responsibilities, aimed at providing hands-on experience with modern web development technologies. Here's a breakdown:

Learning and Training: Interns are expected to actively engage in learning the MERN stack technologies, including MongoDB, Express.js, React.js, and Node.js. This involves studying documentation, tutorials, and actively participating in training sessions.

Assisting Development Teams: Interns may assist development teams in various stages of the software development lifecycle. This can include participating in

brainstorming sessions, contributing to code reviews, and providing feedback on development tasks.

Frontend Development: Interns may be tasked with frontend development responsibilities using React.js. This can involve building user interfaces, implementing UI designs, and optimizing frontend performance.

Backend Development: Interns may work on backend development tasks using Node.js and Express.js. This can include building RESTful APIs, handling database operations, and implementing server-side logic.

Database Management: Interns may gain experience in MongoDB database management, including database setup, querying data, and implementing data models. all of which contributed to building an efficient and user-friendly online calculator platform.

3.4 Internship Scheduling (Grant Chart/ PERT/ Network Chart)

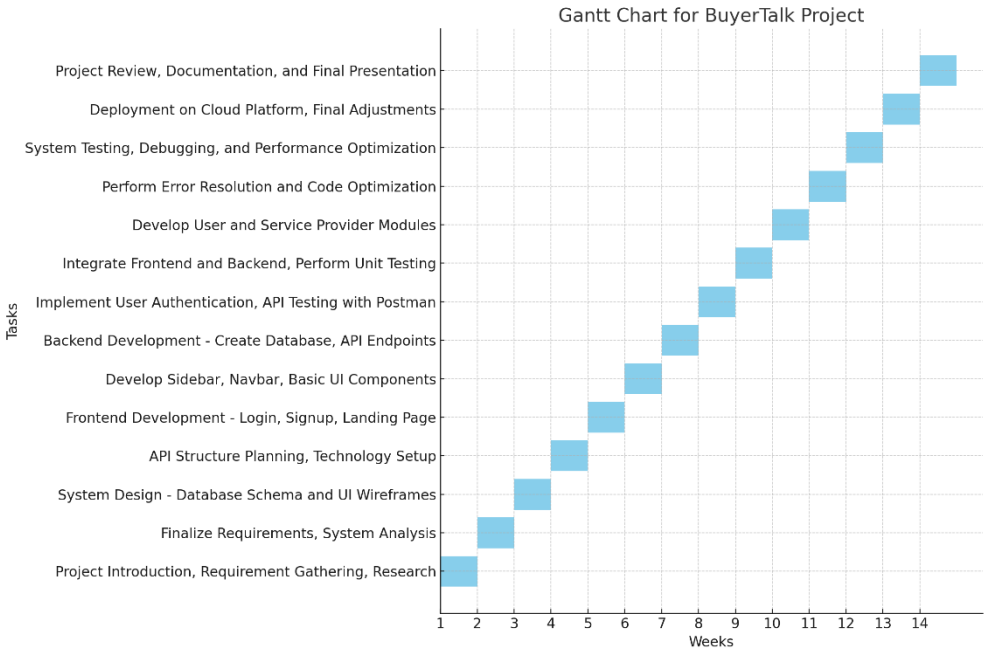


Fig.3.4.1Gantt Chart

CHAPTER 4: DESIGNS

4.1 System Flow Diagram

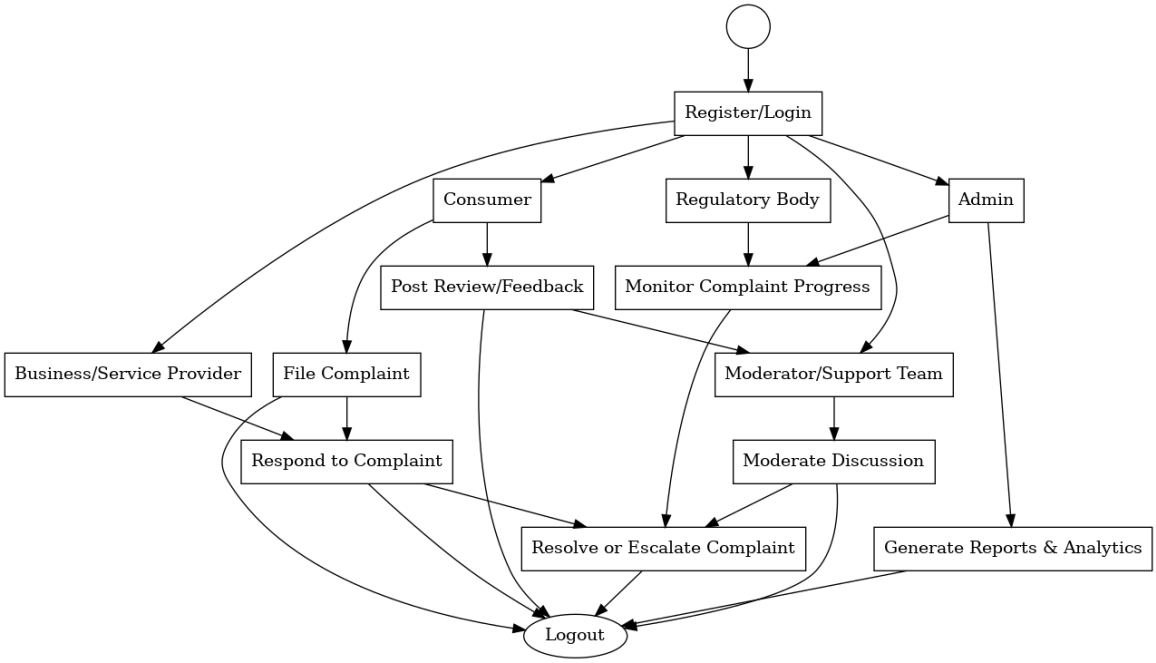


Fig 4.1.1 System Flow Diagram

4.2 Data Dictonary

Attribute	Description	Data Type	Constraints
user_id	Unique identifier for the user	INTEGER	Primary Key, Auto-increment
name	Full name of the user	VARCHAR(100)	Not Null
email	Email address	VARCHAR(150)	Unique, Not Null
password	User password (hashed)	VARCHAR(255)	Not Null
role	Role of the user (Admin, Consumer, etc.)	ENUM	Not Null
registration_date	Date of account creation	DATE	Not Null
status	Account status (Active, Suspended)	ENUM	Default: Active

Table 4.2.1 User Table

Attribute	Description	Data Type	Constraints
service_id	Unique identifier for the business	INTEGER	Primary Key, Auto-increment
user_id	Reference to business owner (User)	INTEGER	Foreign Key (Users)
service_name	Name of the business	VARCHAR(150)	Not Null
description	Business overview	TEXT	Optional
registration_date	Date the business joined the platform	DATE	Not Null
status	Business status (Active, Pending)	ENUM	Default: Pending

Table 4.2.2 Service Table

Attribute	Description	Data Type	Constraints
product_id	Unique identifier for the product/service	INTEGER	Primary Key, Auto-increment
service_id	Associated business	INTEGER	Foreign Key (Businesses)
name	Product or service name	VARCHAR(100)	Not Null
description	Details of the product/service	TEXT	Optional
category	Product category	VARCHAR(50)	Not Null
price	Cost of the product/service	DECIMAL(10,2)	Optional
added_date	Date added to the platform	DATE	Not Null

Table 4.2.3 Product Table

Attribute	Description	Data Type	Constraints
complaint_id	Unique complaint identifier	INTEGER	Primary Key, Auto-increment
user_id	Consumer who filed the complaint	INTEGER	Foreign Key (Users)
product_id	Related product/service	INTEGER	Foreign Key (Products/Services)
description	Complaint details	TEXT	Not Null
status	Complaint status (Open, Resolved, Escalated)	ENUM	Default: Open
filed_date	Date complaint was submitted	DATE	Not Null
resolution_date	Date complaint was resolved	DATE	Optional

Table 4.2.4 Complaint Table

Attribute	Description	Data Type	Constraints
review_id	Unique review identifier	INTEGER	Primary Key, Auto-increment
user_id	Reviewer (Consumer)	INTEGER	Foreign Key (Users)
product_id	Reviewed product/service	INTEGER	Foreign Key (Products/Services)
rating	Product/service rating (1-5 stars)	INTEGER	Not Null, Check (1 <= rating <= 5)
comment	Review comment	TEXT	Optional
review_date	Date of review submission	DATE	Not Null

Table 4.2.5 Review Table

4.3 Relationship of Table

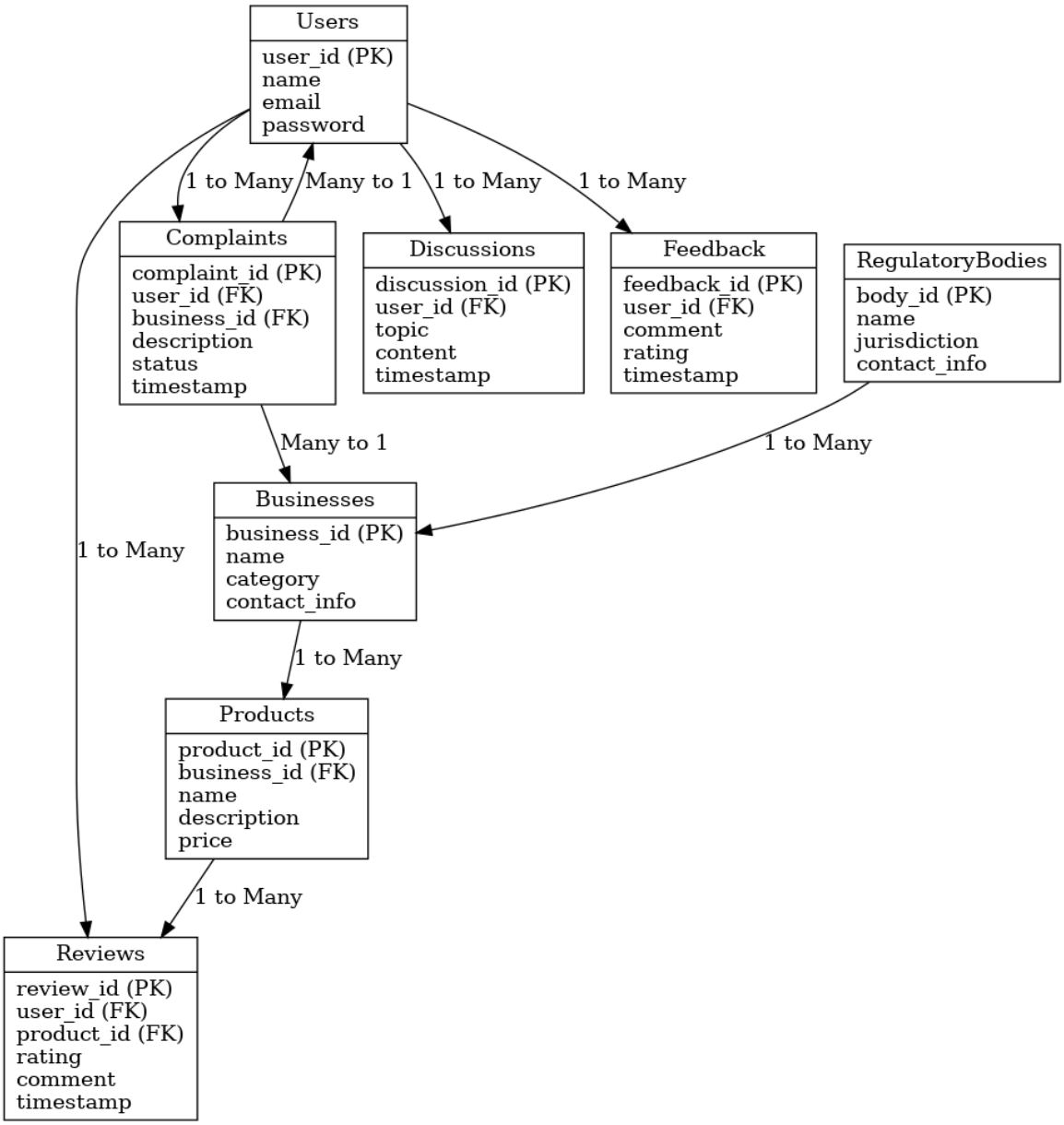


Fig 4.3.1 Relationship of Table

4.4 User Interface

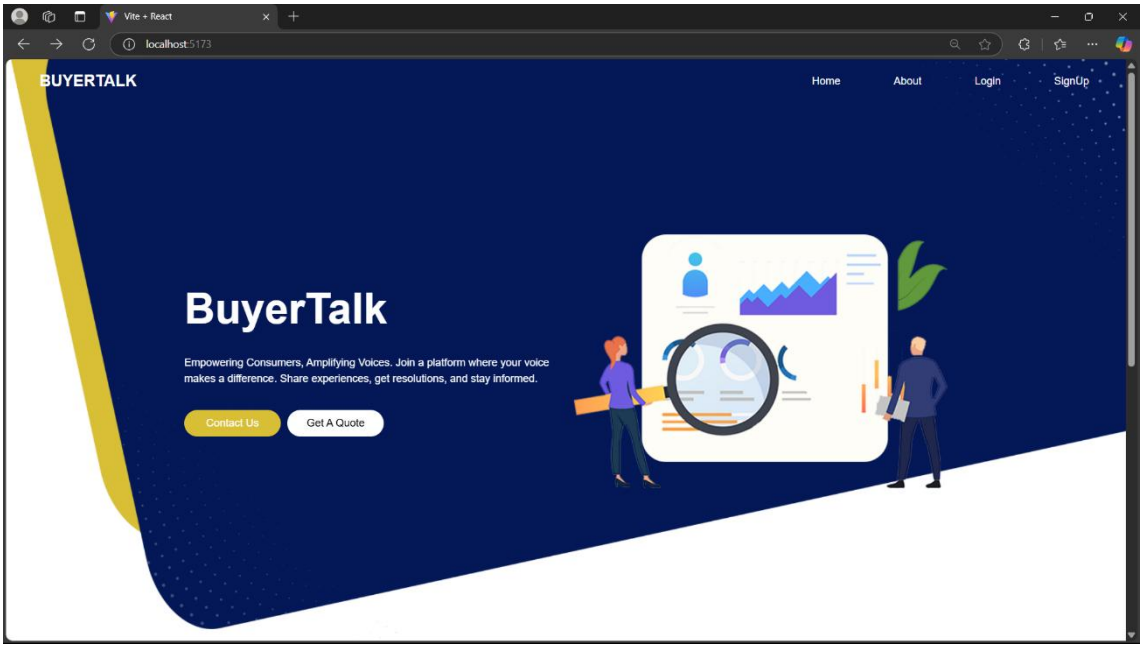


Fig 4.4.1 Home Page

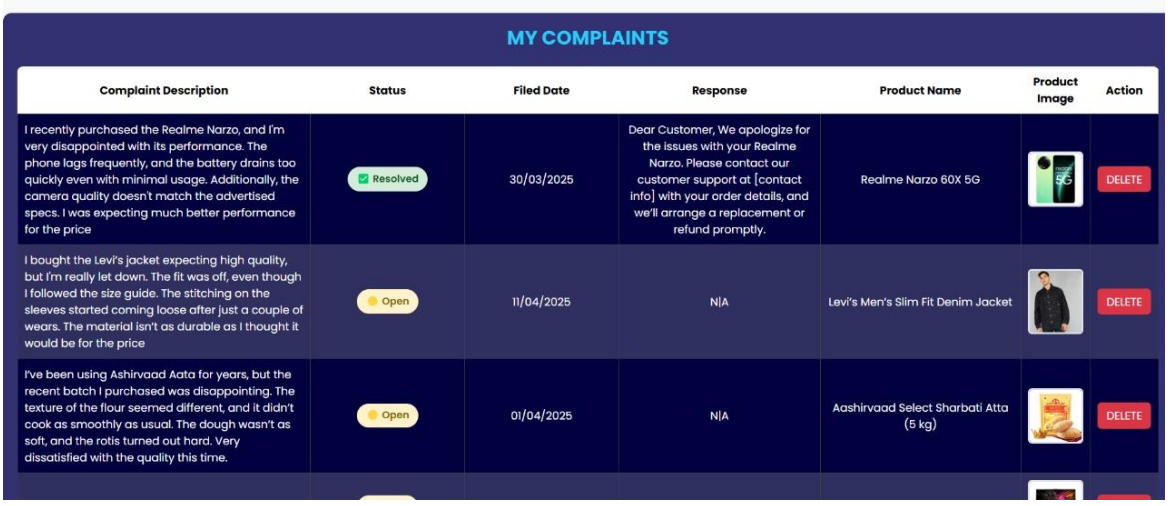


Fig 4.4.2 User Page

Fig 4.4.3 Service Provider Page

CHAPTER 5: TESTING

5.1 Testing Plan/Strategy

The testing plan ensures that all functionalities of the BuyerTalk platform work correctly, efficiently, and securely. The goal is to identify and resolve any issues related to user experience, functionality, performance, security, and integration before deployment. Through rigorous testing, the platform will be validated to ensure a seamless and reliable experience for consumers, businesses, and regulatory bodies.

5.1.1 Testing Levels

- Unit Testing: Tests individual components and functions such as user authentication, complaint submission, feedback forms, and business responses to ensure correct outputs
- Integration TestingEnsures seamless interaction between the React.js frontend, Node.js backend, and MongoDB database. Validates API calls for complaint registration, response management, and user communication.
- System Testing: Evaluates the entire platform to ensure functionalities like user registration, login, complaint management, discussion forums, and notifications work correctly. Verifies UI responsiveness and overall system performance.
- User Acceptance Testing (UAT): Conducted to ensure the platform meets user expectations. Consumers, businesses, and regulatory bodies will navigate the application to validate its usability, functionality, and satisfaction with the complaint resolution process.

5.1.2 Testing Strategies

1. Functional Testing:

   • Verify that users can successfully register, log in, and submit complaints.

   • Test the functionality of discussion forums, feedback submission, and complaint tracking.

   • Ensure role-based access control for consumers, businesses, and regulatory bodies.

- Validate API requests and responses for complaint management and resolution.

2. Performance Testing:

- Measure the platform's loading time and response rate during high-traffic scenarios.
- Test API response times under stress using tools like Postman or JMeter.
- Optimize MongoDB queries for efficient data retrieval.

3. UI/UX Testing:

- Ensure a responsive design across desktop, tablet, and mobile devices.
- Check the layout, buttons, forms, and navigation flow for usability.
- Validate proper error messages, input validation, and user-friendly interactions.

4. Security Testing:

- Prevent unauthorized access using secure authentication and authorization mechanisms.
- Perform data validation to prevent SQL Injection, XSS attacks, and other vulnerabilities.
- Ensure that sensitive information like passwords is securely encrypted.

5. Regression Testing:

- Re-run test cases after updates to ensure new changes do not break existing functionalities.
- Ensure new changes do not break existing functionalities.

5.1.3 Testing Tools

1. Jest (for JavaScript unit testing)
2. Postman (for API testing)
3. Lighthouse (for performance and accessibility testing)
4. Manual Testing (for UI and user experience evaluation)

5.2 Test Result and Analysis

Test Case ID	Test Scenario	Input Data	Expected Result	Actual Result
TC001	Verify user registration	Valid name, email, password	User should be registered successfully	Pass
TC002	Verify user login	Valid email and password	User should be logged in successfully	Pass
TC003	Verify invalid login attempt	Invalid email or password	Error message should be displayed	Pass
TC004	Verify complaint submission by consumer	Complaint details (e.g., product name, issue)	Complaint should be submitted and visible in dashboard	Pass
TC005	Verify business response to a complaint	Business response to consumer complaint	Consumer should receive the response notification	Pass
TC006	Verify discussion forum functionality	Post a comment or query	Comment should be visible on the forum	Pass
TC007	Verify regulatory body complaint monitoring	Complaint status check by regulatory body	Complaint progress should be visible	Pass
TC008	Verify API response for complaint data	API request with valid numbers	Correct complaint details should be returned	Pass
TC009	Verify UI responsiveness on different devices	Open the calculator on mobile/tablet	UI should adapt and display properly	Pass

TC0010	Verify system behavior on high traffic	Multiple user logins and complaint submissions	System should handle the load without crashing	Pass

Table 5.2.1 Test Result Table

CHAPTER 6: OUTCOMES

6.1 Results & Screenshots

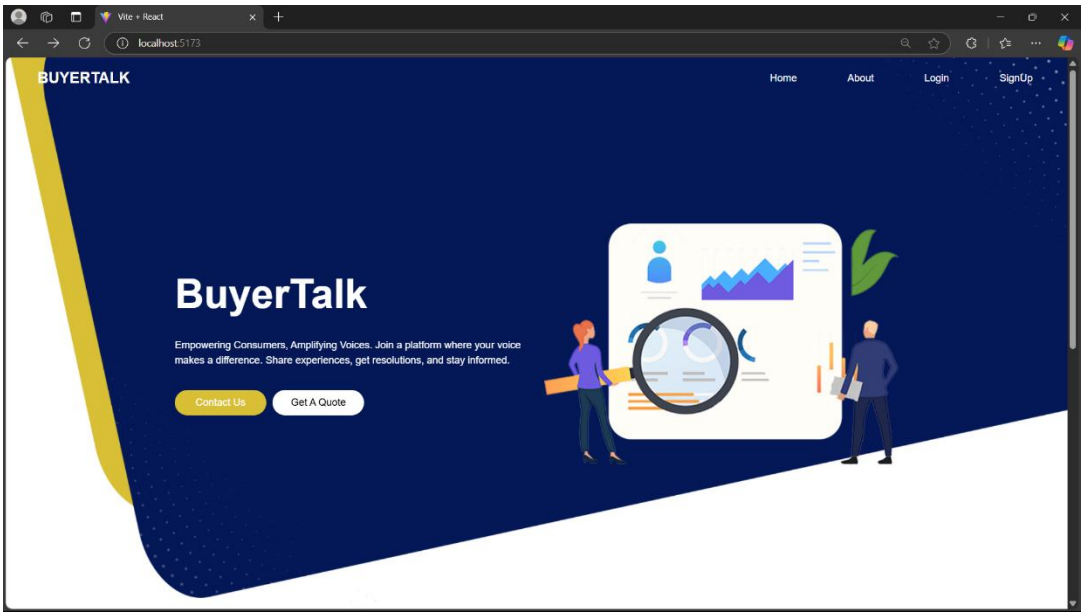


Fig 6.1.1 Home Page

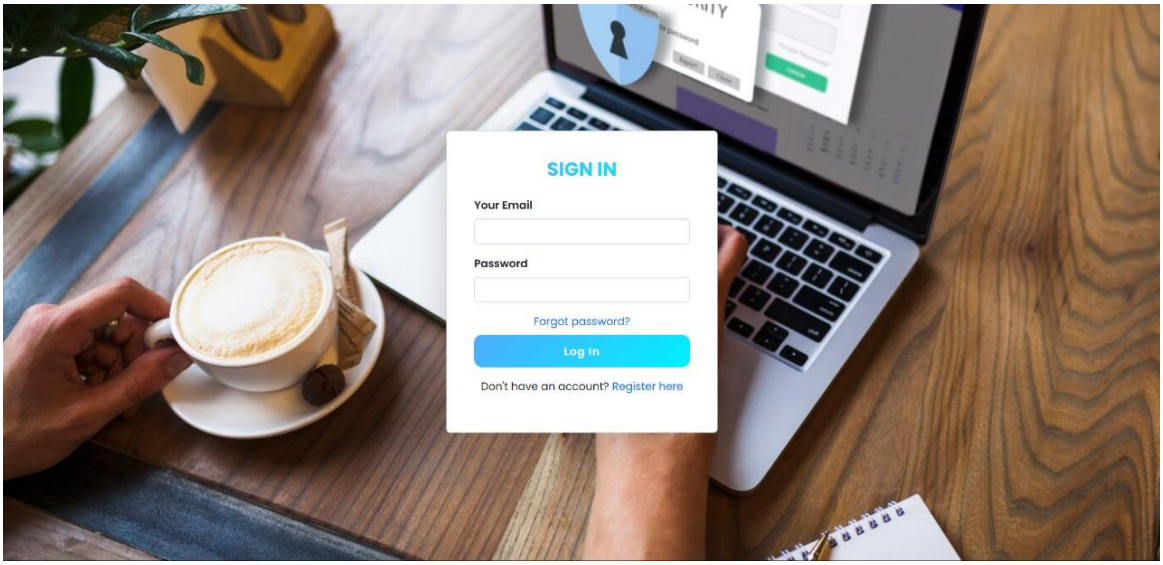


Fig 6.1.2 Login Page

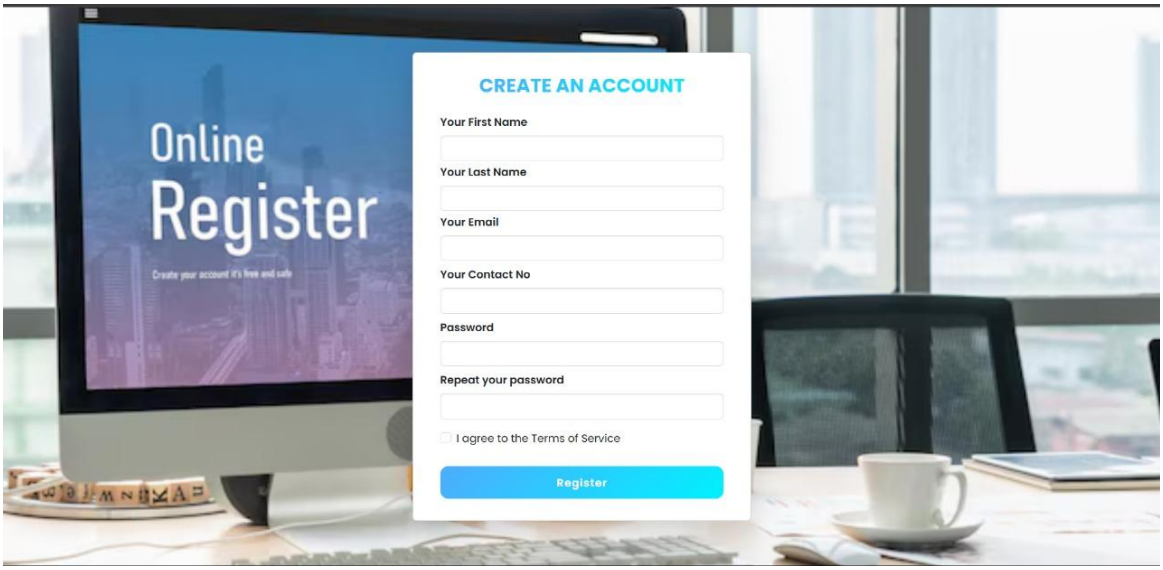


Fig 6.1.3 Sign Up

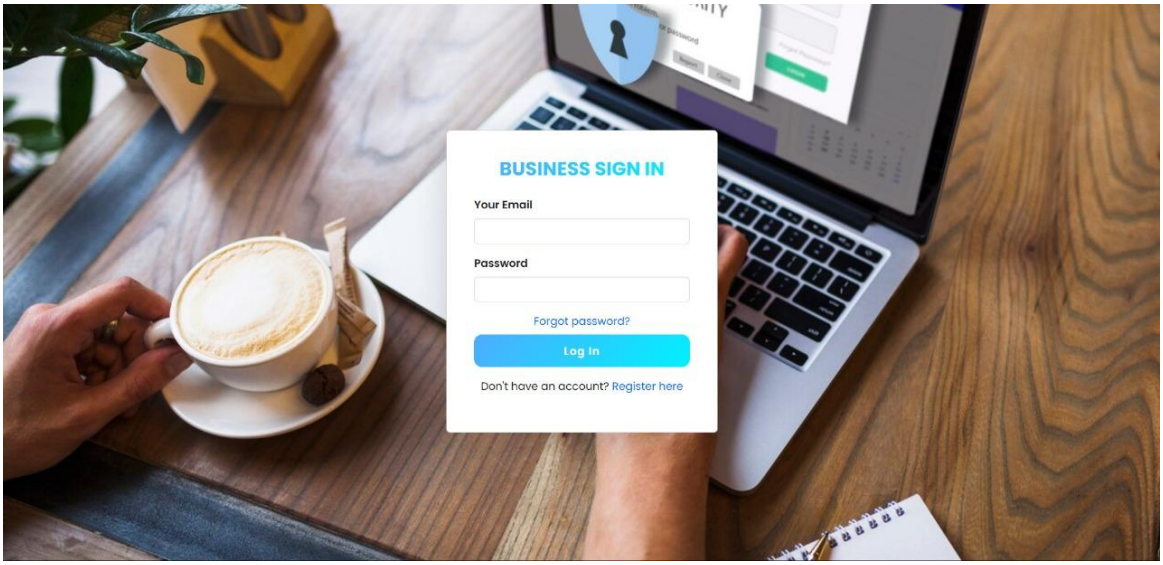


Fig 6.1.4 Service Provider Login Page

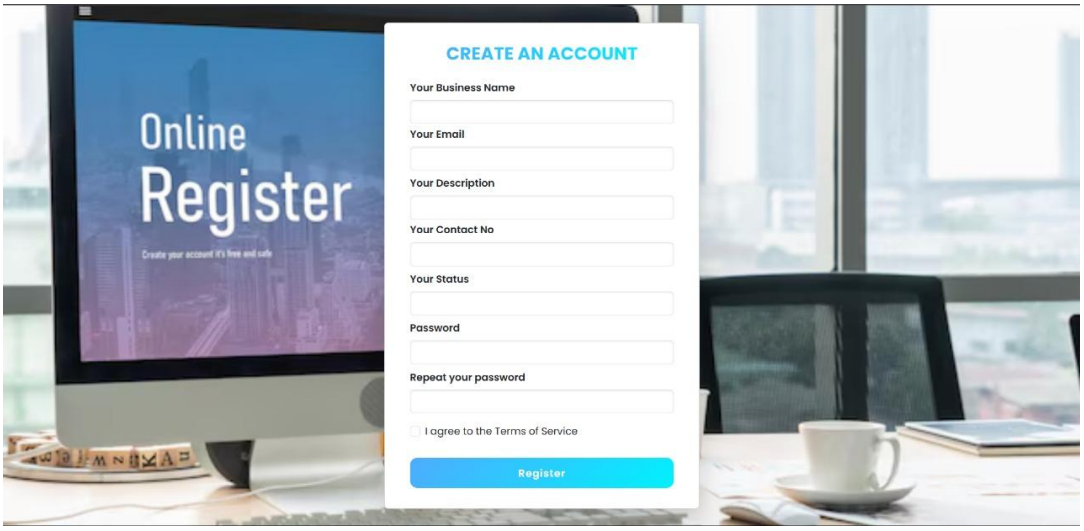


Fig 6.1.5 Service Provider Signup Page

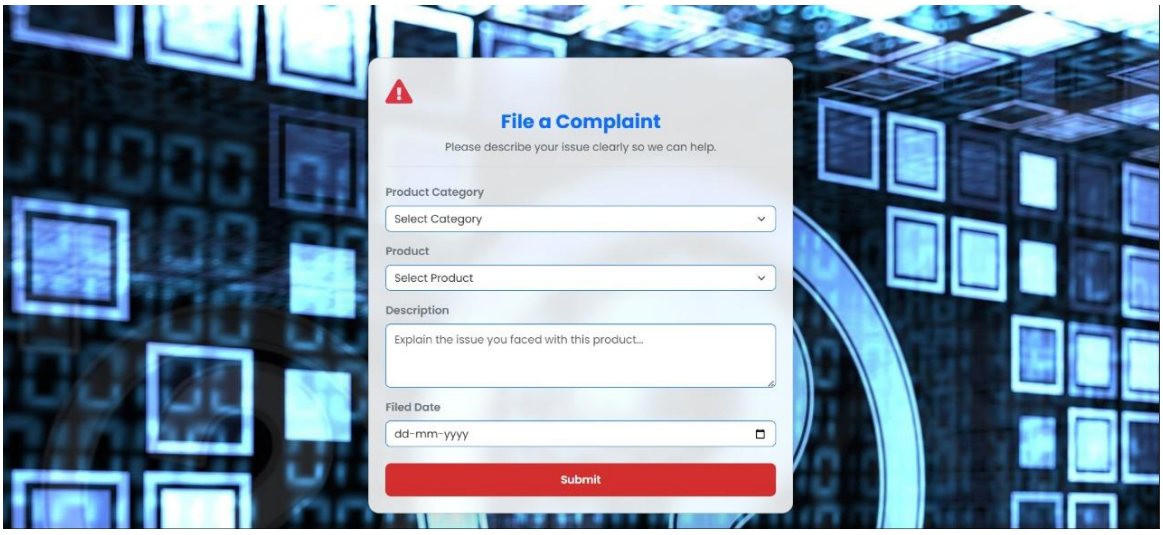


Fig 6.1.6 User Page

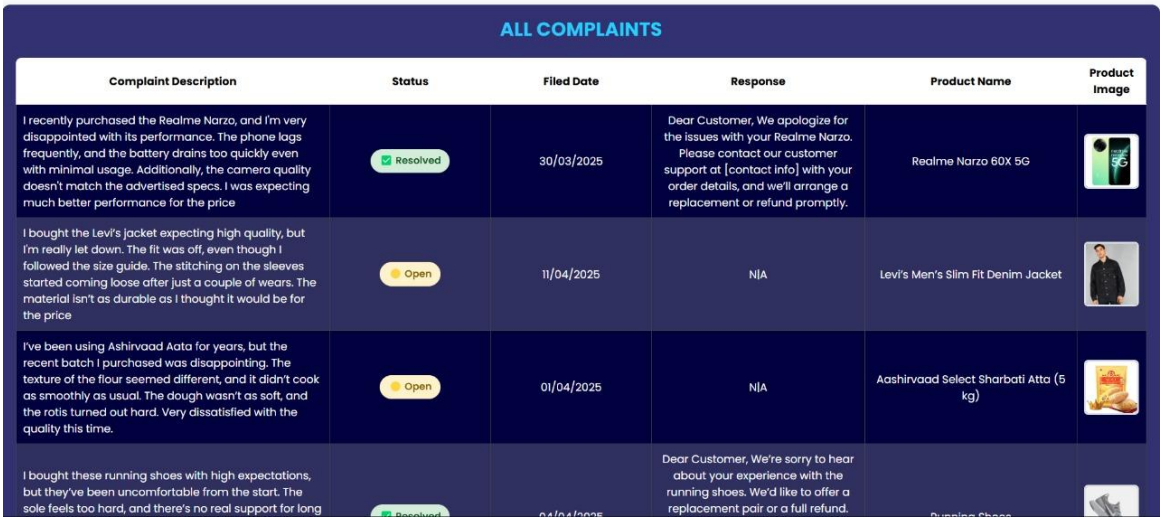


Fig 6.1.7 Complaints Page

Fig 6.1.8 User Complaints Page

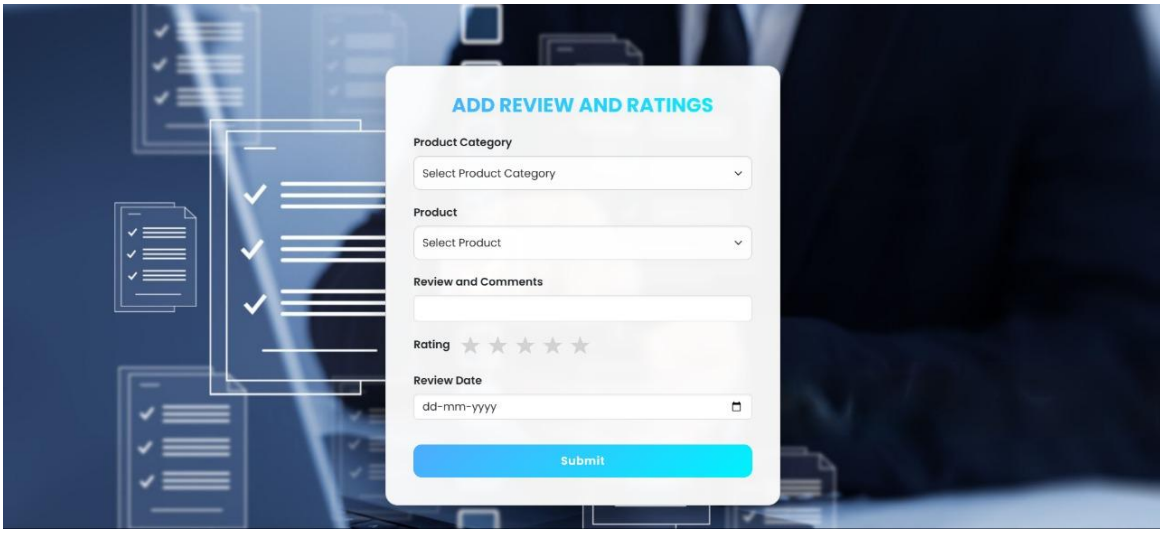
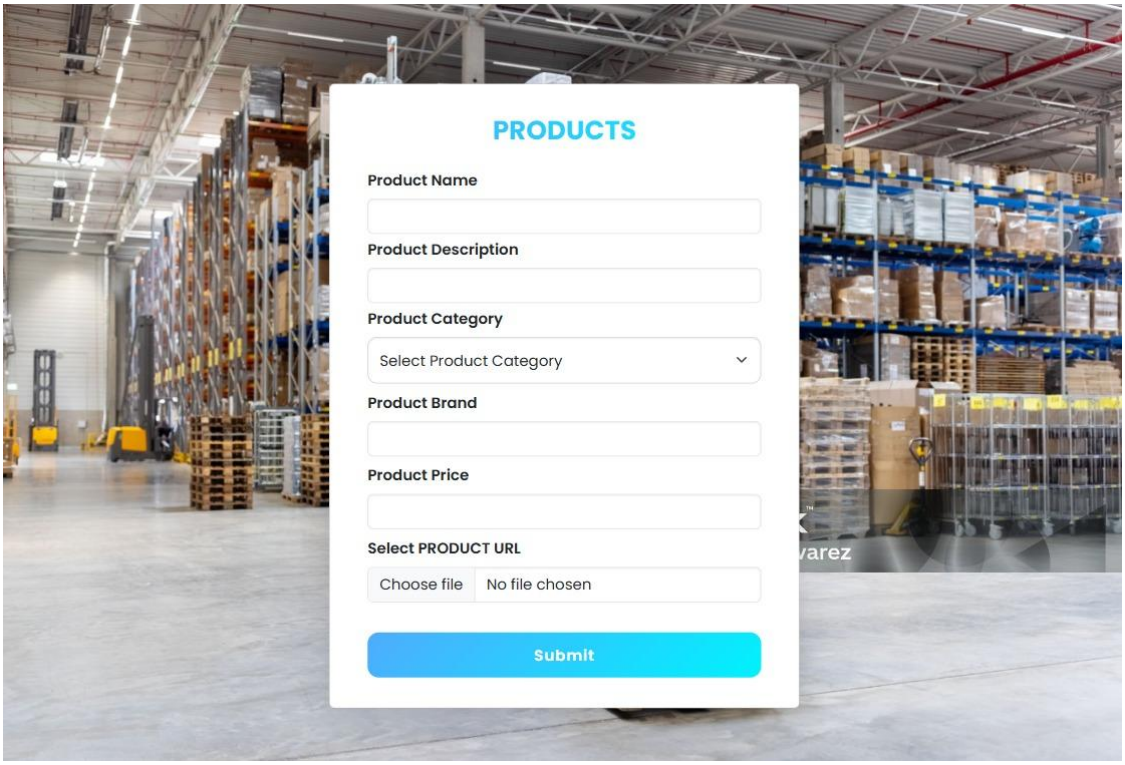


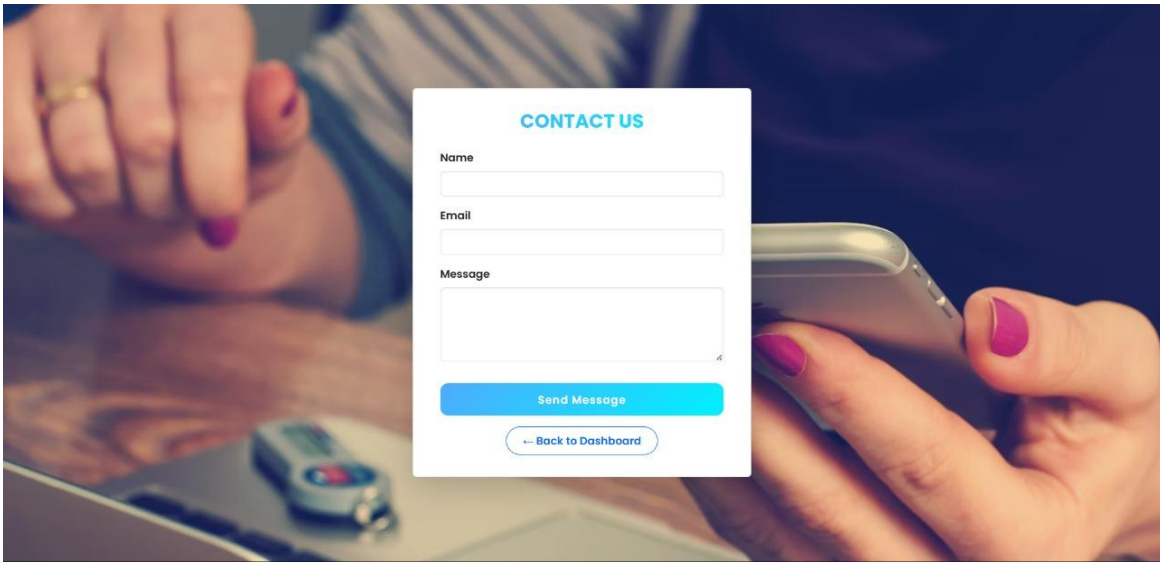
Fig 6.1.9 Add Review Page

Fig 6.1.10 User Review Page

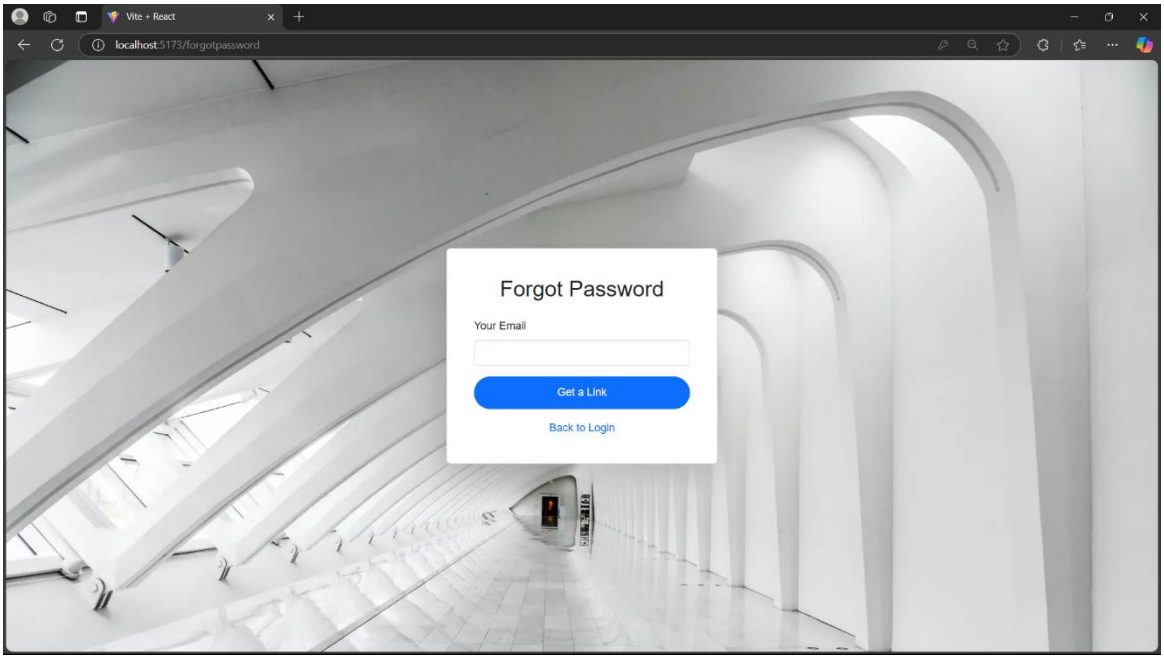


6.1.11 Service Provider Page

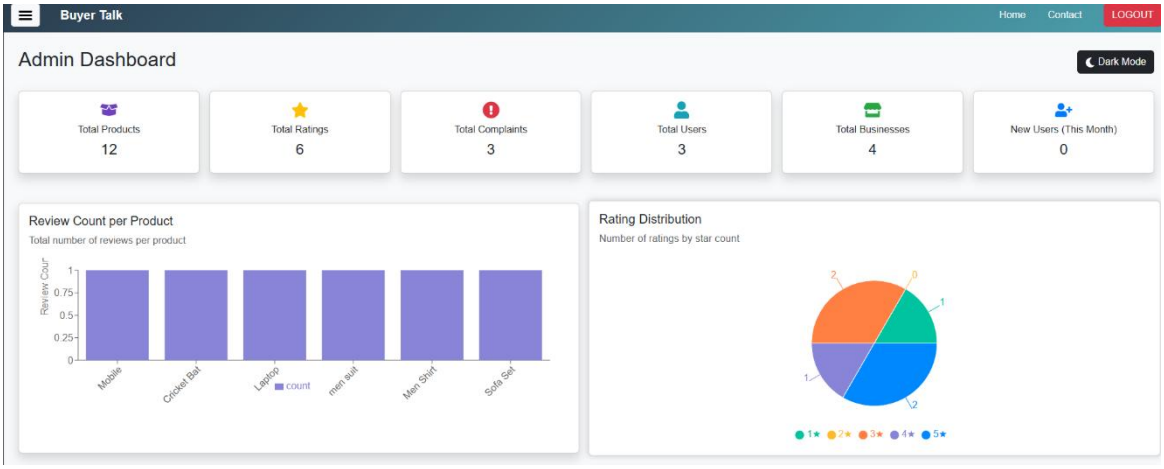
6.1.12 View All Product Page



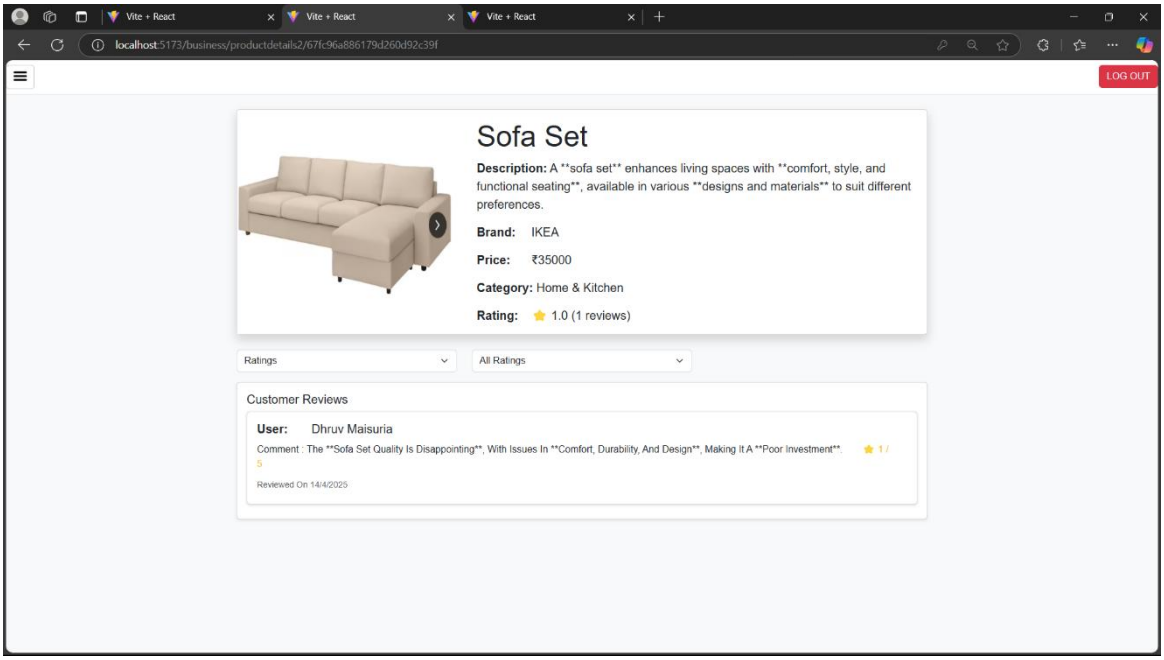
6.1.13 Contact Us Page



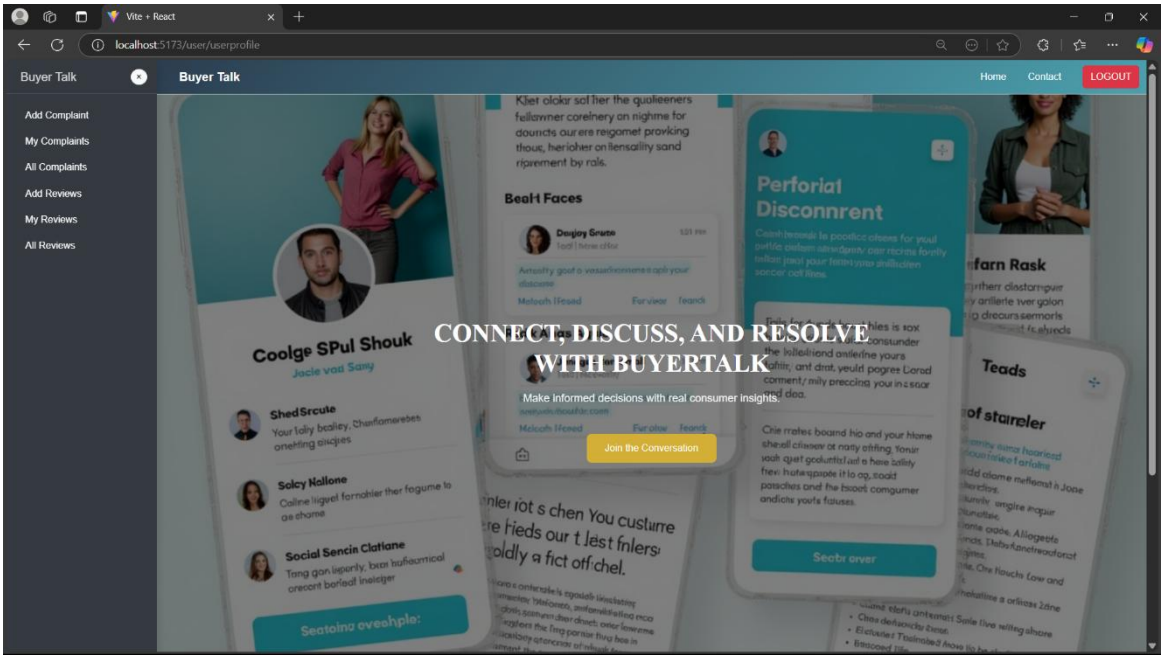
6.1.14 Forgot Password Page



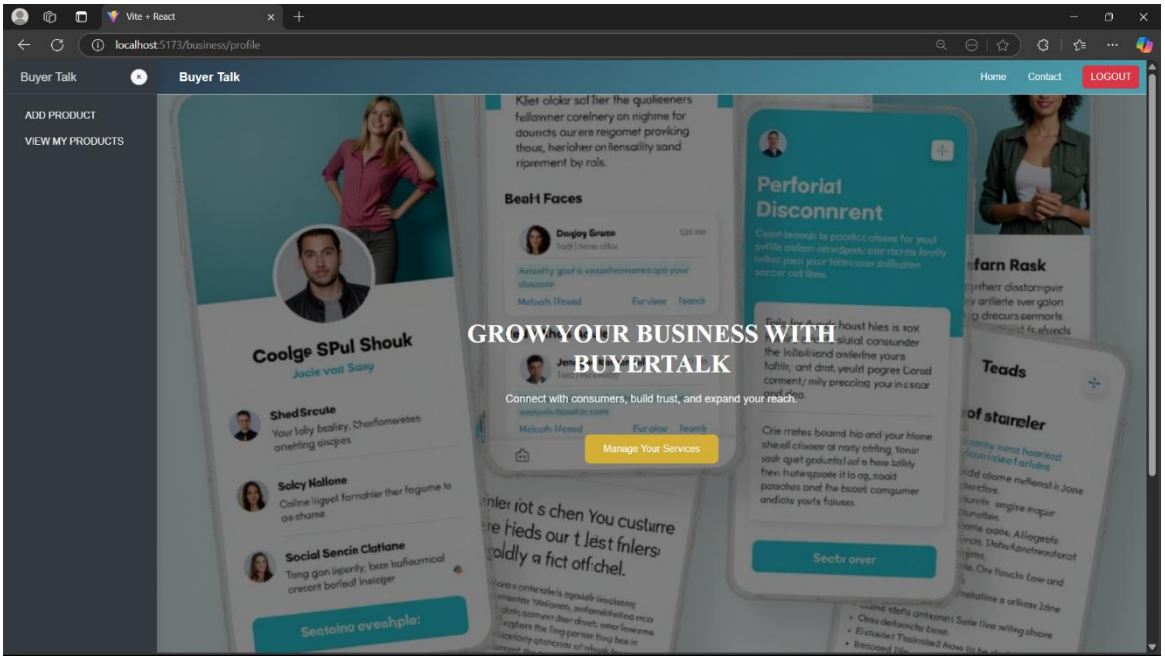
6.1.15 Admin Dashboard



6.1.16 View My Product



6.1.17 User Dashboard



6.1.18 Service Provider Dashboard

CHAPTER 7: CONCLUSION & DISCUSSION

7.1 Conclusion

Conclusion of BuyerTalk Project

The BuyerTalk project successfully achieved its primary objective of creating a transparent and interactive platform for consumers, businesses, and regulatory bodies. By facilitating open discussions, providing a space for complaints and feedback, and enabling businesses to respond effectively, the platform promotes consumer awareness and encourages responsible business practices. The integration of various modules like complaint management, discussion forums, and real-time notifications ensures a seamless and engaging user experience. Additionally, the user-friendly interface and responsive design enhance accessibility across different devices.

Through the implementation of robust security measures, effective database management, and efficient API interactions, the platform demonstrates reliability and scalability. Businesses benefit from valuable customer insights, while regulatory bodies can monitor consumer issues and ensure compliance with relevant laws. Moving forward, BuyerTalk has the potential to expand by incorporating AI-powered analytics for sentiment analysis, chatbot support for faster query resolution, and multilingual support to cater to a diverse user base. This project serves as a stepping stone towards empowering consumers and fostering transparency in the marketplace.

7.2 Summary of Internship Project

During my internship at Grownited Private Limited, I had the opportunity to work on the BuyerTalk project using the MERN stack. This experience allowed me to apply my theoretical knowledge to real-world scenarios, enhancing my skills in frontend and backend development, API integration, and database management. I contributed to key features like user authentication, complaint management, and business response handling. Additionally, I gained hands-on experience in troubleshooting, testing, and ensuring the responsiveness of the platform. The collaborative environment and mentorship from experienced developers further strengthened my problem-solving abilities and understanding of software development processes. This internship has

been a significant step in my professional growth, providing valuable insights into project management and teamwork.

7.3 Problem Encountered and Possible Solutions

1. Delayed Complaint Status Updates

   Users experienced issues where complaint status updates were not reflected in real-time. This was resolved by implementing WebSocket communication for real-time updates and ensuring seamless synchronization between the frontend and backend.

2. API Response Time Lag

   Some API responses took longer than expected due to inefficient database queries. Indexing the frequently accessed fields in MongoDB and optimizing query logic using Mongoose improved the response time significantly.

3. Authentication and Authorization Issues

   There were occasional login failures and incorrect access permissions. This issue was resolved by refining the JWT (JSON Web Token) authentication system and implementing role-based access control to ensure secure and accurate authentication.

4. UI Responsiveness Problems

   Certain pages of the platform experienced layout issues on smaller screens. Implementing responsive design using Flexbox, CSS Grid, and media queries ensured a consistent user experience across devices.

5. Search Functionality Accuracy

   The search feature initially failed to deliver relevant results due to case sensitivity and partial keyword matching. By applying fuzzy search algorithms and normalizing text data, the search functionality was improved for better accuracy.

7.4 Limitation & Future Work

7.4.1 Limitation:

1. Scalability Concerns – The current architecture may face performance issues when handling a large number of users simultaneously.
2. Limited AI Capabilities– The Flask backend is primarily used for API calculations, but it does not store historical user inputs or calculation logs due to the lack of a database.
3. No Multilingual Support – Currently, the platform operates in a single language, restricting accessibility for non-native speakers.
4. Dependency on User Participation – The success of the platform relies heavily on active participation from both consumers and businesses, making it less effective in areas with low engagement.
5. Limited Regulatory Integration – While regulatory bodies can monitor complaints, automated legal enforcement mechanisms are not implemented yet.

7.4.2 Future Work:

1. AI-Powered Resolution Suggestions – Implement AI algorithms to suggest possible complaint resolutions based on historical data.
2. Multilingual Support – Expand language options to cater to a diverse user base.
3. Advanced Analytics Dashboard – Provide businesses and regulators with in-depth analytics and reports for better decision-making.
4. Enhanced Mobile App – Develop a dedicated mobile app to provide users with a more optimized and interactive experience.
5. Third-Party API Integration – Collaborate with payment gateways and e-commerce platforms for seamless dispute resolution processes.

CHAPTER 8: REFERENCES

8.1 Bibliography

1. Official Documentation

    - Node.js: https://nodejs.org/docs
    - React.js: https://react.dev/
    - Bootstrap CSS: https://getbootstrap.com/
    - Mongo DB: https://www.mongodb.com/docs
    - Express js: https://expressjs.com/

2. Online Resources & Tutorials

    - MDN Web Docs (HTML, CSS, JavaScript): https://developer.mozilla.org
    - W3Schools (Frontend Development): https://www.w3schools.com
    - GeeksforGeeks (Python & Flask): https://www.geeksforgeeks.org
    - FreeCodeCamp (Web Development Tutorials): https://www.freecodecamp.org

3. Research Papers & Books

    - "JavaScript: The Good Parts" by Douglas Crockford
    - "Learning React" by Alex Banks and Eve Porcello

4. Stack Overflow & Developer Forums

    - Solutions for debugging and optimizing code were taken from https://stackoverflow.com and developer communities.