**PROJECT REPORT**

**1. INTRODUCTION**

**1.1 Project Overview**

**ResolveNow** is a MERN stack–based web application designed to revolutionize the way organizations handle public grievances and internal complaints. By streamlining the complaint registration, tracking, and resolution workflow, the platform brings structure, transparency, and speed to the process of addressing user concerns. Whether the users are citizens reporting local issues or employees flagging operational hurdles, ResolveNow ensures they are heard and responded to promptly.

Built with a modern, modular architecture, ResolveNow supports seamless interactions across different roles—users, administrators, and resolvers—delivering a full-circle experience that ranges from complaint submission to feedback collection. Its dynamic dashboard, real-time notifications, and role-based access make the platform ideal for institutions seeking to improve accountability, engagement, and service delivery.

**1.2 Purpose**

The purpose of the **ResolveNow** project is to develop a digital platform for structured complaint management, enabling organizations and communities to foster trust and operational efficiency through transparent issue resolution. The system aims to:

* Eliminate delays and inefficiencies of manual complaint handling
* Guide users in submitting detailed and categorized complaints
* Notify users with real-time updates on complaint progress
* Empower admins to assign, filter, and resolve complaints efficiently
* Enable user feedback to improve overall service quality
* Build a secure, scalable, and user-friendly solution using modern web technologies

**2. IDEATION PHASE**

**2.1 Problem Statement**

**Customer Problem Statement:**



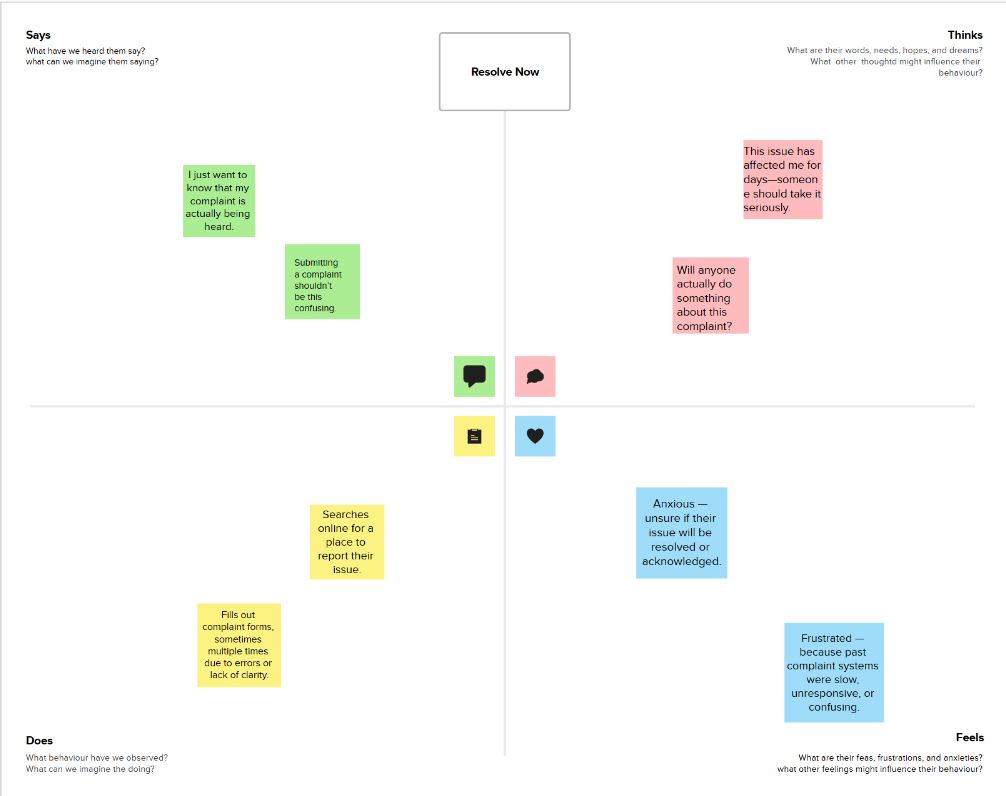
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PS** | **I am (Customer)** | **I’m trying to** | **But** | **Because** | **Which makes me feel** |
| PS-1 | a college student dealing with infrastructure issues on campus | submit a complaint easily and see quick results | the complaint process is slow or unclear | there's no transparent tracking or updates | frustrated and unheard about recurring problems |
| PS-2 | a working professional in a public institution | report technical malfunctions without leaving my desk | I don’t know who’s responsible | there’s no centralized system for filing or following up | disengaged and forced to tolerate inefficiencies |
| PS-3 | a community resident dealing with civic problems | file complaints like garbage collection or water leakage | my past complaints were ignored or unresolved | there’s no feedback mechanism or response accountability | discouraged and hesitant to complain again |
| PS-4 | an admin trying to manage city-wide complaints | assign and resolve issues systematically | data is scattered and complaints aren’t categorized properly | we lack tools to filter, prioritize, and track complaints efficiently | overwhelmed and reactive instead of proactive |

**2.2 Empathy Map Canvas**

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.



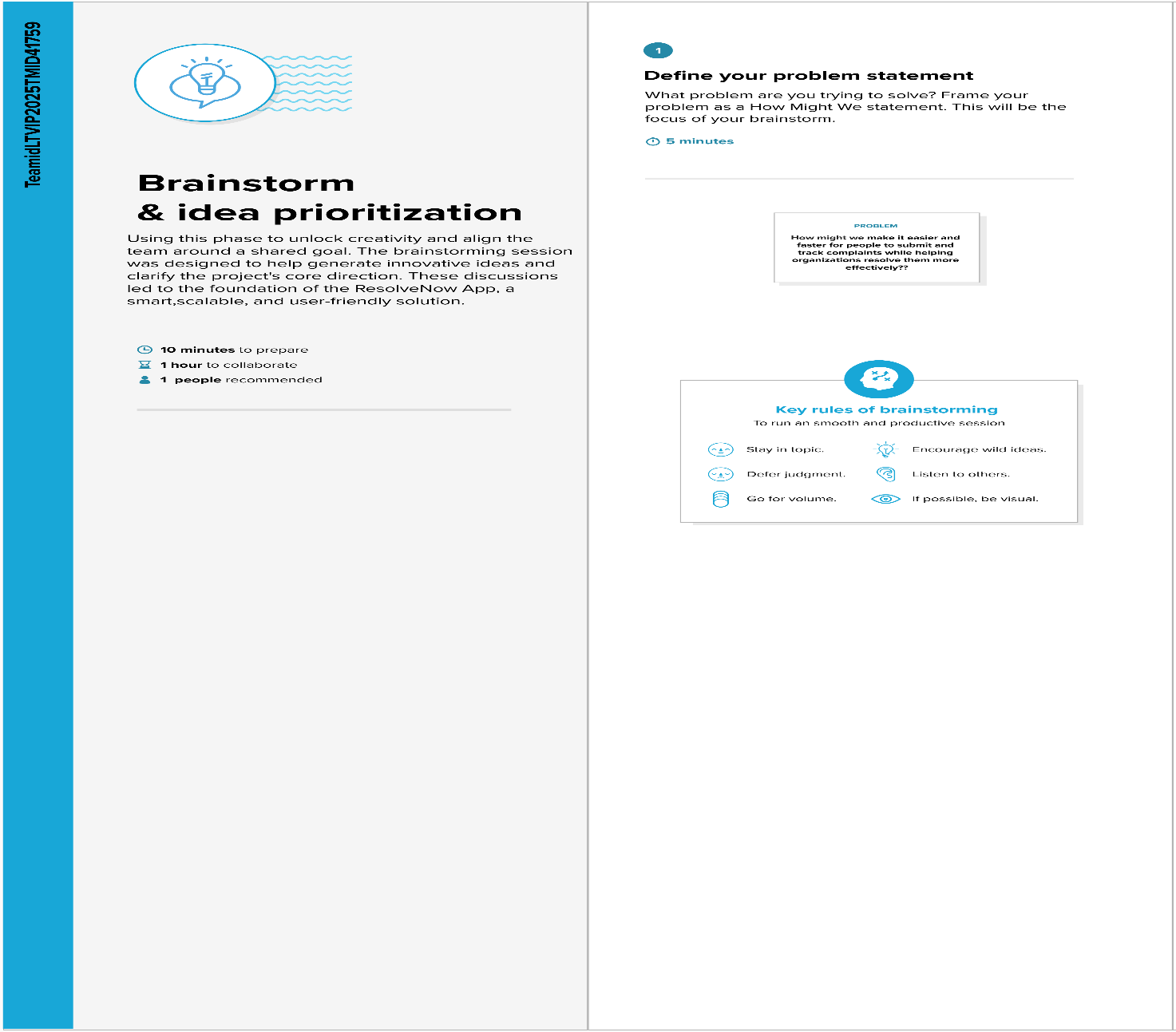
**2.3 Brainstorming**

**Brainstorm & Idea Prioritization:**

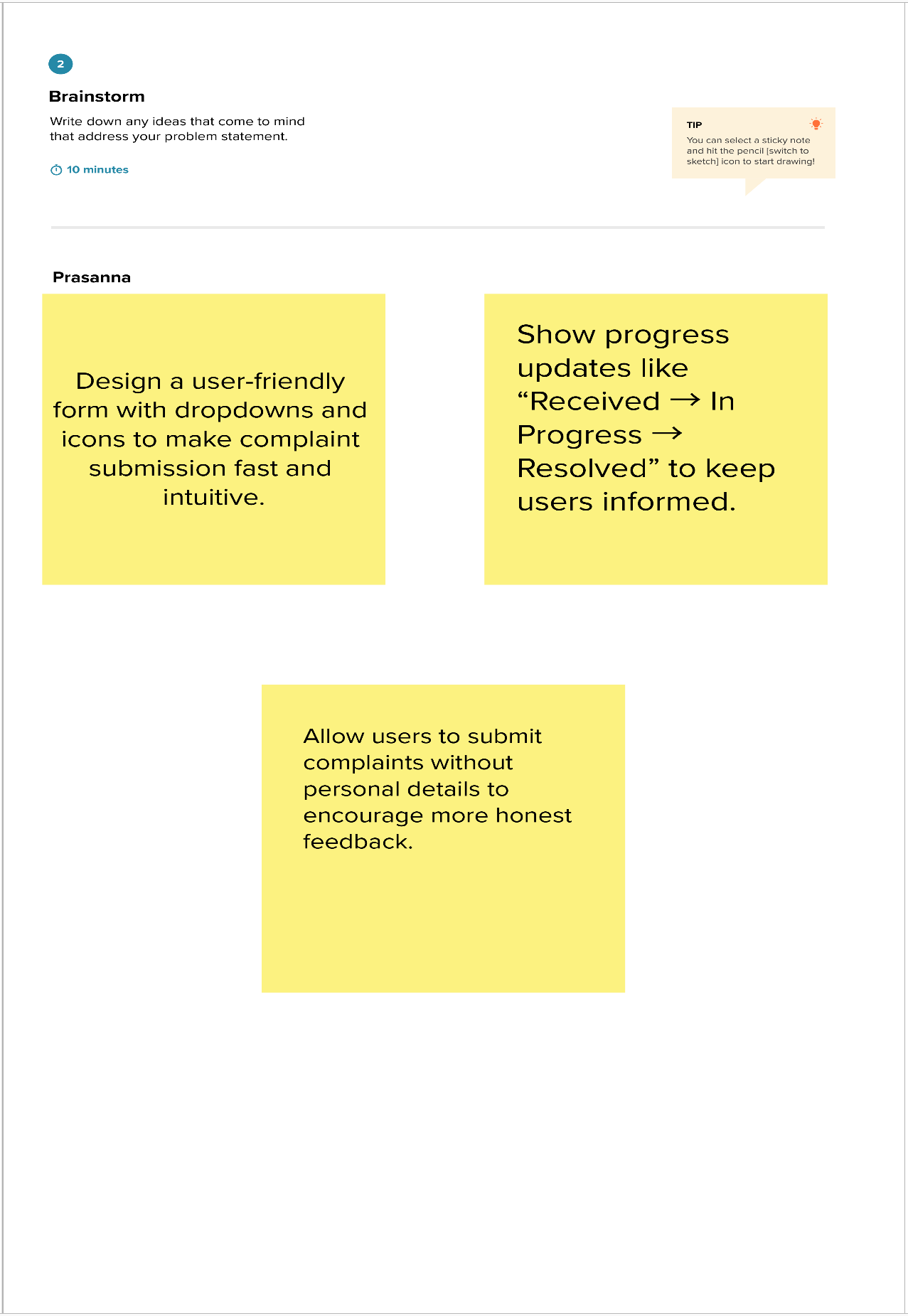
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Link: <https://app.mural.co/t/bookorder0417/m/bookorder0417/1750770438520/1047a88635c51a632439c81f8caa682712057c94?sender=u427be78f24cc5d79ea230073>

**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

****

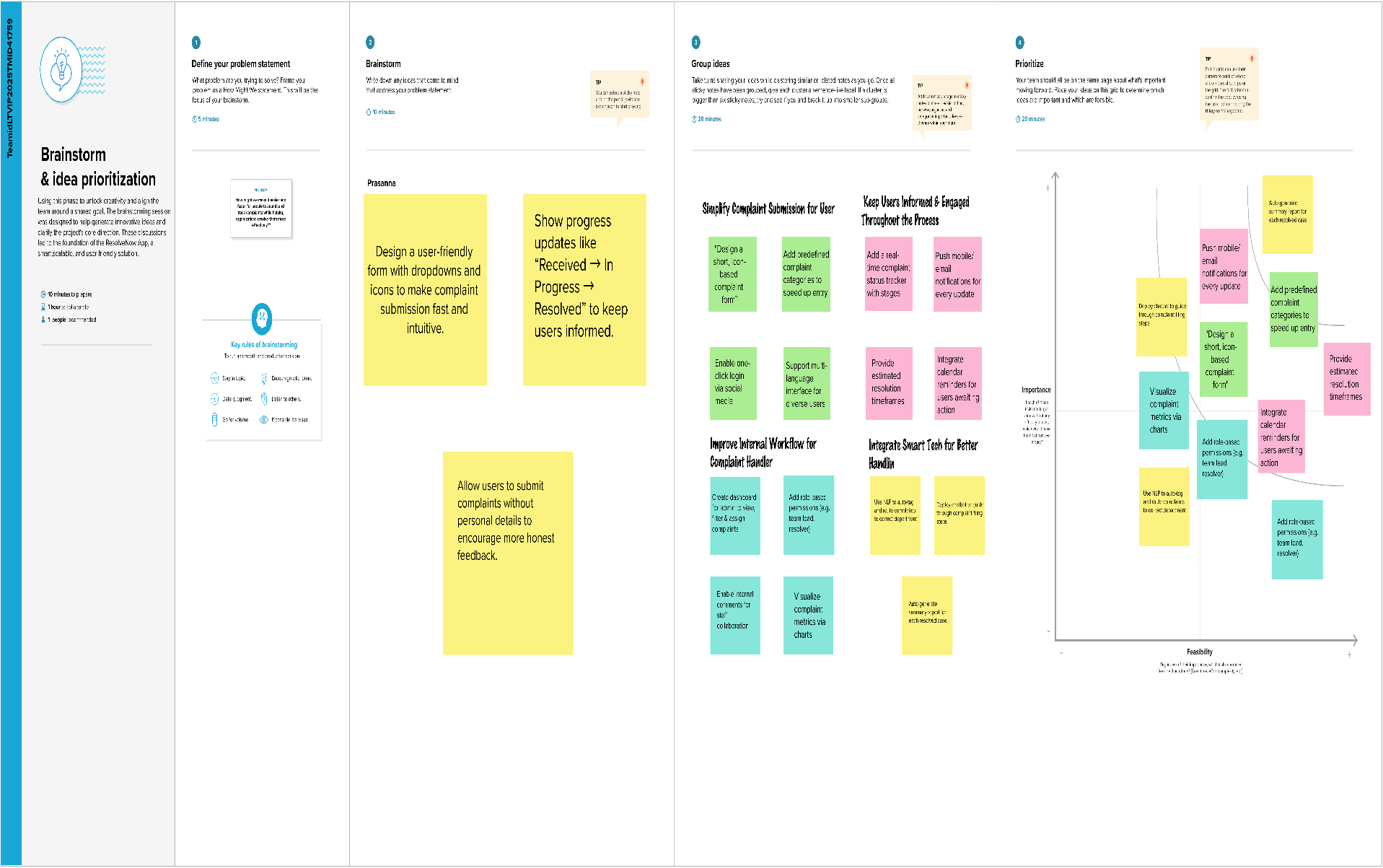
**Step-2: Brainstorm, Idea Listing and Grouping**

****



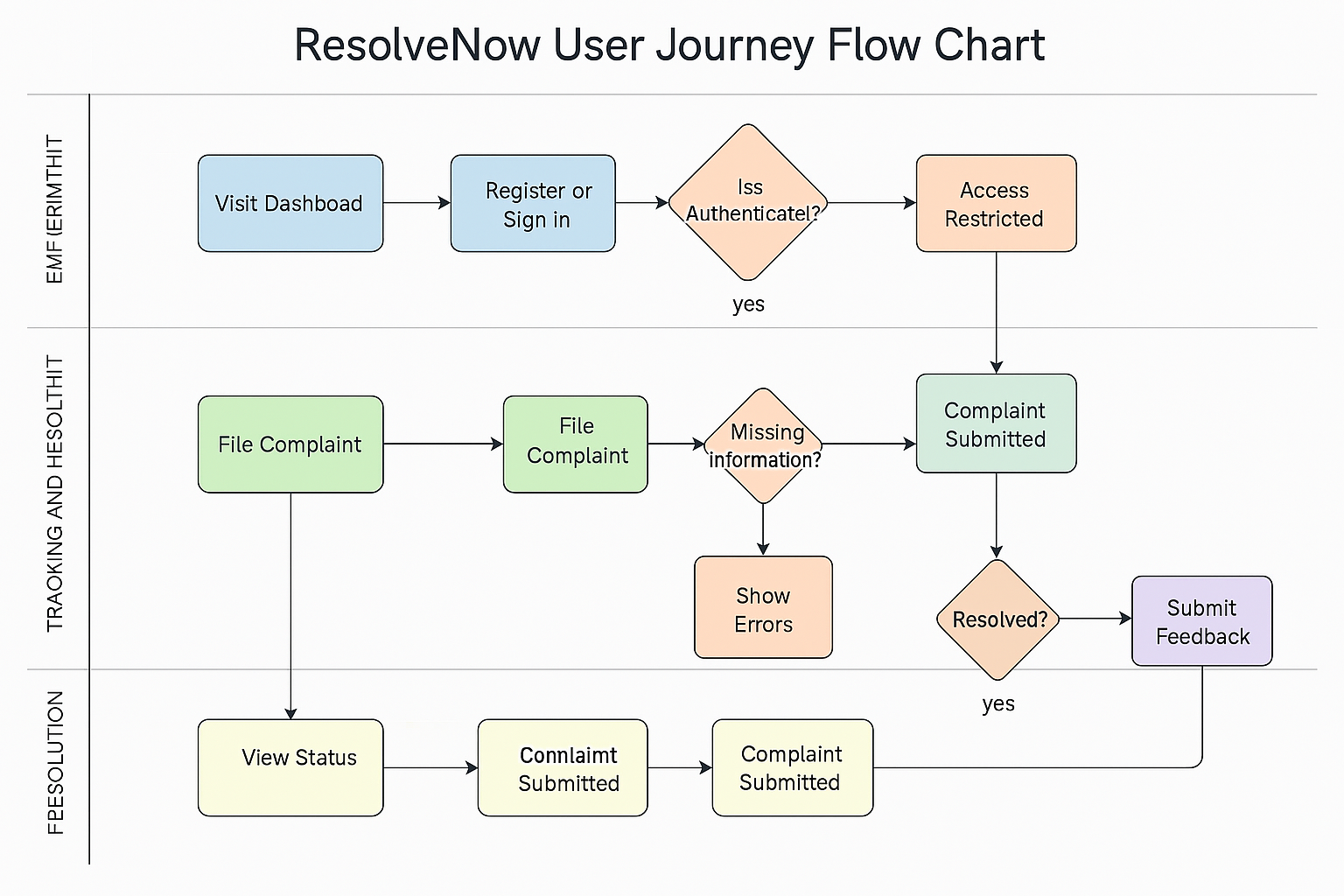
**Step-3: Idea Prioritization**





**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey map**



**3.2 Solution Requirement**

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

**Functional Requirements –Resolve Now App**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Authentication | Sign up, Login, Password Reset, OAuth Login via Google/GitHub |
| FR-2 | Complaint Submission | Submit complaint with category, description, optional attachments |
| FR-3 | Complaint Tracking | View real-time complaint status, progress timeline, estimated resolution date |
| FR-4 | Admin Dashboard | Manage users, assign complaints, update statuses, add internal notes, view complaint metrics |
| FR-5 | Notifications | Automated email/in-app alerts for status updates, resolution confirmations |
| FR-6 | Feedback System | Post-resolution rating system, optional user comments, display summary to admin |
| FR-7 | Role-Based Access | Separate views and permissions for user, admin, reviewer |
| FR-8 | Analytics & Reports | View reports by category, status, department, or time period |
| FR-9 | UI/UX & Accessibility | Mobile responsiveness, language options, accessible design with tooltips and error handling |
| FR-10 | Security & Data Protection | Data encryption, input sanitization, secure API endpoints, login rate-limiting |

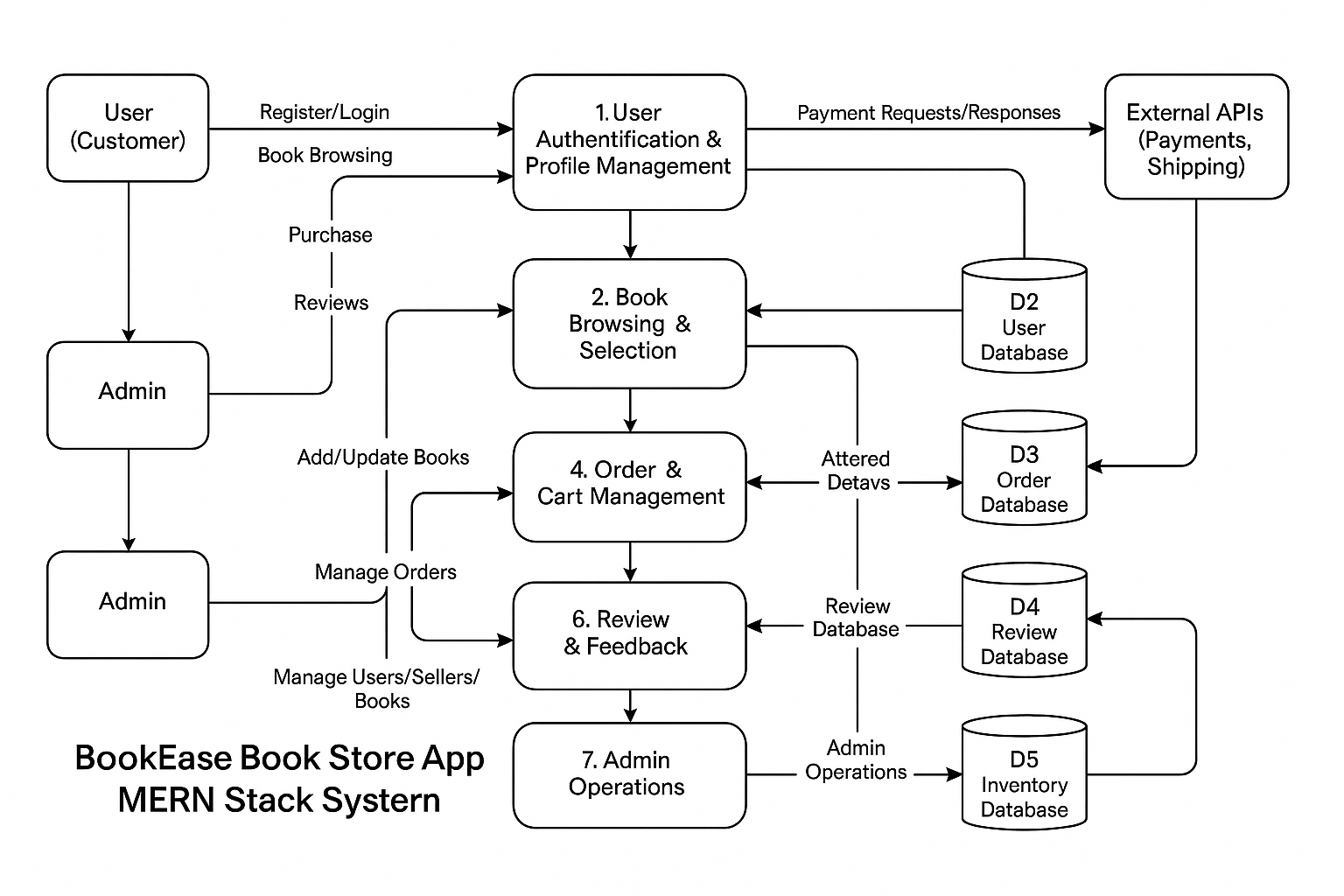
**Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **NFR No.** | **Requirement** | **Description** |
| **NFR-1** | **Usability** | UI should be intuitive and user-friendly for buyers, sellers, and admins |
| **NFR-2** | **Security** | All user and transaction data must be encrypted; implement role-based access and JWT auth |
| **NFR-3** | **Reliability** | All services including browsing, checkout, and notifications must be consistently available |
| **NFR-4** | **Performance** | Pages should load within 2 seconds; checkout and actions should respond instantly |
| **NFR-5** | **Availability** | Ensure at least 99.9% uptime with backup and recovery plans |
| **NFR-6** | **Scalability** | Must support growth in users, inventory, and orders without performance degradation |

**3.3 Data Flow Diagram**

**Data Flow Diagrams:**

****A **Data Flow Diagram (DFD)** illustrates how data moves within the Freelance Finder platform. It captures how users (freelancers and clients) interact with the system, how information flows between different components, and where the data is stored.

Example: DFD Level 0 (Industry Standard)

**User Story Table – Book Nest App**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement (Epic)** | **User Story / Task** | **Acceptance Criteria** | **Priority** | **Release** |
| User (Complainant) | Account Management | As a user, I can register and log in securely. | Account is created, login/logout works correctly. | High | Sprint-1 |
|  | Complaint Submission | As a user, I can submit complaints by category and description. | Complaint is filed with confirmation shown. | High | Sprint-1 |
|  | Complaint Tracking | As a user, I can track the progress and status of my complaints in real time. | Status is accurate and timeline is shown. | High | Sprint-2 |
|  | Notifications | As a user, I receive alerts when my complaint status is updated. | Email or in-app notifications are triggered on each status update. | High | Sprint-2 |
|  | Feedback System | As a user, I can rate the service after resolution. | Feedback is successfully submitted and stored. | Medium | Sprint-2 |
| Admin | Admin Dashboard | As an admin, I can view, assign, and manage user complaints. | Admin can update complaint statuses and assign them to resolvers. | High | Sprint-1 |
|  | User Management | As an admin, I can manage user accounts and control role access. | Admin can activate, deactivate, and edit user roles. | High | Sprint-1 |
|  | Feedback Moderation | As an admin, I can review and analyze complaint feedback. | Feedback is displayed with admin options to mark as addressed. | Medium | Sprint-2 |
|  | Complaint Analytics | As an admin, I can view visual reports by category, status, and performance. | Dashboard loads with correct charts and filters. | High | Sprint-2 |
| Resolver (Staff) | Complaint Assignment | As a resolver, I can see complaints assigned to me. | Only assigned complaints are visible to the resolver. | High | Sprint-1 |
|  | Complaint Resolution | As a resolver, I can update complaint status and add internal comments. | Status updates reflect for users with history retained. | High | Sprint-2 |
|  | Profile Management | As a resolver, I can view and update my contact and role information. | Changes to profile are saved and visible only to authorized users. | Medium | Sprint-2 |

**3.4 Technology Stack**

**Technical Architecture:**

**Book Nest** is designed with a scalable 3-tier architecture consisting of:  
● **Presentation Layer (Frontend):** User-friendly interface for users and sellers to browse and purchase books.  
**● Business Logic Layer (Backend):** Handles user authentication, book management, cart, orders, and notifications.  
● **Data Storage Layer:** Secure storage of user profiles, books, reviews, orders, and transaction records.  
The platform integrates with third-party APIs for notifications (SMS/email), payment processing, and book recommendation engines to enhance usability.

**Table-1: Components & Technologies:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Component** | **Description Technology** |
| **1.** | User Interface | Web and mobile-friendly interface for book browsing and purchases HTML, CSS, JavaScript / React.js |
| **2.** | Application Logic-1 | Book browsing, filtering, cart management, and order placing Node.js, Express.js |
| **3.** | Application Logic-2 | Admin and Seller panels, inventory and user management React.js, Node.js |
| **4.** | Database | Stores user profiles, books, orders, and reviews MongoDB |

**Table-2: Application Characteristics:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Characteristics** | **Description Technology** |
| **5.** | Open-Source Frameworks | Frontend frameworks React.js, Node.js, Bootstrap, Tailwind CSS |
| **6.** | Scalable Architecture | 3-tier architecture with RESTful APIs Microservices |

**References:**

[**React.js Documentation**](https://react.dev/)

[**Node js Best Practice**](https://nodejs.org/en/learn/getting-started/introduction-to-nodejs)

[**JSON Web Server Referance**](https://www.npmjs.com/package/json-server)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)

[MongoDB Docs](https://www.mongodb.com/docs/)

**4. PROJECT DESIGN**

**4.1 Problem Solution Fit**

**Problem–Solution Fit Overview**

The Problem–Solution Fit ensures that **Resolve Now** addresses the challenges faced by individuals and organizations in filing, tracking, and resolving complaints. This alignment validates that the system meaningfully solves real user and operational pain points.

**Purpose**

* Provide a centralized digital platform for users to file and track complaints effortlessly.
* Streamline complaint resolution workflows for organizations to act quickly and transparently.
* Boost user satisfaction through timely updates, clear communication, and accountability.
* Empower administrators with insights, analytics, and structured complaint handling tools.

**Problem Statement**

Both users and organizations face serious gaps in existing complaint handling systems:

* Users don’t know where or how to raise issues clearly and securely.
* Lack of transparency after submission creates frustration and distrust.
* Traditional systems are slow, paper-based, or prone to data loss.
* Organizations struggle with unstructured data, scattered communications, and lack of visibility.
* Feedback loops are weak or non-existent, leading to repeated issues.

**Solution**

Resolve Now solves these challenges through a digital-first complaint ecosystem:

* 📝 Complaint Submission Form: Clear, guided interface for filing structured complaints.
* 🔄 Real-Time Tracking: Users stay updated with live status changes and complaint timelines.
* 📊 Admin Dashboard: For assigning, sorting, and managing complaints across teams.
* 📬 Automated Notifications: Email or in-app alerts at every key stage of complaint handling.
* 🌟 Feedback System: Close the loop with user ratings and insights for continual improvement.
* 🔐 Role-Based Access: Secure, personalized views for users, admins, and reviewers.

**4.2 Proposed Solution**

**Proposed Solution for ResolveNow App**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Parameter** | **Description** |
| **1** | **Problem Statement (Problem to be solved)** | **Individuals face difficulty submitting, tracking, and getting updates on complaints due to slow, opaque, and fragmented systems. Organizations struggle with managing, assigning, and resolving complaints efficiently and transparently.** |
| **2** | **Idea / Solution Description** | **Resolve Now is a MERN-stack web application that allows users to submit and monitor complaints in real-time. It offers streamlined submission forms, a real-time tracking interface, admin dashboard for resolution, and a feedback mechanism.** |
| **3** | **Novelty / Uniqueness** | **• Real-time complaint status tracking  • Role-based dashboards for users and admins  • Automated notifications and feedback module  • Smart categorization with tags  • Mobile-responsive and accessible UI** |
| **4** | **Social Impact / Customer Satisfaction** | **• Empowers users to voice concerns conveniently and confidently  • Builds trust through transparency and faster resolutions  • Helps institutions handle complaints ethically and efficiently  • Reduces delays and communication gaps** |
| **5** | **Business Model (Revenue Model)** | **• Freemium plan for small institutions or teams  • Subscription tiers for government, education, or enterprise use  • Premium features: analytics, integrations, chatbot support  • B2B model for municipalities or public utilities** |
| **6** | **Scalability of the Solution** | **• Modular and scalable with clean MERN architecture  • Ready for multilingual, time zone-sensitive deployment  • Cloud-hosted with elastic database scaling  • API-ready for mobile apps or integration with existing systems** |

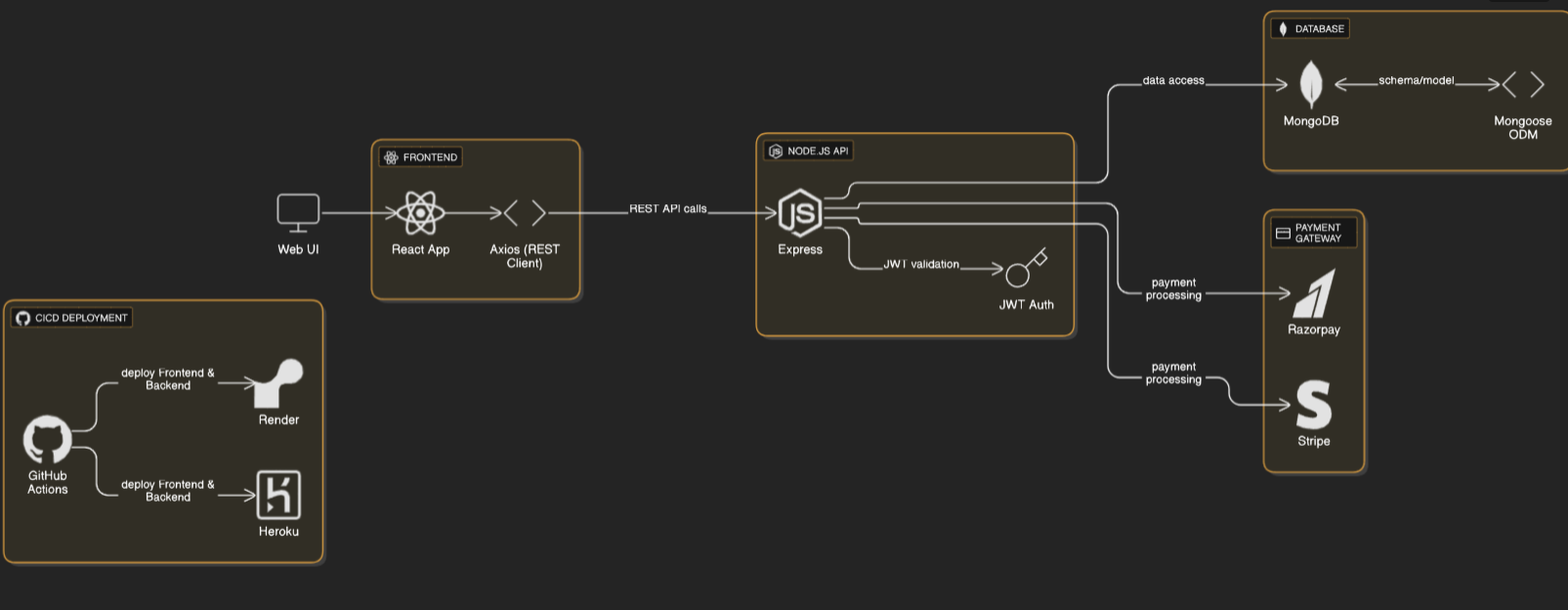
**4.3 Solution Architecture**

**Solution Architecture:**

To deliver an intuitive, secure, and scalable complaint management platform that empowers users to voice concerns and enables institutions to respond with speed, structure, and transparency.Effortless Book Browsing and Discovery

* **Seamless Complaint Submission and Categorization**
* End-to-End Complaint Lifecycle Management
* Scalable Role-Based Access and Authentication

**Example - Solution Architecture Diagram:**

****

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

**Product Backlog & Sprint Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| **Sprint-1** | **User Registration & Authentication** | **USN-1** | **As a user, I can register and log in securely.** | **3** | **High** | **G. Prasanna** |
|  |  | **USN-2** | **As a user, I can reset my password.** | **2** | **Medium** | **G. Prasanna** |
| **Sprint-2** | **Complaint Management** | **USN-3** | **As a user, I can submit a categorized complaint via a simple form.** | **3** | **High** | **G. Prasanna** |
|  |  | **USN-4** | **As a user, I can view and track the real-time status of my complaint.** | **3** | **High** | **G. Prasanna** |
| **Sprint-3** | **Admin Dashboard & Workflow** | **USN-5** | **As an admin, I can assign, filter, and update complaint statuses.** | **4** | **High** | **G. Prasanna** |
|  |  | **USN-6** | **As an admin, I can view analytics and complaint history.** | **2** | **Medium** | **G. Prasanna** |
| **Sprint-4** | **Notifications & Feedback** | **USN-7** | **As a user, I receive notifications for complaint updates.** | **2** | **High** | **G. Prasanna** |
|  |  | **USN-8** | **As a user, I can provide feedback and rate the resolution.** | **2** | **Medium** | **G. Prasanna** |
|  |  | **USN-9** | **As an admin, I can review feedback and use it to improve service quality.** | **2** | **Medium** | **G. Prasanna** |

**Project Tracker, Velocity & Burndown Chart**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| **Sprint-1** | 20 | 6 Days | 15 June 2025 | 20 June 2025 | 20 | 20 June 2025 |
| **Sprint-2** | 20 | 6 Days | 17 June 2025 | 22 June 2025 | 20 | 22 June 2025 |
| **Sprint-3** | 20 | 6 Days | 19 June 2025 | 24 June 2025 | 20 | 24 June 2025 |
| **Sprint-4** | 20 | 6 Days | 21 June 2025 | 26 June  2025 | 20 | 26 June  2025 |

**6. FUNCTIONAL AND PERFORMANCE TESTING**

**6.1 Performance Testing**

**Project Overview:**

**Project Name:** Resolve Now  
**Project Description:** designed to simplify and modernize the way individuals and organizations register, manage, and resolve complaints. It provides a centralized digital platform where users can submit complaints, track their status in real time, and receive timely updates—all while helping organizations streamline their resolution workflows.

🔧 **Key Features:**

* **User Registration & Login** – Secure account creation and access to track complaints.
* **Complaint Submission Form** – Easy-to-use, categorized submission interface.
* **Real-Time Complaint Tracking** – Status updates with timelines and notifications.
* **Admin Dashboard** – For organizations to manage, assign, and resolve complaints efficiently.
* **Feedback Mechanism** – Post-resolution satisfaction rating from users.

💻 **Tech Stack:**

* **Frontend:** HTML, CSS, JavaScript, Bootstrap, React.js
* **Backend:** Node.js, Express.js
* **Database:** MongoDB

Let me know if you'd like me to tweak this for a résumé, GitHub README, or even turn it into a visual one-pager!

**Project Version:** v1.0.0  
**Testing Period:** 2025-06-15 to 2025-06-28  
**Testing Scope:**

* User registration and login
* 📍 **Complaint Tracking:** Status updates, timestamps, progress indicators
* 🛠️ **Admin Features:** Dashboard view, complaint assignment, status editing
* 📬 **Notifications:** Email or in-app updates when complaint status changes
* 📊 **Feedback Module:** Submit rating, store user comments
* Data encryption and HTTPS enforcement.
* Meaningful error messages and confirmation alerts.

#### **Requirements to be Tested:**

* Verify user can register with valid and invalid details.
* Test login functionality with correct and incorrect credentials.
* Ensure password reset via email works with proper token validation.
* Validate session handling and secure logout.
* Check if users can submit complaints with all required fields.
* Validate category selection, attachments, and input formats.
* Ensure form handles invalid, incomplete, or duplicate submissions.

**Credentials:**

**User:** test.user@example.com / user123

* **Seller:** test.seller@example.com / seller123
* **Admin:** admin@docspot.com / admin@12345

**Test Cases:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Module | Test Scenario | Steps to Execute | Expected Result | Priority |
| TC001 | **User Registration** | **Register with valid inputs** | **Enter valid name, email, password and submit** | **User account is created and confirmation message is shown** | **High** |
| TC002 | **User Registration** | **Register with missing mandatory fields** | **Leave fields blank and submit** | **Error messages are shown for each required field** | **High** |
| TC003 | **Login** | **Login with valid credentials** | **Enter correct email and password** | **Redirected to user dashboard** | **High** |
| TC004 | **Login** | **Login with invalid credentials** | **Enter wrong email/password** | **Error message: “Invalid credentials”** | **High** |
| TC005 | **Complaint Submission** | **Submit complaint with all fields** | **Fill form with details, select category, and submit** | **Complaint is saved and confirmation message appears** | **High** |
| TC006 | **Complaint Submission** | **Submit complaint with invalid/missing fields** | **Leave category blank and submit** | **Form shows validation errors** | **High** |
| TC007 | **Complaint Tracking** | **View status updates** | **Go to “My Complaints” and check status timeline** | **Accurate status updates are shown** | **High** |
| TC008 | **Admin Dashboard** | **Assign complaint to resolver** | **Admin logs in, views complaint, assigns to user** | **Assignee receives notification and complaint is updated** | **High** |
| TC009 | **Feedback Submission** | **Submit feedback after complaint is resolved** | **User rates experience and submits comment** | **Feedback is saved and visible in admin panel** | **Medium** |
| TC010 | **Notifications** | **Receive update alert when complaint status changes** | **Submit a complaint and simulate admin changing status** | **Email/in-app notification is received** | **High** |
| TC011 | **Security** | **Unauthorized user accesses admin route** | **Try visiting /admin/dashboard as a regular user** | **Access denied with redirect or warning message** | **High** |
| TC012 | **Performance** | **Submit complaint under load** | **Simulate multiple users submitting complaints simultaneously** | **Server handles load without crashing or slowing down** | **Medium** |

**Bug Tracking:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bug ID | Module | Bug Description | Steps to Reproduce | Severity | Status | Reported By | Assigned To | Date Logged |
| BUG-001 | **Login/Registration** | **Login fails with correct credentials** | **Register → Logout → Try to log in again using same credentials** | **High** | **Open** | **Tester A** | **Dev A** | **01-Jul-2025** |

**Sign-off:**

Tester Name: G. Prasanna

Date: 28-06-2025

Signature: G. Prasanna

**Notes:**

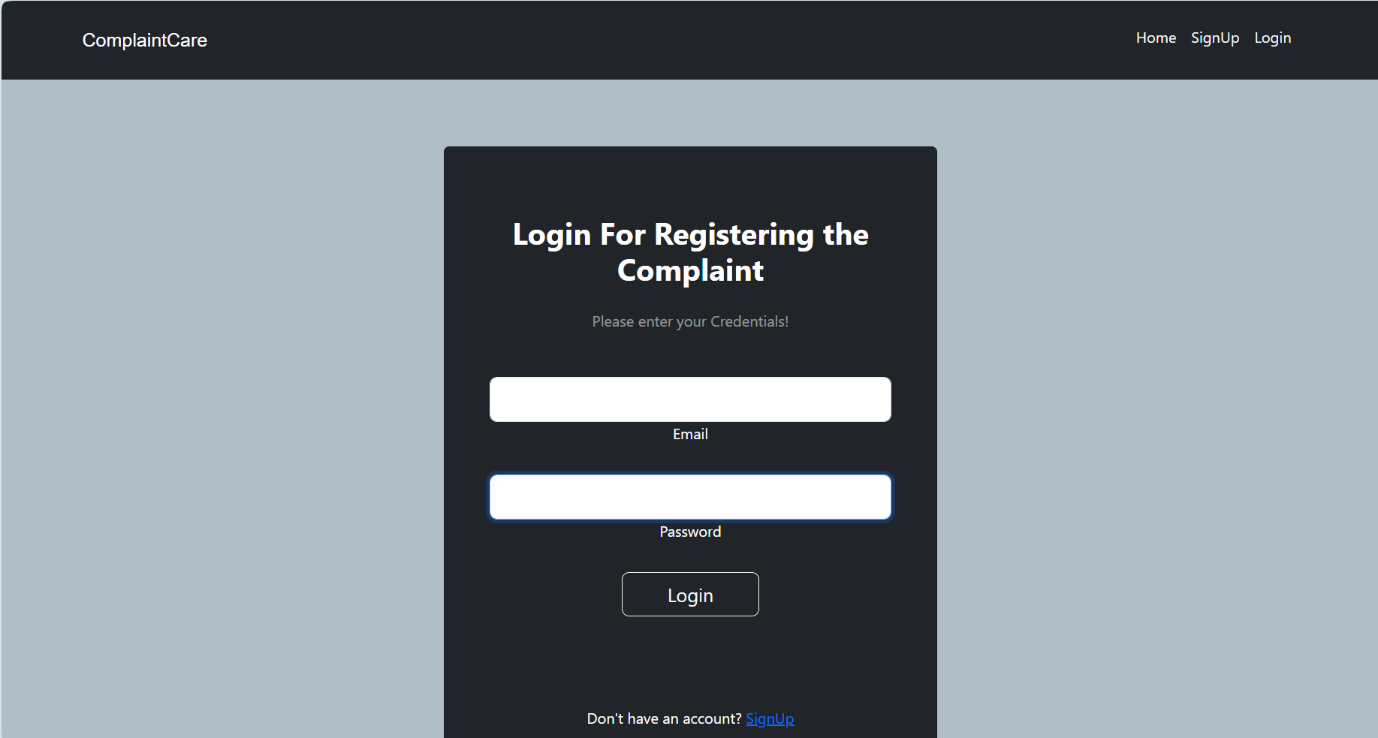
* Functional testing of user flows, form handling, and status changes
* Security testing for data access, encryption, and role control
* Usability testing for smooth navigation and mobile responsiveness

Feedback system validation and notification reliability

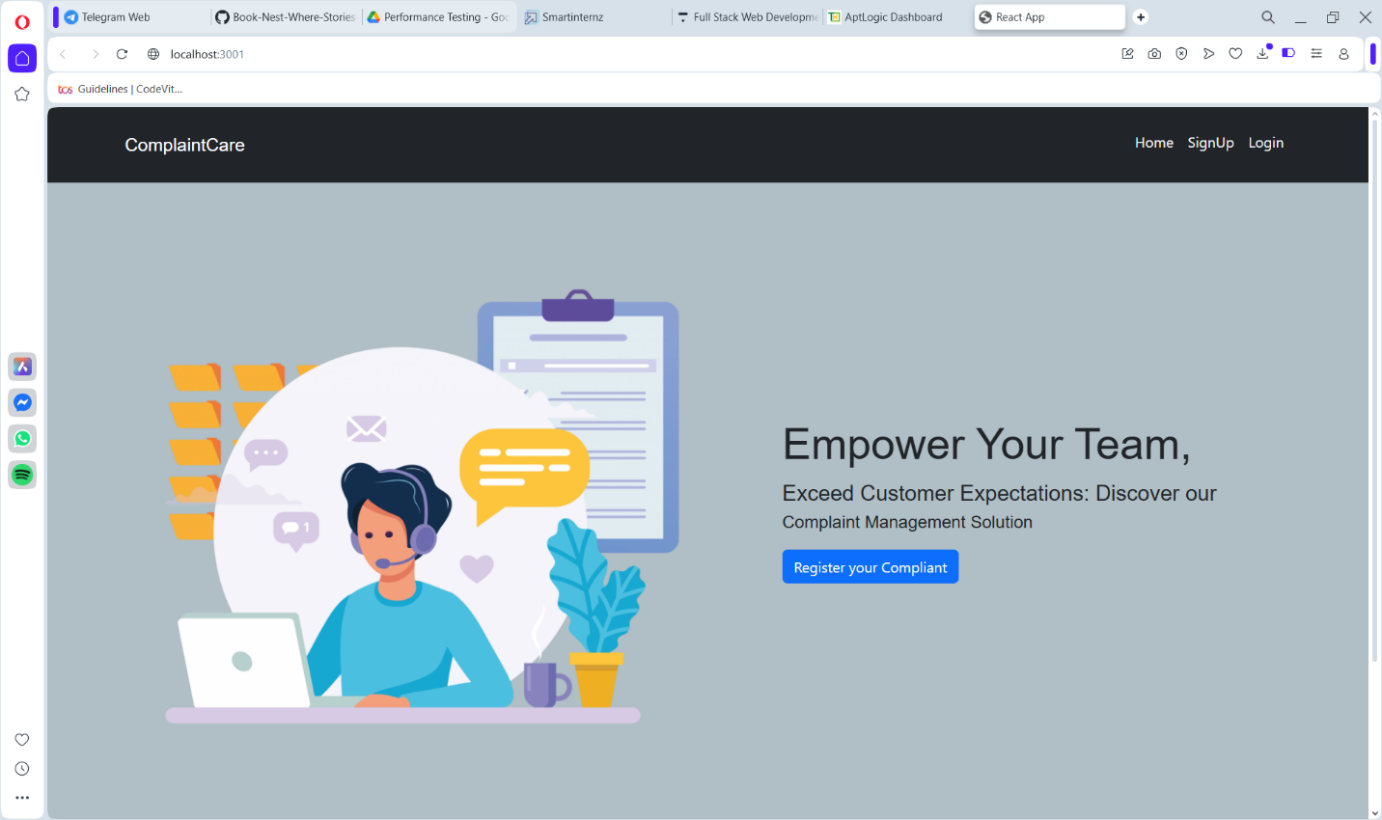
**7. RESULTS**

**7.1 Output Screenshots**

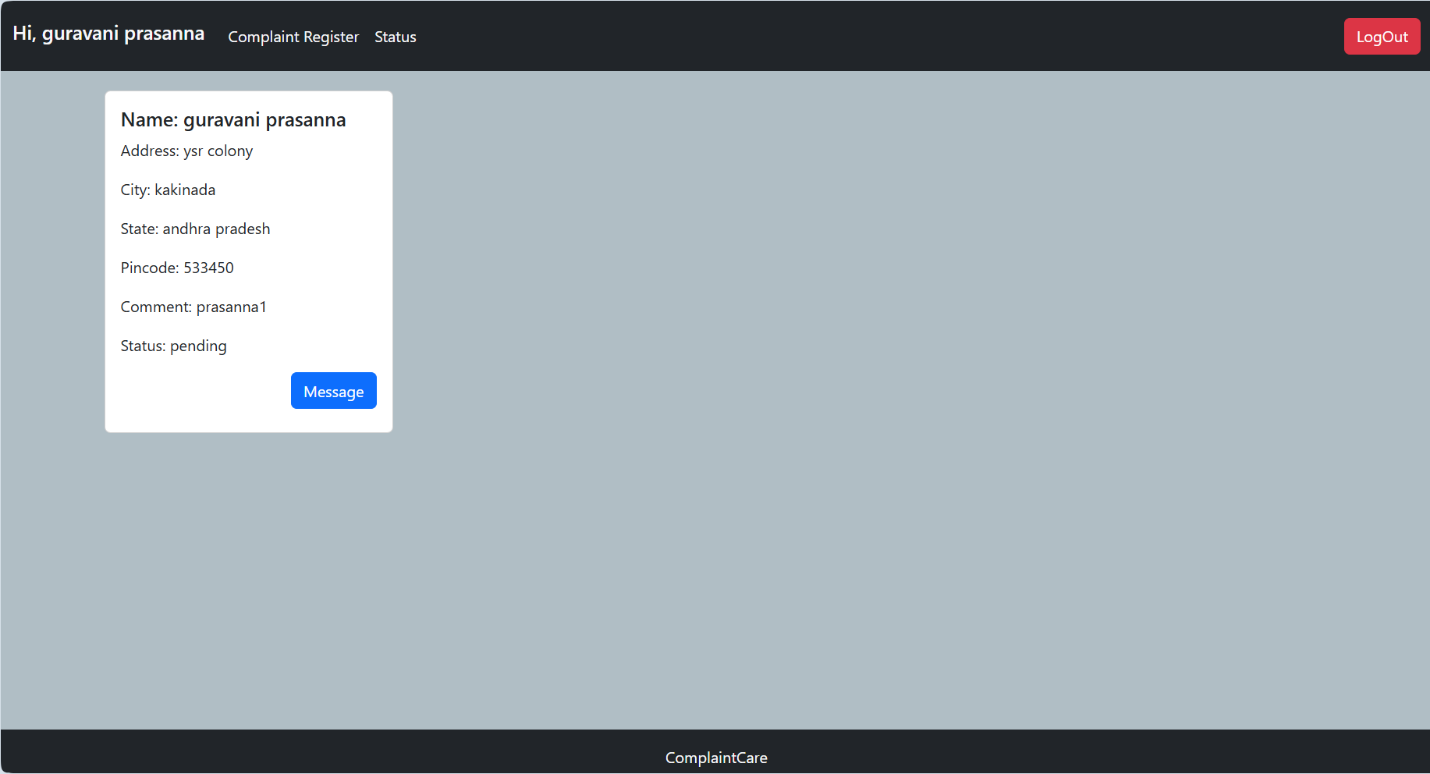
**Login Page:**



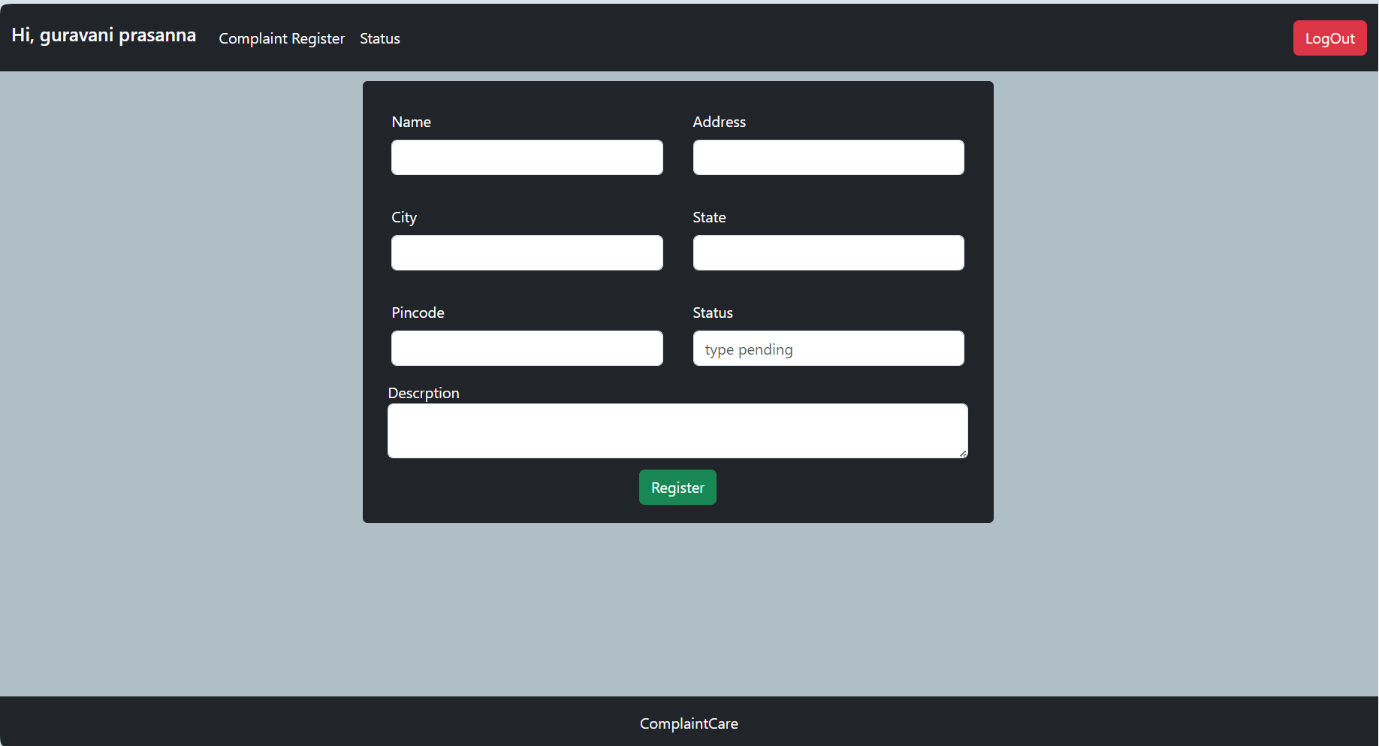
**Home Page:**



**Complaints registered & status**



**Complaint registration**



**8. ADVANTAGES & DISADVANTAGES**

**Advantages of Resolve Now:**

* **Clear Complaint Workflow:** End-to-end visibility and tracking.
* **Instant Notifications**: Keeps users and admins updated in real time.
* **Role-Based Security:** Ensures data protection and proper access.
* **Centralized Records:** All data is stored, traceable, and audit-ready.
* **Analytics Support:** Helps teams monitor performance and trends.
* **Scalable & Web-Accessible:** Works across devices for all users.
* **Feedback Loop:** Improves service based on user input.
* **Easy Integrations:** Supports APIs for alerts and third-party systems.

**Disadvantages of Resolve Now Project**

* **Internet Dependent:** Offline access is limited.
* **Learning Curve:** May require onboarding for new users.
* **Manual Assignment Needed:** Triage is not yet automated.
* **Admin Overload Risk:** High volume may overwhelm single admin.
* **Underused Feedback:** May not always lead to action if not reviewed.
* **Mobile Optimization Gaps:** Needs tuning for older mobile browsers.
* **State Management Overhead**: Managing global state (e.g., with Redux) can become complex as the app grows.

**9. CONCLUSION**

**ResolveNow** stands as a transformative solution in the realm of complaint management, bridging the gap between public feedback and institutional action through technology. By offering a centralized platform for registering, tracking, and resolving grievances, it cultivates transparency, boosts accountability, and fosters trust between users and service providers.

Powered by the MERN stack, ResolveNow ensures scalability, responsiveness, and robust performance, making it suitable for deployment in educational institutions, municipal bodies, and enterprise environments alike. Its role-based structure supports seamless collaboration among citizens, administrators, and resolution teams, while features like real-time notifications, feedback integration, and analytics empower organizations to not just respond—but improve.

.

**10. FUTURE SCOPE**

**Future Scope – ResolveNow**

* **AI-Powered Complaint Triage**Incorporating machine learning models to auto-categorize and prioritize complaints based on urgency, sentiment, and past trends for faster assignments.
* **Multilingual & Voice Support**Enabling complaint submission and system interaction in regional languages and voice-based input for enhanced accessibility, especially in diverse communities.
* **Mobile App Development**Launching dedicated Android and iOS apps to provide offline support, geo-tagging capabilities, and push notifications for instant user updates.
* **Chatbot Integration**  
  Deploying an intelligent chatbot to assist users during complaint submission, answer FAQs, and provide real-time guidance 24/7.
* **Predictive Analytics for Admins**  
  Introducing dashboards that forecast complaint trends, detect service gaps, and recommend preventive maintenance using historical data.
* **Government API Integration**  
  Connecting with municipal systems, educational boards, or civic platforms to enable automatic routing, ID validation, and complaint syncing across agencies.
* **Role Hierarchies & Delegation Chains**  
  Enhancing role-based control with multi-tiered admin structures to support escalation, reassignment, and oversight across departments.
* **Accessibility & Inclusion Enhancements**  
  Adding support for screen readers, large font modes, and alternate color schemes to ensure usability for individuals with disabilities.
* **Verified Complaints & Identity Proof**  
  Optional KYC integration for high-priority complaints, enabling resolution authorities to process them with stronger verification.
* **Public Announcements & Community Polls**  
  Adding features to broadcast resolutions, system maintenance, and community feedback campaigns for inclusive digital governance.