

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES



CTRL+X BUG TRACKING SYSTEM **(Prefinal) Project Report** **Software Engineering**

Submitted to:
Dr. Shantanu Agnihotri sir

Submitted by:
Naman Chanana SAP Id: 500119485
Soumya Jain SAP Id: 500119436
Smriti Walia SAP Id: 500124833
Amulya Jain SAP Id: 500122439

TABLE OF CONTENTS

- 1. Project Overview**
 - 1.1 Project Title**
 - 1.2 Objective**
 - 1.3 Scope**
 - 1.4 Problem Statement**
- 2. System Architecture**
 - 2.1 System Design**
 - 2.2 Data Flow Diagram (DFD)**
- 3. Technical Stack**
 - 3.1 Front-End Technologies**
 - 3.2 Back-End Technologies**
 - 3.3 Database**
 - 3.4 Core Algorithms/Logic**
 - 3.5 APIs/Integrations**
- 4. Project Implementation Status**
 - 4.1 Completed Modules**
 - 4.2 Ongoing Work**
 - 4.3 Pending Tasks**
- 5. Challenges and Problem Areas**
 - 5.1 Technical Issues**
 - 5.2 Design Challenges**
 - 5.3 Solutions Considered**
- 6. Testing and Quality Assurance**
 - 6.1 Testing Methodology**
 - 6.2 Bug Tracking**
 - 6.3 Test Coverage**
- 7. Project Timeline and Milestones**
 - 7.1 Timeline**
 - 7.2 Completed Milestones**
 - 7.3 Upcoming Milestones**
- 8. Documentation and Deliverables**
 - 8.1 Code Documentation**
 - 8.2 Brief User Manual**
 - 8.3 Repository Link**
- 9. Future Plans and Improvements**
 - 9.1 Enhancements**
 - 9.2 Scalability**
- 10. Team and Contribution Details**
 - 10.1 Team Members**
 - 10.2 Individual Contributions**
- 11. Project Prototype**

Bug Tracking System - Project Report

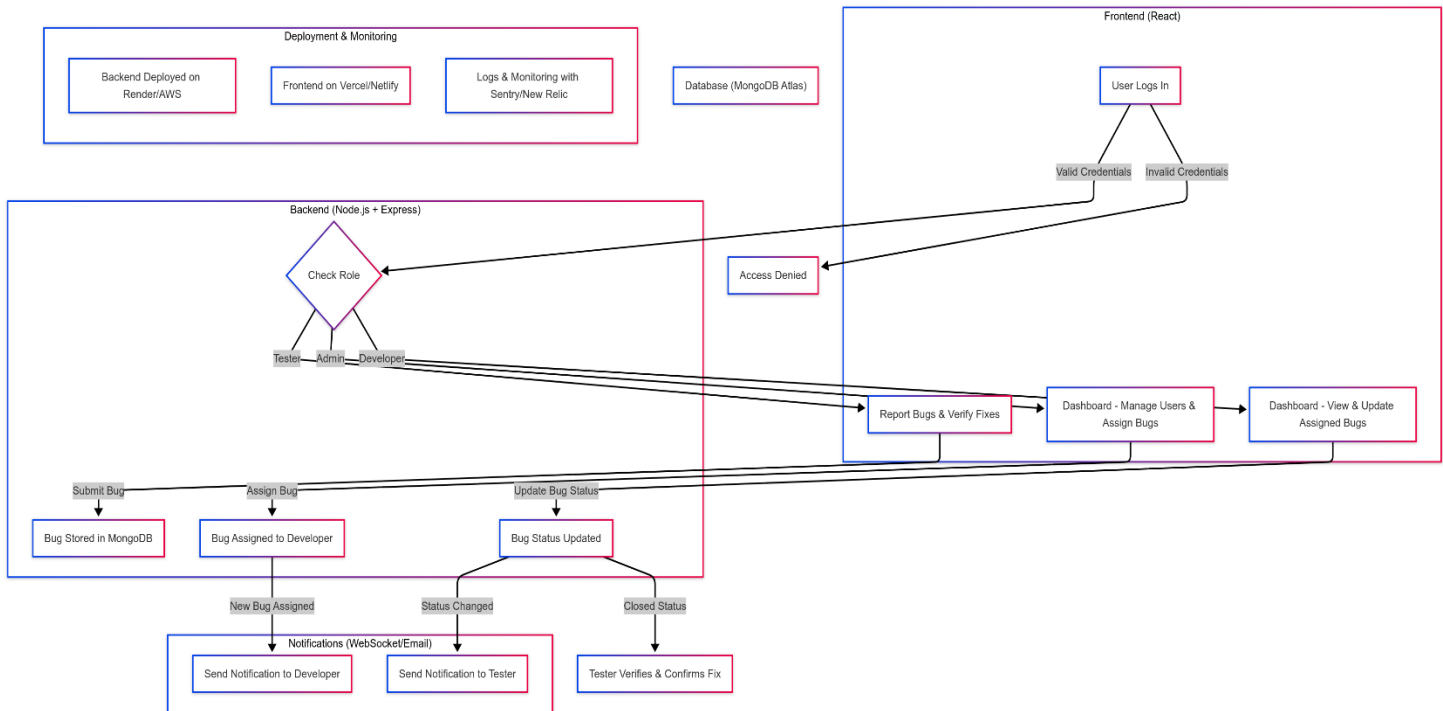
1. Project Overview

- **Title: Bug Tracking System**
- **Objective:**
 - The primary goal of this project is to develop a fully functional and scalable Bug Tracking System that helps software teams track, manage and resolve bugs efficiently.
 - The system ensures structured bug reporting, role-based assignment, resolution tracking and automated notifications to relevant stakeholders.
- **Scope:**
 - **Target Users:**
 - *Developers:* report, assign, and fix bugs
 - *Testers:* identify and submit bugs
 - *Project Managers:* monitor progress and assign tasks
 - **Functional Features:**
 - *User Authentication & Role Based Access Control(RBAC)*
JWT-based secure login with admin & user roles
 - *Bug Submission & Tracking*
Title, Description, Priority, Status, Assignee
 - *Automated Notifications*
Email updates on bug status changes
 - *Advanced Filtering & Searching*
by Priority, Status, Assignee, Date, etc.
 - *Comprehensive Dashboard*
Real-time bug statistics & charts
 - *File Attachments*
Upload screenshots/logs as proof of bugs
 - *Bug Report Exporting*
Download reports in PDF & CSV formats
 - **Problem Statement:**
 - Software development teams struggle with unorganized bug tracking, leading to delays and inefficiencies. This system provides a centralized and automated approach to manage bug reports, assign responsibilities, and track resolution progress in real-time.

2. System Architecture

○ System Design:

- Architecture: Client-Server model based on the MERN (MongoDB, Express.js, React, Node.js) stack.



○ Key Components:

- Frontend: Built with React 18 & Vite for high-performance rendering.
- Backend: Developed with Node.js & Express.js, utilizing JWT for authentication.
- Database: MongoDB, optimized with Mongoose ORM for schema modeling.
- Authentication: Secure login with password hashing (bcrypt) and token-based session management.

- **Data Flow Diagram (DFD):**

- **Level 0 DFD**

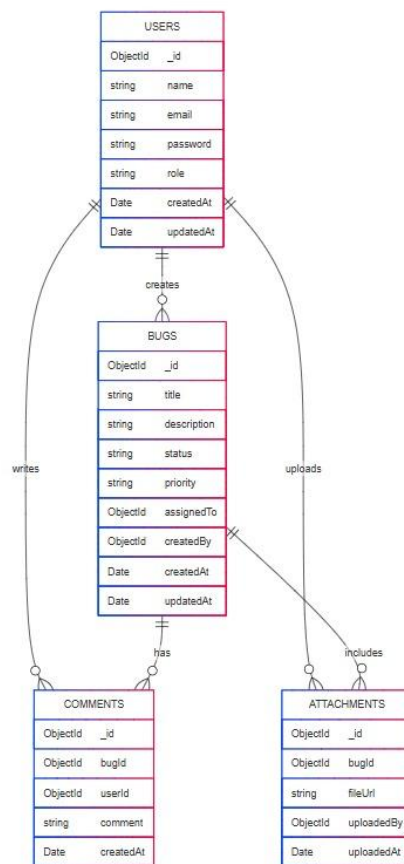
- Overview of how users interact with the system (submit, assign, track bugs).

- **Level 1 DFD:**

- User logs in (JWT Auth)
 - Bug submission and database entry
 - Assign bug to a developer/tester
 - Status updates trigger email notifications

- **Level 2 DFD:**

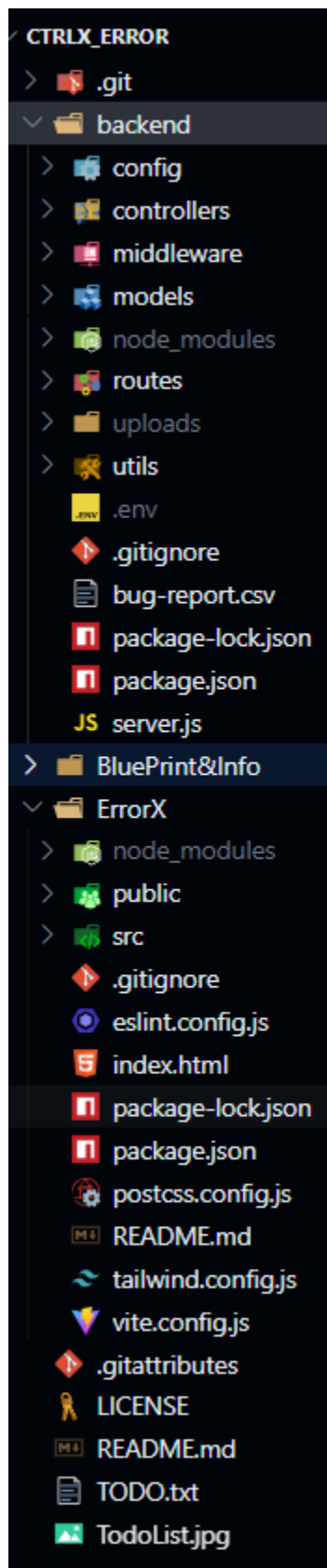
- Detailed breakdown of API interactions, data retrieval, and UI updates.



Database diagram

3. Technical Stack

- **Software Version Control System**
 - *Git and GitHub* for proper version control, bug fixes and project management.
- **Front-End (Handled by Frontend Developer):**
 - *React 18 with Vite* for fast build times and optimized performance.
 - *React Router* for navigation and page routing.
 - *TailwindCSS* for styling and UI responsiveness.
 - UI components designed for seamless user experience.
- **Back-End (Handled by Backend Developer):**
 - *Node.js with Express.js* for REST API development.
 - *JWT Authentication* (Login, Role-based Access Control).
 - *Nodemailer* for email notifications.
 - *Multer* for handling file uploads (screenshots/logs).
- **MongoDB with Mongoose ORM for schema structuring.**
- **API testing and debugging using *Postman*.**
- **Integration & Full-Stack Development:**
 - Ensuring smooth connection between frontend and backend.
 - API validation and error handling.
 - Middleware setup for request validation.
- **Deployment and hosting setup (*Netlify/Vercel*).**
- **Testing, UI Design & Research:**
 - Testing: Unit tests, API testing via *Postman*.
- **UI & Layout Selection:**
 - Researching UX best practices and implementing a responsive design using *Figma*.
- **Documentation:**
 - Preparing user manuals, API documentation, and system workflows.

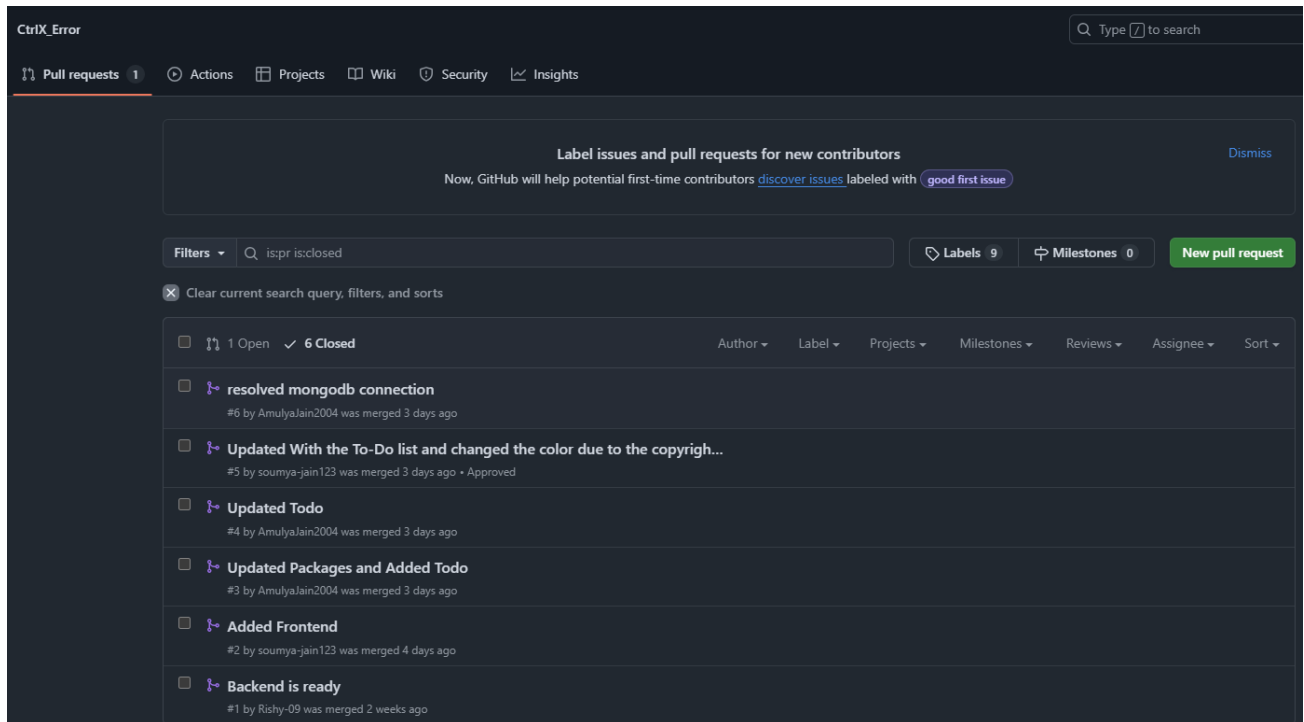


File Structure based on Software Design and Technology Stack

4. Project Implementation Status

○ Completed Modules:

- *Backend API Development* (All endpoints tested via Postman).
- *Authentication System* (JWT secure login & RBAC).
- *Bug Submission, Tracking, and Filtering*.
- *Email Notifications Integration* (Nodemailer tested successfully).
- *Frontend Setup* (React with TailwindCSS, basic UI ready).



○ Ongoing Work:

- *Frontend Bug Reporting & Detail View* (Form integration & state management).

○ Pending Features:

- *File Attachment System* (Upload & retrieve logs/screenshots).
- *Exporting Reports* (Generate & download bug reports as PDF/CSV).
- *Real-Time Status Updates* (Using WebSockets for instant bug tracking).
- *Full Application Testing* (Using testing softwares).
- *Deployment* through hosting services like Netlify/Vercel.
- *Feedback Analysis* of Users and Iterative Fixes
- *Final Software Application*

5. Challenges and Problem Areas

- **Technical Challenges:**

- *Frontend-Backend Sync*: Ensuring smooth data exchange between React and Express.
- *Scalability*: Optimizing MongoDB queries for large datasets.
- *File Storage Management*: Efficiently storing and retrieving bug-related images/logs.

- **Design Considerations:**

- *UI/UX Optimization*: Intuitive dashboard with real-time bug statistics.

- **Accessibility & Security**: Protecting sensitive bug data from unauthorized access.

- **Solutions Implemented:**

- *API optimization* with pagination & indexing.
- *State management* using React Context API.
- *Secure authentication* with encrypted JWT tokens.

6. Testing and Quality Assurance

▪ **Testing Methodology:**

- *Unit Tests*: Conducted using Jest & Mocha for backend logic.
- *Integration Tests*: API testing with Postman to verify full system workflow.
- *User Testing*: Frontend UI tested manually to ensure smooth user interactions.

▪ **Bug Tracking:**

- *Bugs and issues tracked using GitHub Issues*, assigned to team members based on priority.
- *Regular review meetings* to resolve critical bugs before release.

▪ **Test Coverage:**

- *Backend API*: Fully tested with unit and integration tests.
- *Frontend UI*: Currently in manual testing phase.
- *Major Findings*:
 - Backend testing successfully completed,
 - covering authentication,
 - role-based access,
 - bug tracking,
 - email notifications.
 - Some UI adjustments are pending for frontend integration.

7. Project Timeline and Milestones

▪ **Timeline:**

Milestone	Status	Deadline
Backend Development	Completed	15th March 2025
Frontend Setup	Completed	20th March 2025
API Integration	Ongoing	31st March 2025
Feature Completion	Pending	10th April 2025

▪ **Completed Milestones:**

- Backend API fully functional.
- Frontend basic UI integrated with backend.

▪ **Next Steps:**

- Connection establishment between Frontend and Backend.
- Complete file attachment & export feature.
- Deployment of the application.
- Final testing & deployment by April 10, 2025.

8. Documentation and Deliverables

▪ **Code Documentation:**

- *Detailed code documentation* and architecture explanation.
- *API Endpoint References*: A complete list of API endpoints with request parameters and response structures.

API Endpoints

The table below lists the key API endpoints and their functionalities:

HTTP Method	Endpoint	Description	Authentication
POST	/api/auth/register	Register a new user	No (Public)
POST	/api/auth/login	User login and token generation	No (Public)
GET	/api/users	Fetch all users	Yes (Admin)
GET	/api/users/:id	Fetch user by ID	Yes
POST	/api/bugs	Create a new bug report	Yes
GET	/api/bugs	Get all bug reports	Yes
GET	/api/bugs/:id	Get a single bug by ID	Yes
PUT	/api/bugs/:id	Update bug details	Yes
DELETE	/api/bugs/:id	Delete a bug report	Yes (Admin)
POST	/api/comments	Add a comment to a bug	Yes
GET	/api/comments/:bugId	Get all comments for a bug	Yes
POST	/api/attachments	Upload an attachment for a bug	Yes
GET	/api/attachments/:bugId	Get attachments for a bug	Yes

- *Workflow Diagrams*: Visual representation of the system's core operations based on the blueprint files.

▪ **Brief User Manual in README.md:**

- User Guide:
 - How to submit a new bug report.
 - Tracking assigned bugs and updating statuses.
 - Filtering, searching, and managing bug lists.
 - Exporting bug reports as PDF/CSV.
- User Roles & Functionalities:
 - *Admin*: Full control, can assign/reassign bugs, manage users.
 - *Developer*: Can update bug statuses and resolve issues.
 - *Tester*: Can submit new bug reports and track them.
 - *Project Manager*: Can oversee all bugs, track progress, and generate reports.

▪ **Repository Link:**

- *GitHub Repository*: [CtrlX Error](#)
- *Figma Design*: [Frontend Design](#)

9. Future Plans and Improvements

▪ **Enhancements:**

- *AI Integration:* Implement AI-based bug prioritization and auto-assignment.
- *Performance Improvements:* Enhance API response times and reduce latency.

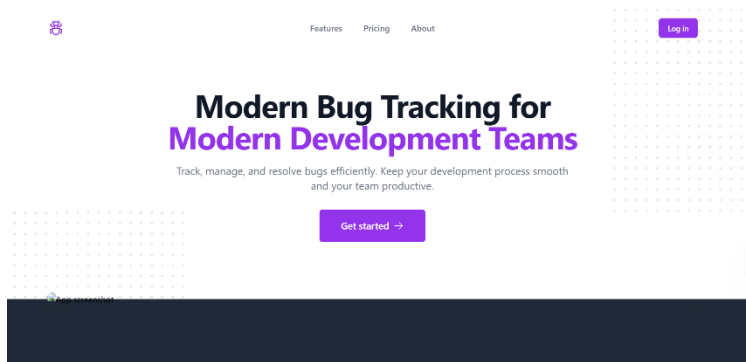
▪ **Scalability:**

- *Optimize MongoDB* queries and implement caching mechanisms.
- *Mobile Compatibility:* Develop a mobile-friendly version for better accessibility.

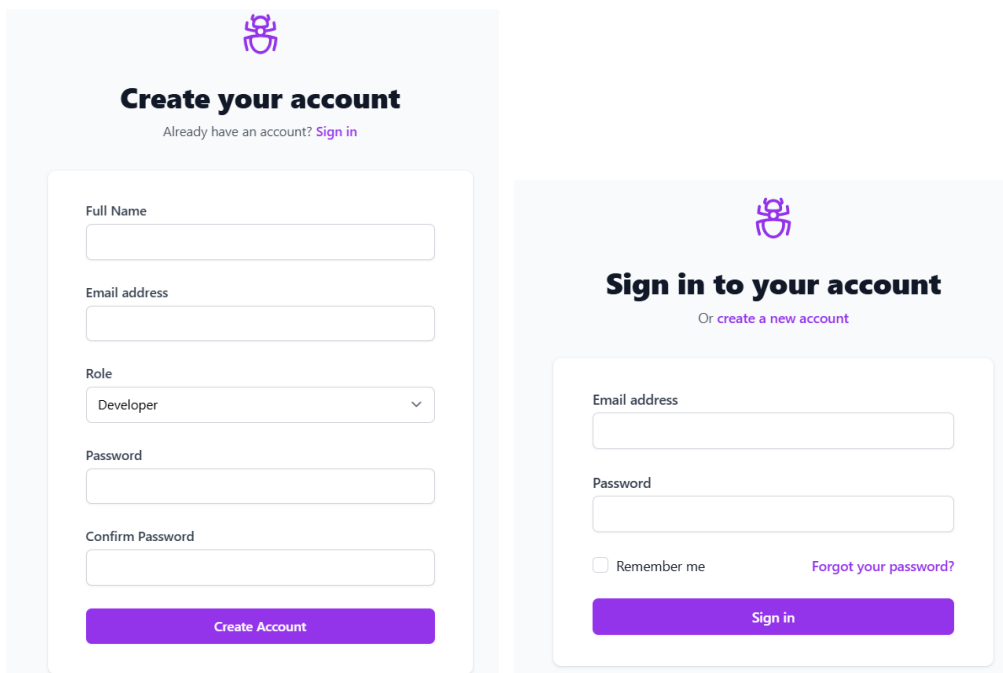
10. Team Members and Contributions

Name	Role	Contribution Details
Naman Chanana	Backend & Full-Stack Lead	API development and testing, Auth & Notifications, Database design, API testing
Soumya Jain	Frontend Developer & UX	UI design implementation, React integration, UX
Smriti Walia	Testing, UI & Research	Research on application and use case, Bug testing, UI layout, documentation and Figma Design
Amulya Jain	Integration & Deployment	Connecting backend & frontend, middleware, Resolved and fixed dependencies and major issues along with Versioning of the software.

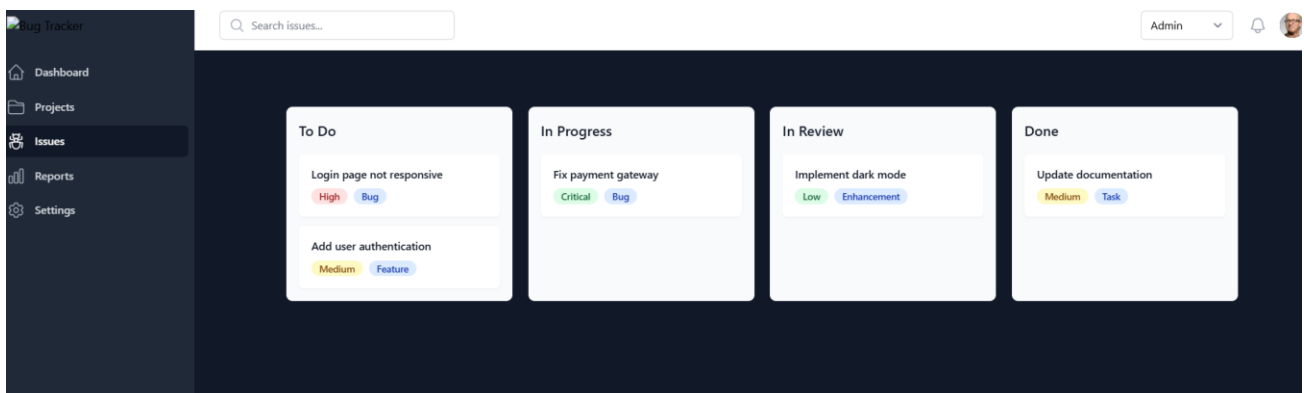
11. Current Working Prototype



Landing Page



SignUp and SignIn



Main Bug Tracker (Issues Section)