

Bug Tracker Web Application — Final Project Summary

Objective

This is a web-based bug tracking system developed for a company (e.g., XYZ Corp). The platform allows public users, testers, developers, and an admin to collaboratively manage and resolve software bugs in an organized and efficient manner.

Tech Stack

- **Frontend:** React, Axios, React Router DOM, Context API (or Redux)
 - **Backend:** Node.js, Express.js
 - **Database:** MongoDB (with Mongoose)
 - **Authentication:** JWT (JSON Web Tokens)
 - **Testing:** Postman (for API testing)
 - **Deployment:**
 - Frontend: Vercel
 - Backend: Render
 - Database: MongoDB Atlas
 - **Containerization (optional):** Docker, Docker Compose
-

User Roles & Permissions

- **Public User:** Can report a bug without logging in.
 - **Tester:** Logs in and reports bugs related to internal projects.
 - **Developer:** Logs in and updates the status of bugs assigned to them.
 - **Admin:** Manages users, assigns bugs, and creates/deletes projects.
-

User Flows

Public User

- Accesses the public bug report page.
- Selects the relevant app/website name.
- Submits a bug with a description, priority, and optional email.
- Bug appears in the admin dashboard for triage.

Tester

- Logs into their dashboard.
- Reports bugs tied to internal projects.
- Can view previously submitted bugs.

Admin

- Logs into their dashboard.

- Views all bugs: public and internal.
- Assigns bugs to developers.
- Manages users and projects.

Developer

- Logs into their dashboard.
- Views bugs assigned to them.
- Updates the status (open → in progress → closed).



Data Structure (Conceptual, Non-Technical)

- **Users:** Stores basic user info, role (admin, developer, tester).
- **Projects:** Represents individual software projects.
- **Bugs:** Each report includes a title, description, priority, status, reporter info, and optional email.



Frontend Pages Overview

- **Login/Register:** Secure access for internal users.
- **Public Bug Report Page:** Simple form for non-logged-in users.
- **Admin Dashboard:** View all bugs, assign developers, manage users/projects.
- **Tester Dashboard:** Report and view bugs.
- **Developer Dashboard:** View and update assigned bugs.



Backend Workflow Overview

- Uses secure JWT-based login.
- Routes are protected and role-restricted.
- Public bug reporting does not require authentication.
- Bugs are categorized as internal or public and managed by admins.



MVP Scope (1-Day Build)

- Login/Register with role-based redirect.
- Role-specific dashboards (admin, tester, developer).
- Public bug reporting page.
- Admin bug assignment.
- Bug status updates by developers.
- Simple project management.
- Desktop-first layout.



Future Enhancements

- Multi-company support.
- Screenshot/image uploads.

- Email notifications.
 - Advanced filters and search.
 - Dashboard analytics.
 - Mobile responsiveness.
-

Summary

This bug tracker application allows a company to efficiently manage software bugs from both internal and external sources. It is designed to be simple, scalable, and user-friendly, with room for future upgrades. This MVP version focuses on a single company and covers all essential workflows, making it ideal for fast development while laying the foundation for a larger SaaS-style platform later.