Bug Tracking System – Final Abstract

1. Project Overview

This project delivers a full-stack web application for tracking bugs and managing software development tasks, inspired by platforms like Jira. The aim was to develop a simplified yet functional issue-tracking system that enables authenticated users to manage projects, assign bugs, and monitor issue resolution in real-time.

The solution was built using a modular architecture that separates backend services, frontend interfaces, and documentation. Emphasis was placed on secure authentication, responsive design, and smooth user experience, making the application suitable for solo developers or small teams.

2. Concept and Features

The core concept behind this bug tracking system is to create a lightweight and efficient interface for issue management. The application supports the following major features:

- User Authentication: Users can register, log in, and securely manage sessions using JWT-based authentication.
- Project Management: Authenticated users can create and manage multiple projects.
- Issue Tracking: Within each project, users can create, assign, and update issues with relevant details.
- Role-Based Access (Optional Scope): Basic access control ensures only authenticated users can view or modify data.
- Responsive UI: Designed using Tailwind CSS, the frontend works effectively on both desktop and mobile screens.

The application mirrors real-world workflows and fosters task accountability across projects.

3. Technical Approach

Backend: - Developed with Java 17, using the Spring Boot framework for building RESTful APIs. - Spring Security and JWT are used for stateless authentication and authorization. - Hibernate ORM integrates with a PostgreSQL database to manage entity relationships. - DTOs are employed to abstract and control the exposure of data models, avoiding circular references and reducing payload size. - Layered architecture includes Controllers, Services, Repositories, and Model classes.

Frontend: - Built with React.js and Tailwind CSS for modern component-based and utility-first UI development. - API communication is managed via Axios, and authentication state is preserved using local/session storage. - React Router is used to protect routes and manage navigation between login, dashboard, and issue pages. - All pages are mobile-optimized, with conditional rendering and responsive utility classes.

Development Tools and Workflow: - Postman and Swagger were used for testing backend endpoints. - The C4 container diagram visualizes the overall architecture. - GitHub was used for version control, issue tracking, and deployment readiness.

4. Challenges and Lessons Learned

Throughout the project, several technical and conceptual challenges were encountered and resolved:

- Frontend–Backend Integration: Ensuring CORS configuration, base URL environment settings, and secure token transmission were critical for a seamless frontend-backend interface.
- Authentication: Implementing secure and stateless JWT authentication improved understanding of token management and Spring Security filters.
- Entity Mapping and DTOs: Circular references in JPA entities necessitated a move to DTOs, improving API response design.
- Responsive Layout Issues: Early reliance on third-party UI libraries (Flowbite) introduced layout bugs, later fixed by building custom Tailwind components from scratch.
- Debugging UI Layouts: Advanced CSS debugging techniques like outline: 1px solid and browser DevTools were crucial in resolving mobile layout spacing issues.

These experiences significantly improved my understanding of full-stack development workflows, secure design practices, and component-based architecture.

5. Conclusion

The Bug Tracking System project is a practical demonstration of full-stack web development, combining RESTful backend development with a responsive frontend interface. It showcases key software concepts such as modularity, authentication, API integration, and user experience optimization.

The lessons from this project have laid a strong foundation for future contributions to collaborative software engineering and scalable web applications.