Full-Stack Task Tracker Web App - Final Report

Executive Summary

This project successfully delivers a comprehensive task management solution built with modern full-stack technologies. The application demonstrates proficiency in enterprise-grade development practices, including RESTful API design, database management, containerization, and responsive frontend development.

Technical Architecture

Backend Stack

- **Framework:** Spring Boot with comprehensive dependency management
- **Database:** Dual database support (H2 for development, PostgreSQL for production)
- **API Design:** RESTful endpoints with full CRUD operations
- **Documentation:** Swagger/OpenAPI 3.0 specification
- **Data Layer:** Spring Data JPA with custom repositories
- **Containerization:** Docker Compose for database orchestration

Frontend Stack

- **Framework:** ReactJS with modern component architecture
- **Features:** Advanced pagination (5 tasks per page), filtering by status and priority
- **Sorting:** Multi-criteria sorting by priority and due date
- **Navigation:** Complete pagination controls (First, Previous, Next, Last)
- **Port:** Development server running on localhost:5173

Key Dependencies & Tools

- **Testing:** Postman collections for comprehensive API testing
- **Development:** Hot reload with Spring DevTools
- **Object Mapping:** Lombok for reduced boilerplate code
- **Database Connectivity:** JDBC API with PostgreSQL driver integration

Core Features Implemented

Task Management

- **Create Tasks:** Full task creation with priority levels and due dates
- **View Tasks:** Organized task lists with clean interface
- **Update Tasks:** In-place editing capabilities
- **Delete Tasks:** Safe task removal with confirmation

Advanced Functionality

- **Filtering System:** Dynamic filtering by task status and priority levels
- **Pagination:** Efficient data loading with 5 tasks per page display
- **Sorting Options:** Multi-field sorting by priority and due date
- **Task Lists:** Hierarchical organization of tasks within lists

API & Documentation

- **RESTful Endpoints:** Complete CRUD operations for both Task and TaskList entities
- **Swagger Integration:** Interactive API documentation at localhost:8080/swagger-ui/index.html
- **Postman Testing:** Pre-configured collections for all endpoints
- **Data Transfer Objects:** Proper DTO implementation with mappers

Database Design

The application implements a robust data model with proper entity relationships between Task and TaskList objects. The dual database approach allows for seamless development with H2 and production deployment with PostgreSQL, both managed through Docker containers.

Development Workflow

- **Version Control:** Git repository with organized structure
- **Environment Management:** Configurable database connections
- **Port Management:** Automated conflict resolution guidance
- **Dependency Management:** Maven-based build system with IntelliJ integration

Quality Assurance

- **API Testing:** Complete Postman collection coverage
- **Documentation:** Auto-generated Swagger documentation
- **Code Quality:** Industry-standard architecture patterns
- **Error Handling:** Proper exception management and user feedback

Conclusion

This Full-Stack Task Tracker Web App represents a production-ready application that successfully integrates modern development practices, comprehensive testing, and user-focused design. The project demonstrates strong technical execution across all layers of the application stack and provides a solid foundation for future enhancements and scaling.

Final Assessment: Project successfully meets all objectives with professional-grade implementation and documentation.