

Sprint Backlog

Goals:

- Database Enhancements:
 - Enhance the PostgreSQL database to support new features: Settings, Inventory, Jobs, Purchases, and Dashboard.
 - Set up necessary database tables and populate them with initial data.
- Frontend Expansion:
 - Expand the React application to include new pages: Settings, Inventory, Jobs, Purchases, and Dashboard.
 - Ensure new pages communicate effectively with the backend via API calls.
 - Implement responsive designs to improve mobile accessibility.
- API Extension:
 - Extend the REST API using Node.js and Express.js to handle requests for new frontend components.
 - Implement comprehensive data validation and sanitization to enhance security.
 - Develop and secure API endpoints.
- Presentation:
 - Create a detailed presentation for the Sprint Deliverable that encapsulates the enhancements to the database, frontend expansions, and API extensions.

Frontend Development:

- Base and Tab Configuration:
 - Revisit and revise the baseline page to incorporate and align with expanded goals.
 - Create and configure a dedicated page for each new feature: Settings, Inventory, Jobs, Purchases, and Dashboard.
 - Evaluate the need for additional navigation tabs to support new features, ensuring comprehensive functionality.
- Dynamic Data Integration:
 - Populate all new pages and tabs with dynamic data sourced from the backend, reflecting real-time updates and user interactions.
- Admin Role Enhancement:
 - Expand admin role capabilities to encompass comprehensive management features like user registration, inventory oversight, job listings, purchase approvals, and dashboard analytics.

Backend Development:

- API Development:
 - Develop new API endpoints for managing Settings, Inventory, Jobs, Purchases, and the Dashboard.
 - Ensure APIs for dynamic data interactions like CRUD operations on inventory and processing orders are robust and secure.
- Security Enhancements:
 - Strengthen endpoint security through rigorous testing and implementation of best practices in data handling and authentication.

Deployment and Integration:

- Docker and Server Hosting:
 - Deploy the project on a server (Windows Desktop acting as server) using dockerized containers
 - Implement Github action to automatically deploy main after running automatic unit testing, which is a separate Github action.

Additional Tasks and Enhancements:

- Implement logging features across all functionalities to aid in debugging and monitoring.
- Setup GitHub Actions for automated testing and deployment facilitation.
- Design and implement modals and interactive elements for better user engagement.
- Address UI/UX enhancements for mobile and desktop views to ensure optimal user experience.

Security:

- Implement Role-Based Access Control (RBAC) to manage user permissions effectively.
- Ensure critical security features like email verification, password management, and secure token handling are robust and up-to-date.