Project Github

Project Website (PWEB)

Inventory Site (Product)

Teammates

Michal Zajac

Brandon Bejarano

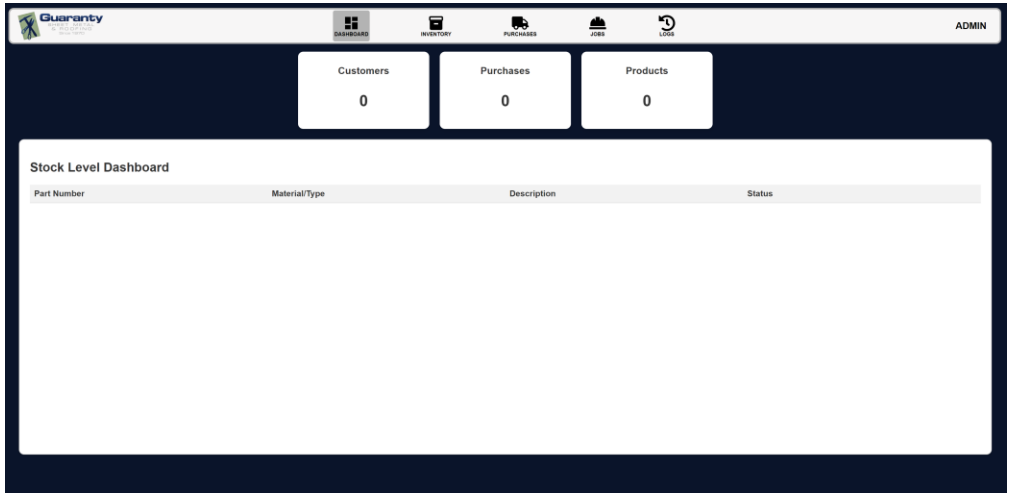
Mason Wittkofski

Jake Carney

Waleed Kambal

Project Focus

- Our project is focused on designing an **inventory management system** for Guaranty Sheet Metal based in New Orleans, LA.
- We're looking to build them a web application that they can use to track their current inventory, create and edit incoming and outgoing shipments for jobs, and manage supplies received.

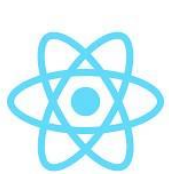


High Level Functionality (1 of 2)

- **Dashboard**
  - Displays "low" and "out of stock" products
  - Displays the number of jobs, purchases, and products
- **Inventory**
  - Add, delete, edit inventory
  - Filter inventory using checkboxes and Search bar
- **Purchases**
  - Add, delete, edit purchase orders
  - Search for purchase orders
  - Add all "low" and "out of stock" products to order
  - Generate an excel file for the new order that can be sent to supplier
  - Adjust mark-up pricing based on new inventory received

High Level Functionality (2 of 2)

- **Jobs**
  - Add, delete, edit jobs
  - Upload estimate pdf that the sponsor uses
  - Add Necessary and Used parts for the job
  - Remove product from stock if its available
- **Logs**
  - Implemented logging of every action in the app to be sure who changed, what changed, and when it changed, so the sponsor can manage the integrity of the data in the app
- **User Management**
  - Role Based Access Control
  - Add/Remove a user from the application
  - Edit the role of a user in the app if ever needed
  - Edit of the user's information (profile)
    - Ex. Username, Password, Email
  - Update Web token based on changes



Major Sprint Achievements

- Got the app **hosted** on an actual server without using external hosting services
- Set up an **automated testing** workflow through GitHub that checks if tests pass, that blocks merges and pushes from being added to main branch that cause issues.
- Finished Inventory, Jobs, and Purchases pages which interact with each other and show updates based on items added, edited, or removed from the database
- Integrated the JSON Web Token into the **management of users** to match updates that occur
- Implemented role-based access control (**RBAC**) to allow admins to view logs, manage users and add users to the application.
- Implemented a simple informative **Dashboard** monitoring stocks, as well as quick stats about the number of products, customers, and purchases made.
- Implemented **logging** for security reasons.
- Fixed some **security issues** by using environment variables.

Brandon's Contributions

- Self-hosted our web application on a spare PC
- Set up a DNS for the web application
- Set up a production version of the Docker Compose file for the server
- Setup GitHub Action to automatically run Cypress Test whenever a pull request is made to the main branch
- Setup GitHub Action to automatically update server whenever merge is made to the main branch
- Created documentation for the Nginx reverse proxy
- Created tests for the Logs page

Reflection:

- I learned to work with multiple different tools like Docker, Nginx, Node.js, React and Express to create a reverse proxy, REST API, and a web application
- The hardest part was learning how to self-host our project on a server and have it pull the most up to date version of the main branch from GitHub.

Jake's Contributions

- Focus: **Inventory Page**
  - Developed front end for inventory page, API calls to database, and designed modals
  - Developed logic for filtering inventory
  - Implemented autofill and auto part number generation
  - Excel file upload
- Created logging page and logic for logging.
- Creating modals for other pages.
- Code clean up.
- Helped with little things on every page.
- Administrative things to make sure we met standards for project and with sponsor.

Reflection:

- Learned that we should implement a pipeline as the very beginning of the project.
- The hardest part of the project was figuring out how we should host the project.

Waleed's Contributions

Focus: **Dashboard Page**

- Developed an API for role-based user access specific to the Dashboard page.
- Ensured that the "Manage Users" button on the Dashboard correctly retrieved and displayed user roles, and the control to remove and edit roles.
- Created an API to fetch and display accurate, live data from various pages on the Dashboard.
- Completed all required documentation for the Dashboard as specified in the project's Blackboard.
- Organized and categorized all Markdown documents related to the Dashboard into their designated folders.

Reflection: I do not know how to read or know how to properly read merge conflicts (I broke main 4x). Learned how to thread api to the front end. Team building/collaboration, code discussions, and worked through bugs and features with a team.

Mason's Contributions

- Focus: **Purchases Page**
- Developed Complete Front-End Interface
- Wrote multiple REST Api's to update orders, and their respective items and quantities.
- Added status tracking to orders to trigger necessary actions.
- Created End to End testing for the purchases page and its functions
- Made documentation for all features I impacted

Reflection:
I learned a lot during this project. How to work on a full stack software team. Also how to expose myself to new tools and languages. The hardest part for me was being able to wrap everything together into one system.

Michal's Contributions

- Focus: **Jobs Page**
- Created database tables
- Developed a Frontend for Adding, Editing, Removing Jobs
  - Uploading and managing Estimate PDF w/ 5mb limit
  - Included Necessary Parts + Used Parts sections
- Developed 25+ API endpoints
- Created E2E testing for Jobs page using Cypress
- Created documentation
- **Reflection:** I exposed myself to new technologies and learned valuable skills in full-stack development. I had a great experience building a real product and collaborating in a team.

Significant Items on Backlog

- Completed
  - Role based access control
  - Implementing the pipeline
  - Getting it hosted on self-hosted server with a dns
  - Implementing API
  - Getting all of the pages to work together
  - User Authentication and Web token usage to verify
- Not-Completed
  - PDF scrapers for estimates and invoices.
  - Upload of invoice pdf
  - Access Guaranty Sheet Metal server in Louisiana
  - Email verification

Demo