ProdWizard

CSI 2999 Final Project Report

Group 2

Jacob Chunn, Chris Stefaniak, Fulton Zinser, Peyton Skwarczynski, Evan Bogedin

Professor Hany Othman

**Abstract:**

ProdWizard sets out to fill a gap in the manufacturing industry, allowing employers to manage, track, and statistically analyze their employees. Through browsing the web, there were no manufacturing employee management systems that had the functionality to track statistics of their employees. Through the usage of a web application as a frontend for users to interact with our product, administrators of any manufacturing company are able to easily add tasks for their employees to complete. The backend was created in Java using the Springboot framework and Maven as a build tool, with the entire project being deployed with Kubernetes. Our approach to this project was to set out to make an application that allows employees to complete tasks from their employers, while still being able to view the tasks of all employees. This facilitates manufacturing companies in remaining on the same page internally, and provides up-to-date information across cross-functional teams.

**Table of Contents:**

**Introduction - pg 2**

**Project Scope - pg 3**

**Literature Review - pg 3**

**Methodology - pg 3**

**Project Implementation - pg 4**

**Discussion - pg 4**

**Recommendations - pg 5**

**Lessons Learned - pg 5**

**Project Timeline - pg 6**

**Team Member Contributions - pg 7**

**Project Evaluation - pg 7**

**References - pg 7**

**Introduction:**

Manufacturing is an important part of our modern industrial world. Every product that is sold on the market, has to be made by someone. As products become more popular, they must be produced in a way that allows them to be more widely available. This is where the manufacturing process comes into play. Manufacturing allows complex products, like cars or household appliances, to be able to be produced on a large scale. The importance of manufacturing is clear but the process itself is quite intricate, requiring many people over cross-functional teams to understand their objectives and their respective descriptions. This is why we set out to make ProdWizard fulfill the need of managing manufacturing companies’ employees. Additionally, part of our objective is to make the tasks assigned within ProdWizard able to be seen across employees, promoting transparency within manufacturing companies. This is an important concept of ours as it allows companies to continuously be on the same page with everyone internally.

**Project Scope:**

The scope of ProdWizard is to provide a web application for both employers and employees of manufacturing companies. Although there are a preset of tasks included in the program, the project should allow administrators to create tasks and view statistical data about the individual tasks assigned.

**Literature Review:**

Monday - Monday.com is a website that allows either a single person or a group of people to keep track of tasks to be accomplished. Monday is an inspiration for our web application, but we felt that its scope was too general. To make our project more marketable, we decided to create our employee management system specifically for the manufacturing industry.

Trello - Another web management system is Trello, which serves to manage tasks specifically catered towards the workplace. Trello was also different from Monday in that they contained a more user friendly GUI. For our final project, we felt that it was important to enhance the user experience by implementing a navigable GUI. By adding in valuable information about recurring tasks that any manufacturing workplace may need to complete, we completed this additional goal.

**Methodology:**

We created this project using a full stack web application using a Java Spring framework backend and a vanilla JavaScript frontend. We connected these two pieces together using a REST API and a plethora of different endpoints. This was done to simulate how a real world continuous application would be deployed. The project also contains a Helm chart for deployment on Kubernetes to further simulate a true production environment. For the pods in the aforementioned Kuberentes cluster, dockerfiles are included in the repository to create docker images of backend and frontend. Our deployment method simulates a real-world production environment by making ProdWizard very scalable if desired.

**Project Implementation:**

We have two main HTML files that our user can traverse between, the login page and the home page. In the login page, the user can login with a username and password that is searched for in the backend. If these credentials are found, then the user will be granted access to the home page through the distribution of a session token. Once users are at the home page, they can use a variety of buttons which trigger REST API calls to store or get information from the backend. Through the input of information into the backend, the data will be manipulated to create the leaderboard and averages for various tasks that are returned through REST API calls.

**Discussion:**

Our initial project objectives included the ability to track various stats such as ‘parts sorted per hour’, ‘parts made per hour’, ‘goods imported per hour’, ‘goods exported per hour’, ‘items approved/tested/inspected per hour’, or ‘employees checked on per hour’. With ways to interpret data gathered from task completion, such as a performance ranking board or graphs representing the role-specific data. While due to time constraints we only implemented parts per minute and the leaderboards but did not have time for a graph representation of the stats.

**Recommendations:**

Currently, our employee management system, ProdWizard, accomplishes our initial goal of being an accessible and navigable web application to manage a company, tailored to the manufacturing industry. To set our application apart from current web employee management systems, we additionally included statistics on completed company tasks. Some further improvements we can make in our application include:

-Create graphs of gathered data

-Add to statistical analysis of completed tasks by including more attributes linked to submitted tasks

-Add to user profiles by including more personal information (work & personal email address, work & cell phone number, work & home address)

**Lessons Learned:**

We learned a few valuable lessons from creating this project. One is that frameworks (especially web frameworks) exist to make the developer’s life easier. Developing a frontend with vanilla JavaScript proved to be much more challenging and unreadable than creating a frontend with a framework or library such as React or Svelte. In the future, we will consider one of these tools to aid in our development. We also learned that it is important to delegate user stories before beginning work on a project so that everyone on the team is contributing equally. Along with this is dealing with bottlenecks by either recognizing that the bottleneck does not exist or through “mobbing”, or pushing through the bottleneck with frequent rotation paired programming.

**Project Timeline:**

Weeks 1-3 - Gathered project ideas. Discussed pros and cons of each potential project.

Weeks 4-6 - Finalized employee management system project idea. Discussed potential features of the project, including: required features to set our project aside from current employee management systems, features to include if we have additional development time. Drew pictures overviewing the layout of the web application.

Weeks 7-10 - Began the initial development phase. Created the login page and established a working REST API call between the frontend and backend

Weeks 11-13 - We expanded the amount of API calls a user can make through the implementation of more buttons which were accessed on the home screen. These api calls sent and retrieved information from the backend. Statistics were implemented in the backend to produce features such as the leaderboard and task rates. In the later weeks of this period, the Helm chart for deployment was created.

Weeks 14-15 - Wrapped up any last minute software development. Created final project report. Recorded final presentation. Submitted individual peer reviews. Engaged with peers by replying to other groups’ presentations.

**Team Member Contributions:**

Jacob Chunn - Java programming with Springboot/Maven, mainly backend. Minor frontend programming.

Chris Stefaniak - Helm chart for Kubernetes deployment and REST API implementation

Fulton Zinser - Learned about back end development, Research into other similar applications.

Peyton Skwarczynski - Web application GUI. Learned Javascript, HTML, CSS for frontend development

Evan Bogedin - Javascript, HTML, learned about back end development, kubernetes and docker

**Project Evaluation:**

We would consider our project a success. It has deviated from our initial vision, but for the modified vision of our project, it is a success. We worked initially to obtain a minimally viable product, which is something tangible that users can test and we can collect data on these tests. Our project has surpassed this point and has a wealth of features.

**References:**

“A New Way of Working.” *Monday.Com*, monday.com/. Accessed 29 Nov. 2023.

“Trello Brings All Your Tasks, Teammates, and Tools Together.” *Trello*, trello.com/. Accessed 29 Nov. 2023.