**Capstone Project Document**

**Essential Basket**

**Introduction**

Purpose

* As consumers become more reliant on online shopping, we realized there was a huge opportunity to do business in this space. Therefore, we decided to develop an online supermarket platform to meet customers' needs for food and daily needs.
* This problem is well worth solving because many people in modern life are more dependent on online shopping, especially in the fast-paced life.
* There are not many online food and daily necessities shopping platforms on the market. Users may face problems such as poor shopping experience, delayed delivery, and limited product selection.
* Provide an easy-to-navigate, simple-to-operate website where users can easily find the food and daily necessities they need. Fast and reliable order processing and delivery services to increase user satisfaction.
* Several competitors already exist and operate in the online food and lifestyle shopping space. However, there are many opportunities to further improve user experience and service quality.

Industry/Domain

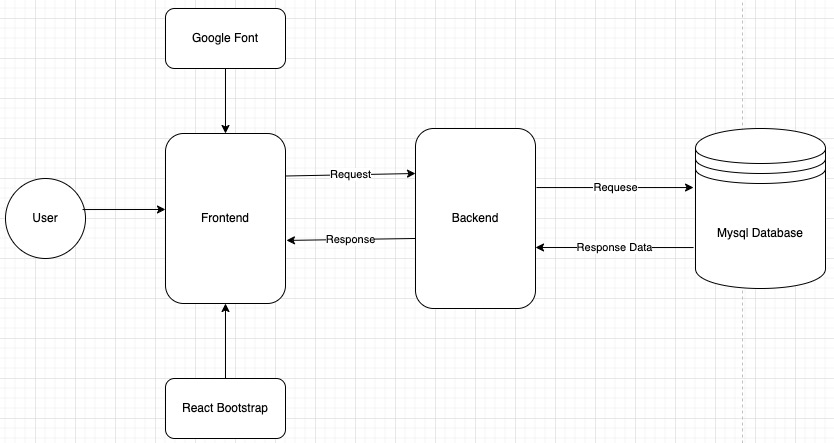
* The project is in the online retail industry.
* Online retail industry is in a stage of rapid growth and constant evolution.
* This project has relevance to other industries, particularly in digital payments, logistics and food supply chains. Online supermarket platform is not only about e-commerce but also handles key processes such as payment and delivery.

Stakeholders

* Customers: As users, customers are the most important stakeholders. They will use the online food and grocery supermarket software to browse products, place shopping orders, pay and get delivery. They hope that product quality is guaranteed, and the ordering process is simple.
* Management: Management is concerned about whether online food and grocery supermarket software can improve work efficiency and sales. They want operations to be simple and efficient.

**Product Description**

Architecture Diagram



User Stories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **User Story Title** | **User Story Description** | **Priority** | **Additional Notes** |
| 1 | Customer | Customers can get a clear and tidy product list with product names and prices clearly visible. | 5 |  |
| 2 | Customer | Customers can view detailed product information, such as product specifications, product descriptions, product ratings, and product reviews, by clicking on the products in the product list. | 5 |  |
| 3 | Customer | Customers can add products to their cart with one click. | 5 |  |
| 4 | Customer | Customers can browse the items in their shopping cart and add new items, change quantity, or delete them. | 5 |  |
| 5 | Customer | Customers can place orders to purchase the items in their shopping cart in the simplest way. | 5 |  |
| 6 | Customer | After the customer places the order, they can also change the billing information, such as the recipient's name, address, phone number, etc. | 4 |  |
| 7 | Customer | Customers can check the status and history of their own orders. | 4 |  |
| 8 | Customer | Customers can rate products or add comments. | 3 |  |
| 9 | Supermarket Manager | Supermarket managers can add new products and change the information and status of existing products. | 5 | Does not actually delete the product from the database, but sets the available attributes available/unavailable |
| 10 | Supermarket Manager | Supermarket managers can process the customer's order and change the customer's order status according to different processing stages. | 5 | Order status changes make it easier for managers to manage orders and for customers to view them. |
| 11 | Supermarket Manager | The supermarket manager hopes that customers can only change the order information when the order status is "Pending" so that the manager can process the order later. | 4 |  |
| 12 | Supermarket Manager | Supermarket managers can manage product comments and ratings given by customers and can delete comments and reset product ratings if necessary. | 4 |  |
| 13 | Supermarket Manager | Supermarket manager wants all admin functions to disallow customers access | 5 |  |
| 14 | Supermarket Manager | Supermarket managers have access to not only admin functions but also all customer functions for customer side user experience. | 5 |  |
| 15 | Supermarket Manager | Supermarket managers want to subtract the corresponding quantity from the product stock in the database after the customer places an order. | 4 |  |

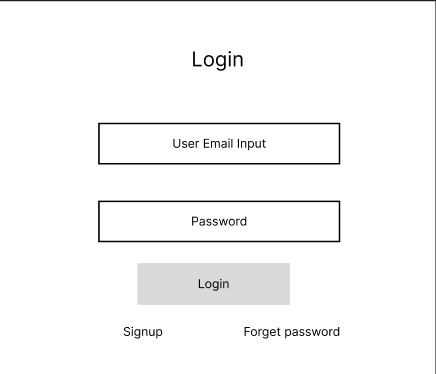
User Flow

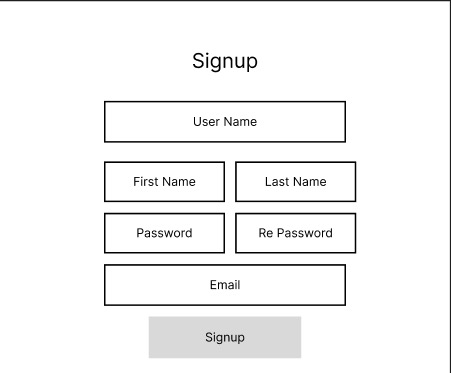
* CustomerA diagram of a product

  Description automatically generated
* A diagram of a product

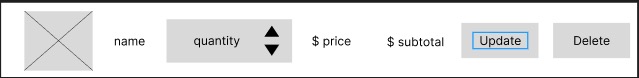
  Description automatically generatedAdmin

Wireframe Design

* NavBar
* Login Component
* Signup Component



* Product Card
* A screenshot of a website

  Description automatically generatedProduct Detail
* Cart Card
* A screenshot of a phone

  Description automatically generatedOrder Information Input

A diagram of a software company

Description automatically generated with medium confidenceDatabase Design

Out of Scope

* Forget Password:

Password reset feature that allows users to reset their password via email. Users can confirm they are the legitimate account holder by clicking on the "Forgot Password" link. This can include sending a verification code to the user's registration email. Upon successful verification, allowing the user to create a new password with the verification code.

* Online Payment:

Provides a variety of payment methods to meet the different users’ needs.

* Message Push:

Once the customer order status is updated, push message will be notified to the user.

* Order Tracking:

When the order is being dispatched, the order status is "shipping", and customers can see the logistics information and tracking.

Non-functional Requirements

* Performance Requirements:

The website should provide fast response times under normal load conditions to ensure users can browse and shop smoothly. Responding to user requests is expected to be completed within 1 second to ensures that users experience smooth performance while browsing and shopping.

* Availability Requirements:

The website should be highly available and able to provide services under a variety of conditions, on different devices, and at different times.

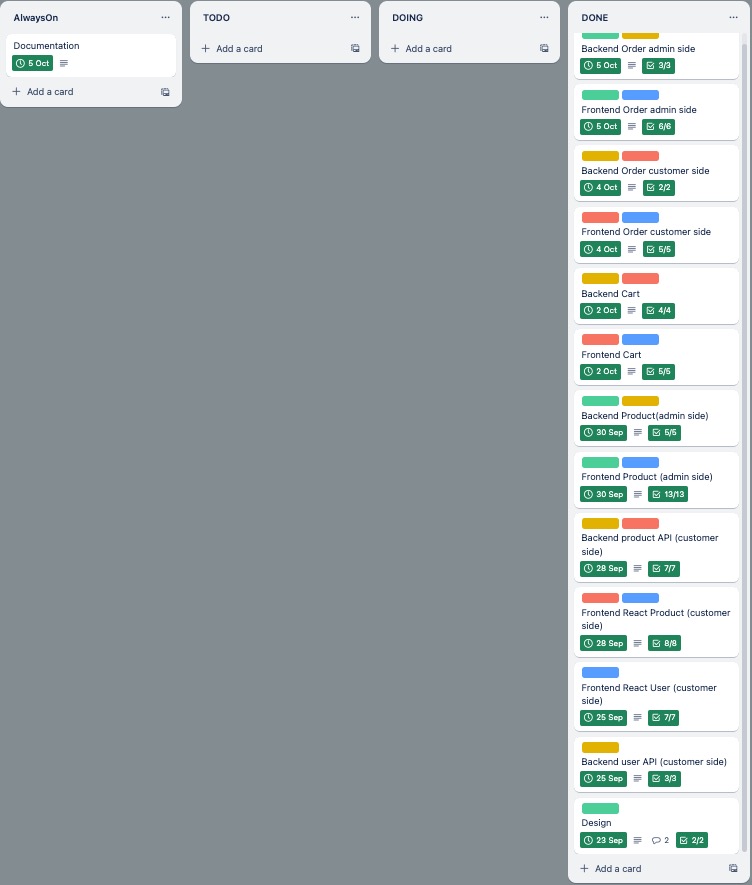
* Security Requirements:

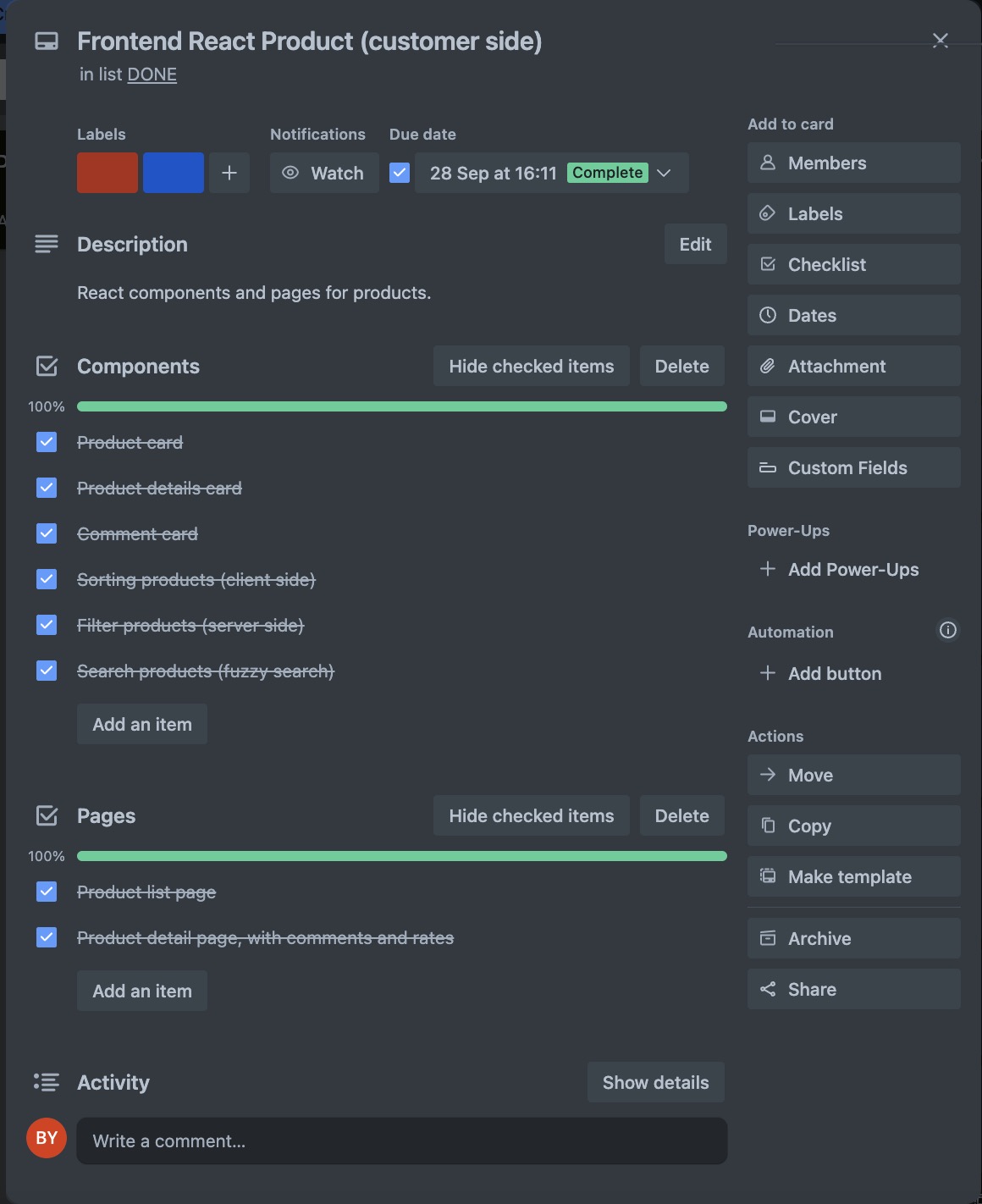
User data should be stored encrypted to protect user privacy.

* Reliability Requirements:

Ensure that the system does not fail over an extended period of time.

**Project Planning**

****

****

**Testing Strategy**

What were steps undertaken to achieve product quality?

* Requirements analysis:

The functional and non-functional requirements of the project were analyzed.

* Comprehensive Testing:

Simulate users (managers and customers) to operate all functions of the project to check whether each function can run correctly, and different functions cooperate correctly with each other.

* Security Testing:

We tested the encryption of passwords during registration and the comparison of encrypted passwords during login.

How was each feature of the application tested?

* Unit Testing:

The functionality of each unit (module or component) is tested to confirm its behavior under different input conditions and navigation.

How did you handle edge cases?

* Input restrictions:

Restrict the values that can be input into the textboxes. Restriction methods include using regular expressions, setting the textboxes type, setting the maximum and minimum values allowed for input in the textboxes, and determining whether the textboxes are empty.

* Error Handling:

Use error handling methods to catch any potential exceptions and prevent system crashes.

**Implementation**

What were the considerations for deploying the software?

* Platform and Environment
* Security
* Cost

**End-to-end Solution**

How well did the software meet its objectives?

* The project achieves its goals effectively and delivers on its promises mostly.

**References**

Where is the code used in the project? (link to GitHub)

* GitHub Repo: <https://github.com/BomingYu/Capstone_Project>
* Login Token: <https://github.com/jobatkinIOD/capstone-example/blob/main/backend/middleware/auth.js>
* File Upload Middleware: <https://github.com/jobatkinIOD/capstone-example/blob/main/backend/middleware/uploads.js>

What are the resources used in the project?

* Sequelize Documents: <https://sequelize.org/docs/v6/category/core-concepts/>
* React Bootstrap: <https://react-bootstrap.netlify.app/>
* Google Fonts: <https://fonts.google.com/>
* Flaticon: <https://www.flaticon.com/>