Capstone Project

Front End (React)

Products Inventory

[CP-FER-001]

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| Apr-17-2020 | 1.0 | Initial Version | Front End (React) team |

# Products Inventory

The proposed system should have provision to view the list of products and allow only the registered and authenticated users to add, edit, delete the products and view the product detail. The user interface to the system should be implemented using React. **The data about the products should be managed in a JSON server**. The client application should interface with the JSON Server **through http** to retrieve and store data.

**Note:**

1. The features given below are the **minimum basic features** that are required to be implemented. But you need not limit yourself. You can try to make it as close to the real system and it is left to your imagination and the domain knowledge you possess.
2. To have a responsive user interface and make the user interface better, **you should use Bootstrap**. Also, make the UI as modern as possible by using Bootstrap Cards, Grid, Carousel etc. instead of having just a plain table.
3. Appropriate input fields should be used to make it user friendly. For example, where date is inputted, you should use a date field and not a string field.
4. **Validations should be done**, wherever applicable, and make it user friendly.
5. Wherever list of records are viewed, there should be **Search filter** provided to only display a subset of records, so that user can quickly see it when there are many records to be viewed.
6. While viewing product list, when different fields of the products are displayed, **customization** should be provided to display only a subset of the fields. – See at the end of the document for a Note on Customization of fields.
7. It is mandatory to use **React routing** and **lazily load the routes**.
8. **Output screenshots** of the various features of the application should be submitted. As the application should be responsive, the output screenshots should include, as it appears on the desktop as well as it appears on the mobile. The output as seen on the mobile need not be from actual mobile but as simulated on the Google Chrome browser.
9. React Component DOM **test cases has to be implemented** for various features of the application. The test cases are mandatory.
10. It is important to submit the output screenshots of the execution of React TDD Jest test cases before it can be taken up for evaluation.

Following functionalities should be implemented in the system.

1. Registration and authentication
2. About
3. Product Inventory Management

## Registration and Authentication

Users should be able to register to identify themselves to the system so that they can sign in and perform add, edit and delete operations. **Viewing of product list summary should be allowed even without signing in.**

1. Register
2. Sign In
3. View registration details

**Register**

To register with the system, the following input details should be obtained.

1. Email ID
2. Password (for local authentication)
3. First name
4. Last name
5. Location
6. Mobile Number

**Note:**

1. Email ID should be used as the user ID for signing in.
2. The user registration data should be maintained in JSON server apart from the products data information.

**Sign In**

User should provide his/her email ID and sign in. If the user is already registered, the system should allow to sign in. Otherwise, it should display an error saying that the user is not registered. Authentication should be implemented using local authentication (maintaining user credentials in the backend server, i.e., JSON Server).

**View registration details**

The user should be able to view the details he/she had provided during registration.

## About

Users should be able to view the purpose of the application.

## Product Inventory Management

Users should be able to perform the following operations.

1. View Product List (Summary)
2. View Product Detail
3. Add Product
4. Update Product
5. Delete Product
6. Chart – Top viewed products

**Note**:

1. Viewing of products (Summary) should be allowed even without signing in. Add, Edit, Delete operations and View Product Detail should be allowed only for authenticated users.
2. Users using the application may or may not sign in. It is optional but when the user attempts to perform add or edit operation or view product detail, **the component should prompt the user to sign in** before performing these operations. Use the router transition concepts.

**View Product List (Summary)**

The list of products in the inventory should be displayed. Few fields of the product can be displayed. Provision should be provided to customize which fields can be displayed.

There should also be an option to navigate from this page to add a product or select a product to update or delete.

**View Product Detail**

Clicking on a product in “View Product List” page should navigate to display the complete details about the product.

**Add Product**

While adding the product, the following input details can be accepted.

* Product name
* Product description
* Manufacturer
* Price
* Quantity

Product ID should be generated with a unique ID before adding the product.

**Note:** Use the router transition concepts to notify, if the user accidentally moves out of the “Add Product form page” after inputting some data in the form.

**Update Product**

When the list of products are displayed, one product can be selected to update at a time. The product name, product description, manufacturer, price and quantity can be allowed to be updated.

**Note:** Product ID should be used as the key to look for product details and it cannot be modified.

**Delete Product (s)**

When the list of products are displayed, one or more products can be selected to delete at a time.

**Chart – Top viewed products**

The details of the product which has been viewed often (through the **“View Product detail”**) is considered as a top viewed product. Provision should be there to display a chart of the top viewed products. It can be top 3, 5, 10, etc., which should be customizable.

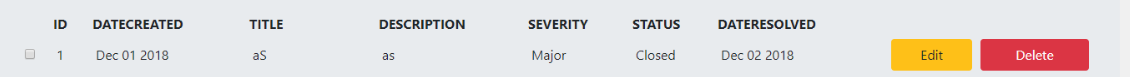
You can refer **chart.js** - <https://www.chartjs.org/> (OR) **react-chartjs-2** - <https://www.npmjs.com/package/react-chartjs-2> to understand how to display a chart.

**Note on Customization of fields**

Some employees are not able to understand the meaning of customization of fields. Go through the below mentioned note to understand what it means.

Note that the UI below is provided only to give an explanation of customization of fields. The explanation is given by showing a table UI as an example. It does NOT mean you should use a table to display the product list. The UI that needs to be done for Capstone Project should be a good looking **responsive** UI (Bootstrap Grid, Card Layout etc.).

“Customization of fields” means on loading the application, it should display **all the fields**  like below.



**You should provide a user interface** for user to customize to say which fields, the user wants to see at any point of time. If you see below, the user has customized to see only  **DATECREATED**, **TITLE** and **DATERESOLVED**.  The UI you provide for customization is up to you.

Note that the user interface to provide for customization has not been intentionally shown as we want each employee to think on their own and come up with a solution. In the earlier batches, different employees have thought in different ways and provided different type of UI. You can decide what type of UI you want to provide for customization.

