**Grocers Project**

**DESCRIPTION**

Create a website that deal with online purchase of groceries.

**Background of the problem statement:**

You work at XYZ solutions as a MEAN stack developer. Due to the Covid-19 situation every business is becoming online. You are asked by your management to lead and develop a dynamic website for online purchase of groceries.

**Implementation requirements:**

1. Create a rich frontend
2. Create admin and user portals
3. Create functionalities for admin portal such as adding, updating, and deleting an item and user.
4. Create functionalities for user portal such as, registration, sign in, purchasing item, adding to the cart, and wish listing items.
5. Create database for storing all application related information.

**The following tools must be used:**

* VS Code
* React JS
* NodeJS
* MongoDB or MySQL
* AWS cloud services for deploying the application

**The following things to be kept in check:**

1. You need to document the steps and write the algorithms in them.
2. The submission of your GitHub repository link is mandatory. In order to track your tasks, you need to share the link of the repository.
3. Document the step-by-step process starting from creating test cases, then executing them, and recording the results.
4. You need to submit the final specification document, which includes:
   1. Project and tester details
   2. Concepts used in the project
   3. Links to the GitHub repository to verify the project completion
   4. Your conclusion on enhancing the application and defining the USPs (Unique Selling Points)

**Use Case Diagram**

Sign in

Admin



Sign up

Sign in

User

**Use Case Diagram for Admin**

Add Products

Sign in

Admin



Logout

Generate Reports

Update Products

Delete Products

User

**Note:**

1. Once anyone signs in all the operations should happen in one single page

2. Validate the form wherever required for invalid inputs

**Admin Panel**

* + Admin will sign in with the already pre-entered credentials present in the database, if the credentials wrong show the error messages and allow admin to re-login
  + After the successful login admin must able to see the Dashboard with the following operations
    - Add products
    - Update products
    - Delete products
    - Generate Reports
    - Logout
  + **Add Products**
    - Admin can add the new products with details like product name, product price, product quantity with the auto-generated product id and etc if necessary
  + **Update Products**
    - Admin can update the products price & quantities
    - Discounts for the particular product if required
  + **Delete Products**
    - Admin can delete the products which has no demands (zero demand in the market)
  + **Generate Reports**
    - Admin can view the reports that are ordered by users
    - Admin must able to customize the reports like
      * Daily reports
      * Weekly reports
      * Monthly reports
      * Particular product reports
      * Particular customer reports
  + **Logout**
    - Log out from the application

**Use Case Diagram for User**

Admin



Part of the Cart

Select Items

View Items

Checkout

Delete Items

Edit Profile

Order Status

Funds

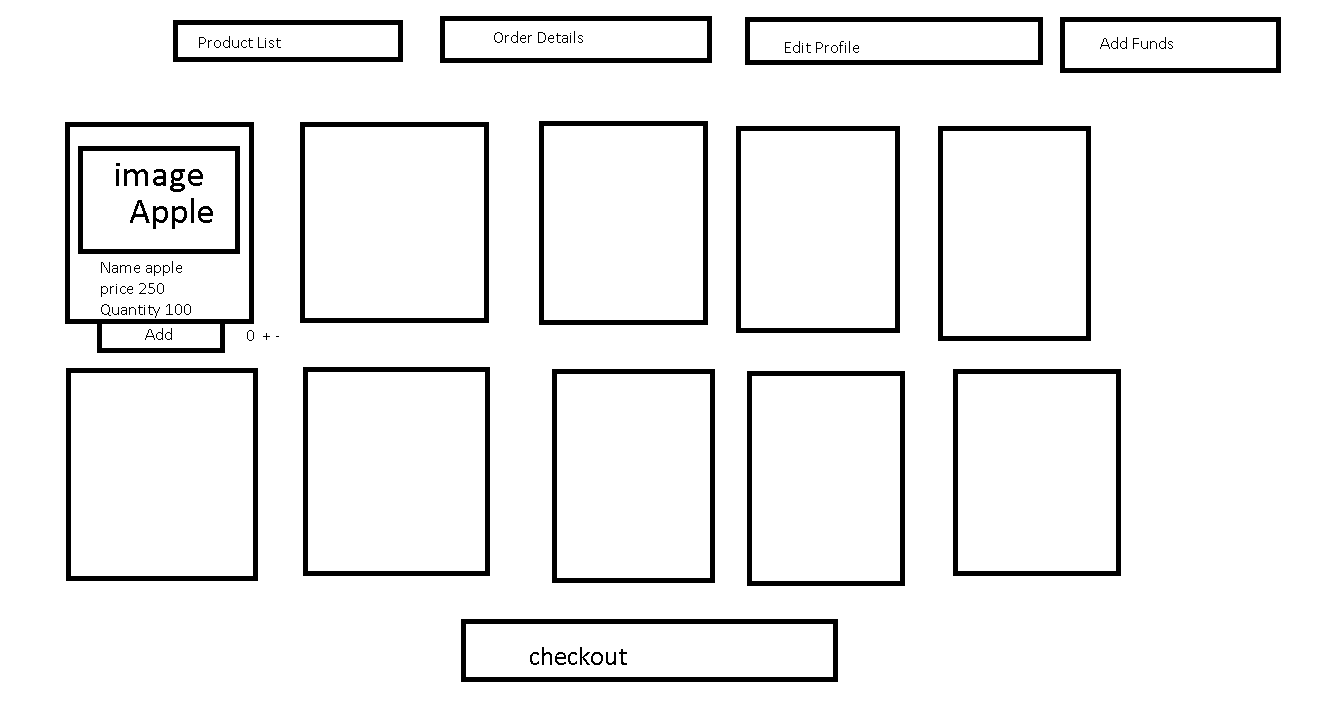
Sign up

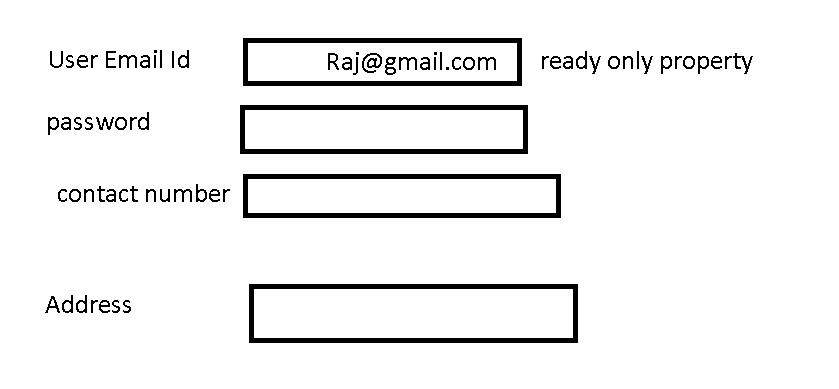
Sign in

User

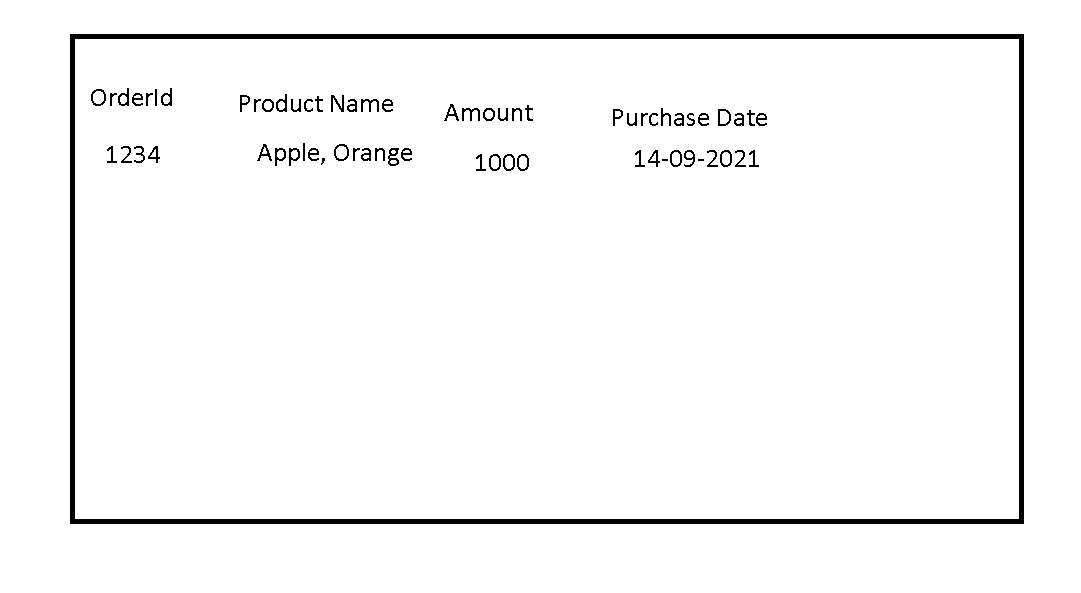
Log out

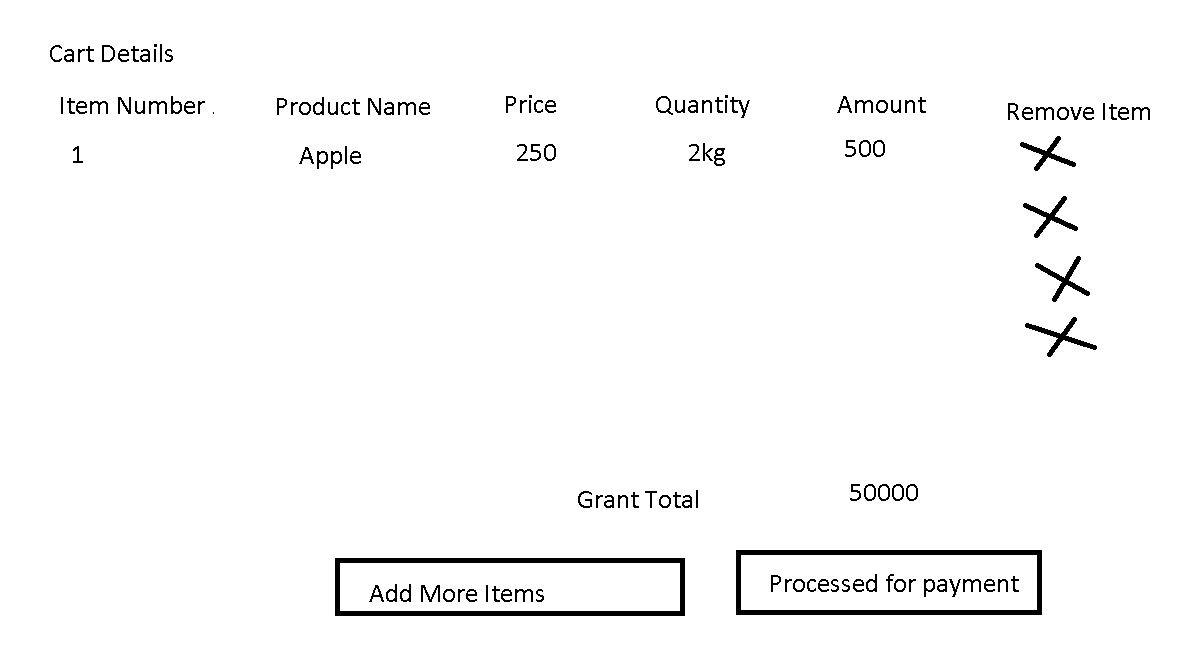
**Sample of user dashboard**

****

****

**Add funds**

****

****

**User Panel**

**Sign up:**

* + User can signup entering details like firstname, lastname, email, password, dob, phone no, address and user id should be auto-generated
  + Once user registers, user should able to see the auto-generated id using that user can sign in
  + Once user registers there should be some default amount to purchase items

**Sign in:**

* + Sign in will take user id and password, on successful authentication go to the dashboard of the user else show the error messages and allow user to re-login
  + If the user logs in successfully the application takes to the dashboard where the user can do the following operations:
    - Select items to the cart
    - Delete items from the cart
    - Update items quantity
    - View items in the cart
    - Checkout
    - Order Status
    - Edit Profile
    - Funds
    - Logout
  + **Select Items:**
    - Use must able to add items to the cart
  + **Delete items:**
    - User must able to delete items from the cart
  + **Update items:**
    - User can update the quantities
  + **View items in the cart:**
    - User could able to see all the items added in the cart
  + **Checkout:**
    - Once all the items are selected user can checkout to order the items, which will deduct the amount if the money is present in the funds else user gets the error message insufficient amount, if the sufficient money is present in the funds, order must be placed successfully
  + **Order Status:**
    - User can see the history of orders whichever items he/she ordered.
  + **Edit Profile:**
    - User must able to edit the profile like password, address, phone number, email id and etc
  + **Funds:**
    - Funds will show the available amount to purchase the items
    - User can add money by entering the account number (read only) and amount, all the users will have some account number with balance in their dummy bank account.
    - The amount user adds will be added to the profile funds which is utilized at the time of ordering the items
  + **Logout:**
    - Logout from the application

**Important Notes:**

* Validate the forms and display the appropriate error messages
* Style the components to have a good look & feel
* Generate the primary key’s where ever is necessary
* After the successful login all the operations must happen in a single page mandatorily
* Create one dummy account that will have account number and amount so that users can add money to the funds
* All the requirements in the use case must be met, however adding additional features after meeting all the requirements is welcome.

Steps to submit in capstone project

First you create your project

both project you have to keep in one folder.

frontend : react backend : express with mongoose with mongodb

push this project in github account.

Writeup \*

create world file or document file

1. provide the git URL

2. please provide the file structure of your project mean backend folder and

frontend folder details.

3. steps to run the application.

screen

1. screen short of running application

2. first page

3. login with invalid username and password what error message it will display.

4. after sucessfully login it display dashboards.

5. display project details.

6. add product

etc keep in word file or any document file if you want convert it into pdf.

source code

1. please delete node\_module from both folder ie backend and frontend.

2. if you want to download the node\_module folder . you have to open the command

prompt in project location and run the command as npm install. This command read all instruction avaiable

in package.json file and base upon the dependency tag it download in local machine.