**Online**

**Grocery**

**Store**

**Web Application**

**(Using MERN)**

**Team Members :**

>>> Mopuri Lokesh Dhananjayan -

NM ID:**6E85DBC95345240B33347BF472F41076**

>>>Kevin Easter Raj A –

NM ID:**C46478122902779BDA84E18801250BBA**

>>> Mohammed Asraf Ali S -

NM ID:**1595E4439D60887BB11443CF3276E5A2**

>>> Saranya A –

NM ID:**1CA252CBF516F00A1254BA08BE17BFA0**

**GitHub repository:**[**https://github.com/mohammedasrafali/Online-grocery-shop-.git**](https://github.com/mohammedasrafali/Online-grocery-shop-.git)

**Introduction to Grocery Store Apps :-**

In today’s digital age,grocery store apps are transforming the way we shop. These applications enhance the Shopping experience by offering convenience,personalization and efficiency .This presentation explores the future of these apps and their impacts on consumer behavior and retail strategies.



**Integration with Smart Technology:-**

* The integration of grocery apps with smart technology is revolutionising shopping.

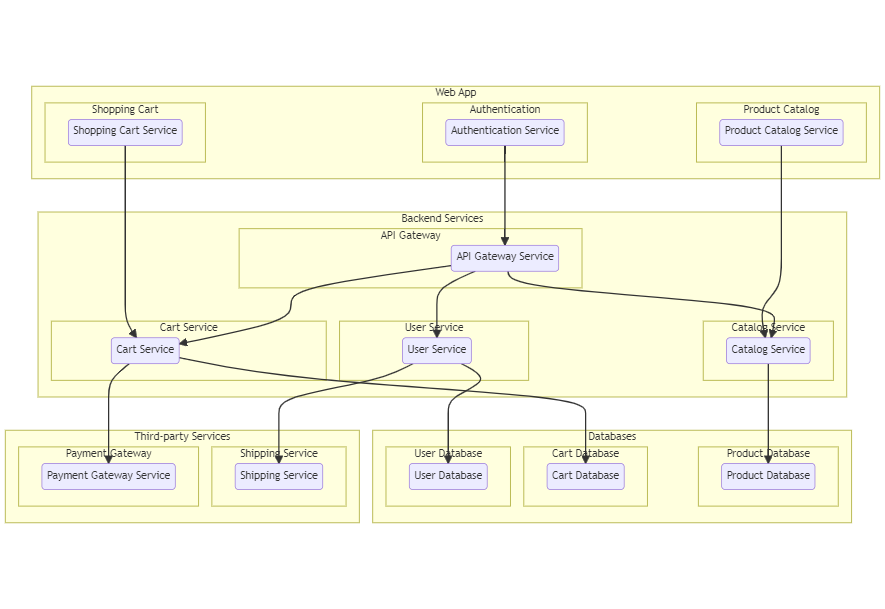
* Features like voice commands, smart carts, and automated checkouts streamline the shopping process.
* This innovation not only saves time but also enhances the overall shopping experience for consumers.





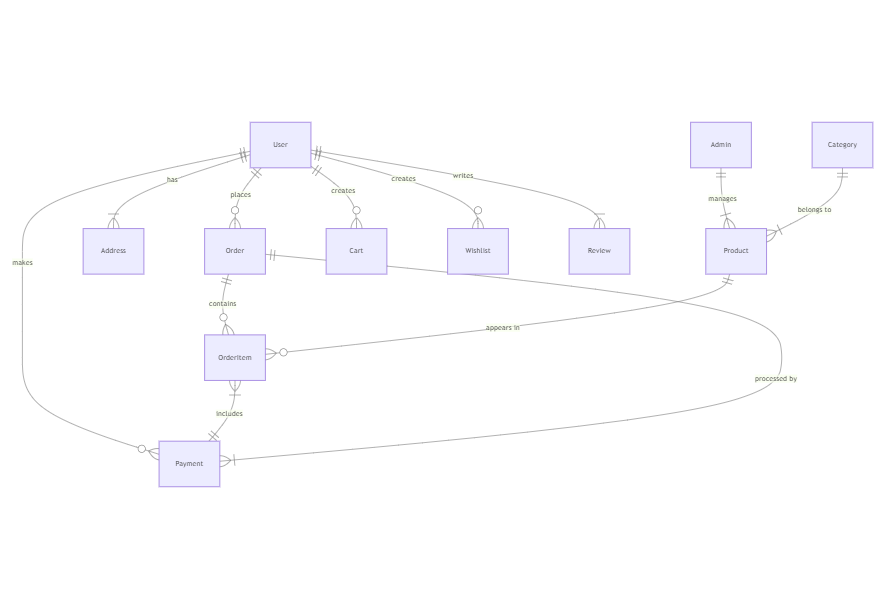
**ARCHITECTURE**

**Technical Architecture:**

****

* **The technical architecture of an flower and gift delivery app typically involves a client-server model, where the frontend represents the client and the backend serves as the server.**
* **The frontend is responsible for user interface, interaction, and presentation, while the backend handles data storage, business logic, and integration with external services like payment gateways and databases.**
* **Communication between the frontend and backend is typically facilitated through APIs, enabling seamless data exchange and functionality.**

**ER DIAGRAM:**

****

* **The Entity-Relationship (ER) diagram for an flower and gift delivery app visually represents the relationships between different entities involved in the system, such as users, products, orders, and reviews.**
* **It illustrates how these entities are related to each other and helps in understanding the overall database structure and data flow within the application.**

**Project Flow**

* **Frontend Development**
* **Backend Development**
* **Integration**

**Frontend Development:**

* **Frontend development involves building the user interface (UI) and implementing the visual elements of the online shopping web application.**
* **It focuses on creating an intuitive and engaging user experience that allows users to interact with the application seamlessly.**

**Backend Development:**

* **Backend development involves building the server-side components and logic of the online shopping web application.**
* **It focuses on handling the business logic, processing requests from the front end, and interacting with the database.**

**Integration:**

* **Integration is the process of combining and connecting the frontend and backend components of the online flower shop web application to create a unified and fully functional system.**
* **It involves establishing communication channels, exchanging data, and ensuring seamless interaction between the frontend UI and backend APIs**

**Current Trends in Grocery Apps :-**



* Grocery apps are embracing technology to improve user experience.
* Features like online ordering, real-time inventory, and digital coupons are becoming standard.
* These trends cater to the modern shopper's desire for speed and efficiency, making grocery shopping more accessible than ever.

**Personalization and User Experience :-**



* The future of grocery store apps lies in personalization.
* By utilising data analytics, these apps can offer tailored recommendations and promotions based on individual shopping habits.
* This level of customization enhances the user experience, fostering customer loyalty and satisfaction.

**Challenges and Opportunities :-**



* While grocery store apps present numerous opportunities, they also face challenges like data security and user adoption.
* Addressing these issues is crucial for maximising the potential of these apps and ensuring a seamless shopping experience for customers .



**SOURCE CODE :-**

**App.jsx :-**

import { useState } from 'react'

import HomePage from './pages/HomePage'

import {BrowserRouter,Routes,Route} from 'react-router-dom';

import ShopPage from './pages/ShopPage';

import ShopDetailPage from './pages/ShopDetailPage';

import ContactPage from './pages/ContactPage';

import Cart from './pages/Cart';

import Checkout from './pages/Checkout';

import TestimonialPage from './pages/TestimonialPage';

import ErrorPage from './pages/ErrorPage';

import SignUpPage from './pages/SignUpPage';

import LoginPage from './pages/LoginPage';

import Home from './pages/Admin/home';

import AdminAddProducts from './pages/Admin/AdminAddProducts';

import AdminUpdateProducts from './pages/Admin/AdminUpdateProducts';

import { CartProvider } from "react-use-cart";

function App() {

  return (

    <>

    <CartProvider>

      <BrowserRouter>

        <Routes>

          <Route path="/" element={<HomePage/>}></Route>

          <Route path="/login" element={<LoginPage/>}></Route>

          <Route path="/signup" element={<SignUpPage/>}></Route>

          <Route path="/shop" element={<ShopPage/>}></Route>

          <Route path="/shopDetail" element={<ShopDetailPage/>}></Route>

          <Route path="/contact" element={<ContactPage/>}></Route>

          <Route path="/cart" element={<Cart/>}></Route>

          <Route path="/checkout" element={<Checkout/>}></Route>

          <Route path="/testimonial" element={<TestimonialPage/>}></Route>

          <Route path="/error" element={<ErrorPage/>}></Route>

          <Route path="/adminProducts" element={<Home/>}></Route>

          <Route path="/adminAddProducts" element={<AdminAddProducts/>}></Route>

          <Route path="/adminUpdateProducts/:id" element={<AdminUpdateProducts/>}></Route>

        </Routes>

      </BrowserRouter>

    </CartProvider>

    </>

  )

}

export default App

**Home Page.jsx :-**

import React, { useState, useEffect } from 'react';

import NavBar from '../components/NavBar'

import Spinner from '../components/Spinner'

import Footer from '../components/Footer'

import Banner from '../components/Banner'

import BestSellerProduct from '../components/BestSellerProduct'

import Feature from '../components/Feature'

import FeaturedProducts from '../components/FeaturedProducts'

import Fact from '../components/Fact'

import FruitShop from '../components/FruitShop'

import Hero from '../components/Hero'

import ModalSearch from '../components/ModalSearch'

import Testimonial from '../components/Testimonial'

import VegitableShopTest from '../components/VegitableShopTest'

import ButtonToTop from '../components/ButtonToTop'

//import initializeMainScript from '../script/mainFunction'

import axios from 'axios';

import ProductCard from '../components/ProductCard';

export default function HomePage() {

  //home pages

  const [categories, setCategories] = useState([]);

  const fetchCategories = async () => {

    try {

      const response = await axios.get('http://localhost:5000/api/categories');

      setCategories(response.data);

      console.log('Categories API Response:', response.data);

    } catch (error) {

      handleFetchError('categories', error);

    }

  };

  const handleFetchError = (type, error) => {

    console.error(`Error fetching ${type}:`, error);

  };

  useEffect(() => {

    // Initialize your main script when the component mounts

    // initializeMainScript();

    fetchCategories();

  }, []);

 // Empty dependency array means this effect runs once after the initial render

  // useEffect(() => {

  //   console.log('Categories:', categories);

  // }, [categories]);

  // useEffect(() => {

  //   console.log('Products:', products);

  // }, [products]);

  return (

    <div>

      <Spinner/>

      <NavBar/>

      <ModalSearch/>

      <Hero/>

      <Feature/>

      <div className="col-lg-4 mx-auto text-center">

          <h1>Our Organic Products</h1>

      </div>

      <FruitShop/>

      <FeaturedProducts/>

      <VegitableShopTest/>

      <Banner/>

      <BestSellerProduct/>

      <Fact/>

      <Testimonial/>

      <Footer/>

      <ButtonToTop/>

    </div>

  )

}

**Product Page :-**

import React from "react";

import NavBar from "../components/NavBar";

import Spinner from "../components/Spinner";

import Footer from "../components/Footer";

import ModalSearch from "../components/ModalSearch";

import Categories from "../components/Categories";

import SideFeaturedProducts from "../components/SideFeaturedProducts";

import { Link } from "react-router-dom";

import FruitShop from "../components/FruitShop";

export default function ShopPage() {

  return (

    <div>

      <Spinner />

      <NavBar />

      <ModalSearch />

      {/\* Single Page Header start \*/}

      <div className="container-fluid page-header py-5">

        <h1 className="text-center text-white display-6">Shop</h1>

        <ol className="breadcrumb justify-content-center mb-0">

          <li className="breadcrumb-item">

            <Link to="/">Home</Link>

          </li>

          <li className="breadcrumb-item">

            <a href="#">Pages</a>

          </li>

          <li className="breadcrumb-item active text-white">Shop</li>

        </ol>

      </div>

      {/\* Single Page Header End \*/}

      {/\* <FruitShop/> \*/}

      {/\* Fruits Shop Start\*/}

      <div className="container-fluid fruite py-5">

        <div className="container py-5">

          <h1 className="mb-4">Fresh fruits shop</h1>

          <div className="row g-4">

            <div className="col-lg-12">

              {/\* <div className="col-6" /> \*/}

              <div className="row g-4">

                <div className="col-lg-3">

                  <div className="row g-4">

                    <div className="col-lg-12">

                      <Categories />

                    </div>

                    {/\* Price slider starts \*/}

                    <div className="col-lg-12">

                      <div className="mb-3">

                        <h4 className="mb-2">Price</h4>

                        <input

                          type="range"

                          className="form-range w-100"

                          id="rangeInput"

                          name="rangeInput"

                          min={0}

                          max={500}

                          defaultValue={0}

                          oninput="amount.value=rangeInput.value"

                        />

                        <output

                          id="amount"

                          name="amount"

                          min-velue={0}

                          max-value={500}

                          htmlFor="rangeInput"

                        >

                          0

                        </output>

                      </div>

                    </div>

                    <SideFeaturedProducts />

                  </div>

                </div>

                <div className="col-lg-9">

                  <div className="row g-4 justify-content-center">

                    <FruitShop />

                    {/\* <div className="col-12">

                      <div className="pagination d-flex justify-content-center mt-5">

                        <a href="#" className="rounded">

                          «

                        </a>

                        <a href="#" className="active rounded">

                          1

                        </a>

                        <a href="#" className="rounded">

                          2

                        </a>

                        <a href="#" className="rounded">

                          3

                        </a>

                        <a href="#" className="rounded">

                          4

                        </a>

                        <a href="#" className="rounded">

                          5

                        </a>

                        <a href="#" className="rounded">

                          6

                        </a>

                        <a href="#" className="rounded">

                          »

                        </a>

                      </div>

                    </div> \*/}

                  </div>

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

      {/\* Fruits Shop End\*/}

      <Footer />

    </div>

  );

}

**Key Features :-**

\* Bootstrap 5 - Utilises the latest Bootstrap framework for a responsive and modern layout.

\* HTML5 & CSS3 - Employs cutting-edge web technologies for enhanced functionality and aesthetics

\* Burger Menu Offers a convenient and sleek navigation menu for easy access to site sections.

\* Breadcrumbs Provides users with clear navigation paths for seamless browsing. A

\* Sticky Top Navigation Ensures easy access to navigation options as users scroll through the page.

\* Drop-Down Menu Organizes site content for intuitive navigation and exploration.

\* Split Header - Divides the header section to highlight key information and promotions.

\* Hero Image - Captivates users with visually striking imagery upon entering the website.

\* Header Carousel Showcases featured products or promotions in an interactive carousel.

\* Header Search Option Enables users to quickly find products or categories of interest.

\* Tabbed Content into tabs comparison. for Organizes information easy browsing and

\* Product Cards Presents products in visually appealing and informative card layouts.

\* Product Carousels Displays product collections or categories in dynamic carousels.

\* Sales Banner Highlights ongoing sales or promotions to attract user attention.

\* Testimonial Carousel - Showcases customer testimonials to build trust and credibility.

\* Newsletter Subscription UI Allows users to subscribe to newsletters for updates and promotions.

\* Category Search Filter - Facilitates product discovery by enabling users to filter by category. A

\* Pricing Filter Enables users to filter products based on price range.

\* Pagination UI Provides easy navigation through product listings with pagination controls.

\* Comment Form UI Allows users to leave feedback or reviews on products or services.

\* Back-to-top Button Offers a convenient way for users to return to the top of the page.

\* Geolocation Utilizes geolocation services for personalized user experiences .

\*Contact Form UI - Provides a user-friendly interface for contacting customer support or making inquiries.

\* Detailed Footer Includes comprehensive information and links for user convenience.

\* Footer Navigation Offers additional navigation options in the footer for easy access to site sections.

\* Clean Codebase Maintains a well- structured and organized codebase for easy maintenance and scalability.

\*SEO-friendly Theme - Optimized for search engines to improve visibility and ranking.

\* Page-speed Optimized Ensures fast loading times for improved user experience and SEO performance.

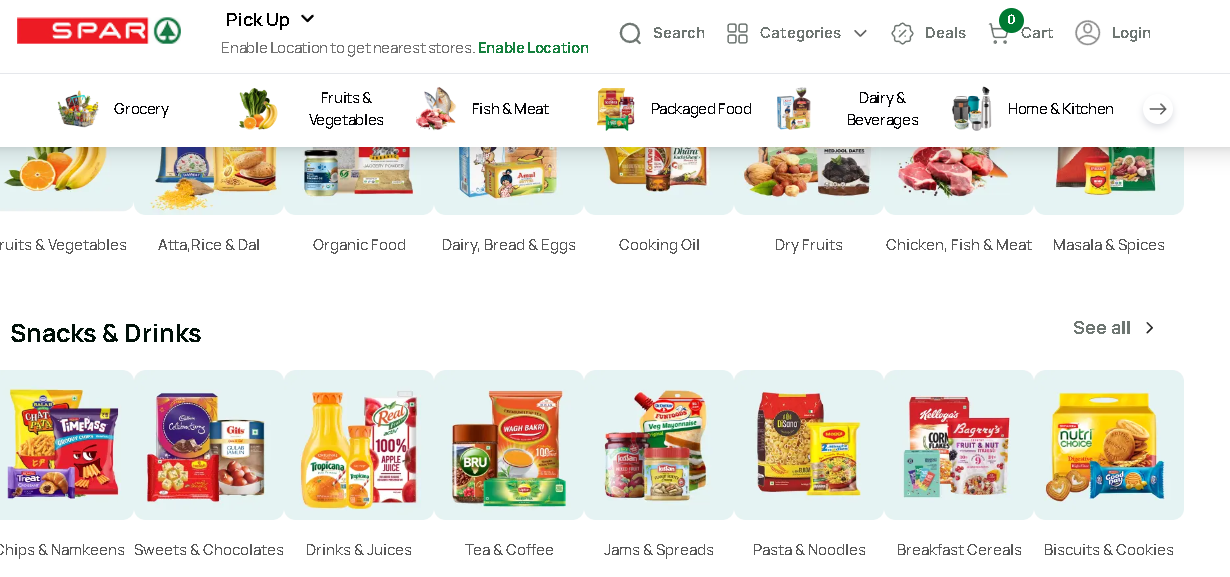
\* Multipage Design Structured with multiple pages for comprehensive content organization.

\* 100% Responsive various screen Adapts seamlessly to sizes and devices for optimal viewing experience.

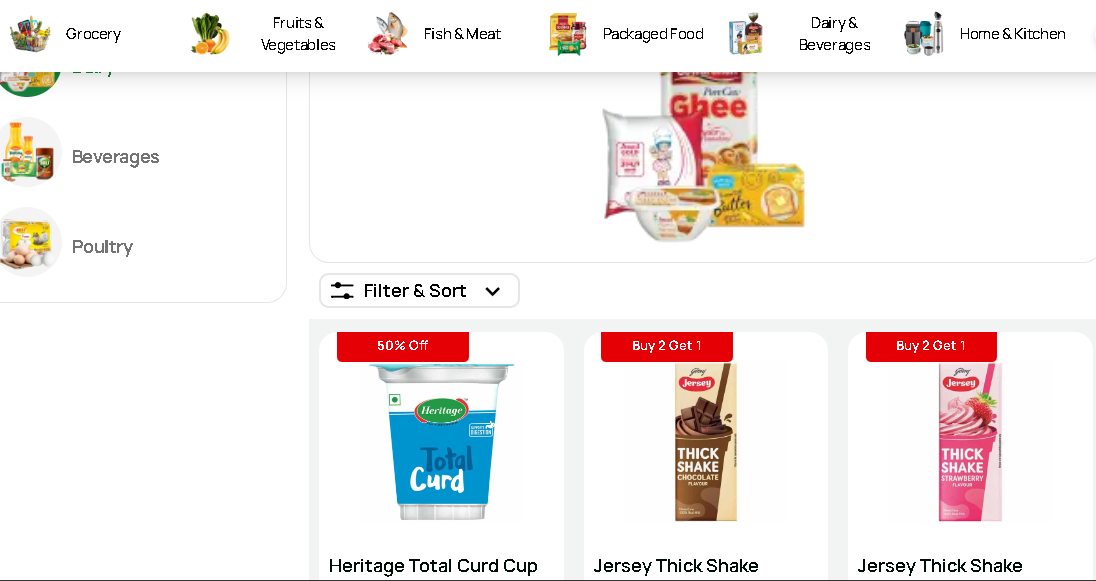
\* Cross-browser Compatible compatibility across Ensures different web browsers for broader accessibility.

\* Admin Panel Allows administrators to dynamically add, delete, and update products on the website, providing flexibility and ease of management.

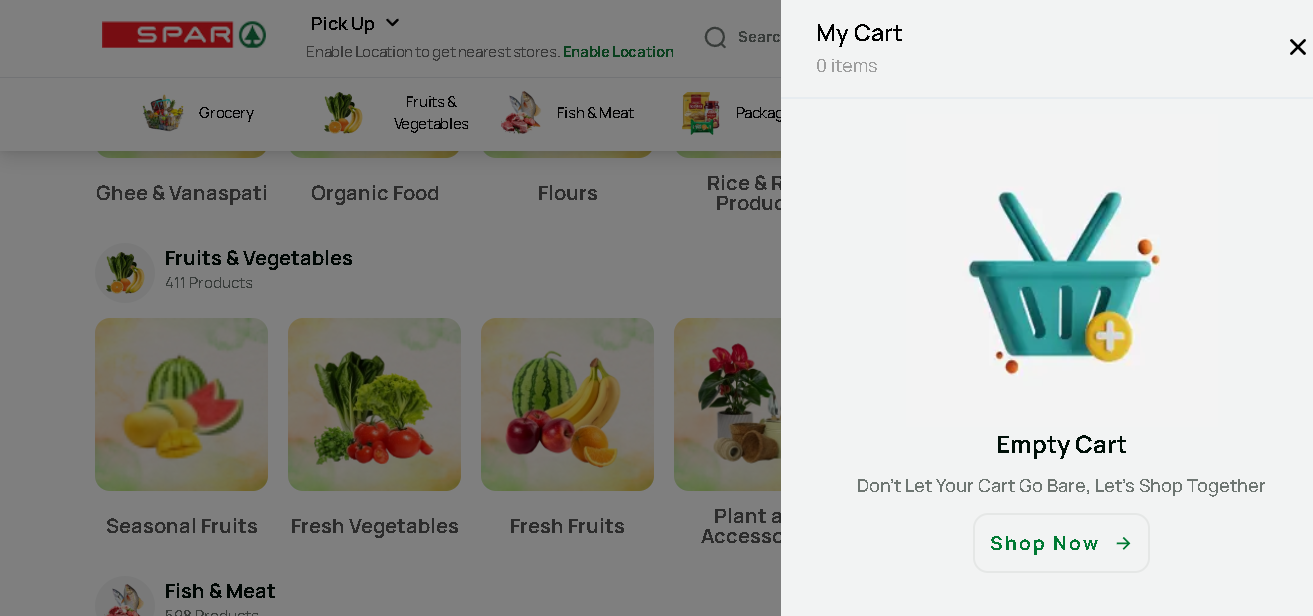
**HOME PAGE :-**

****

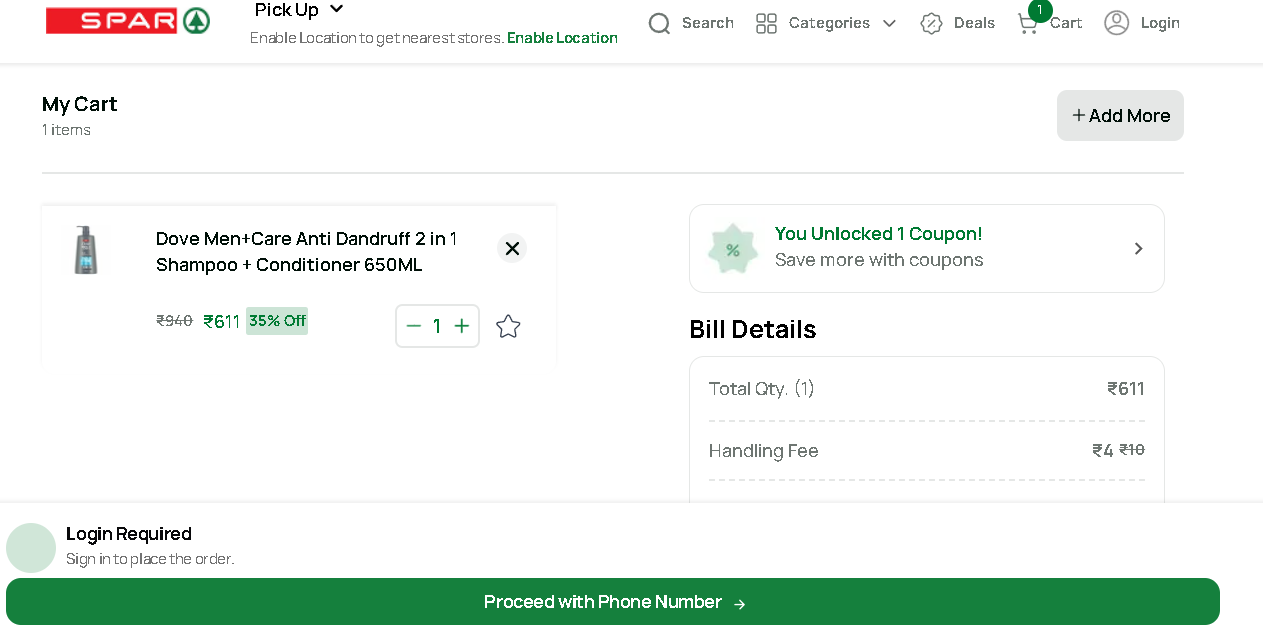
**PRODUCT PAGE :-**

****

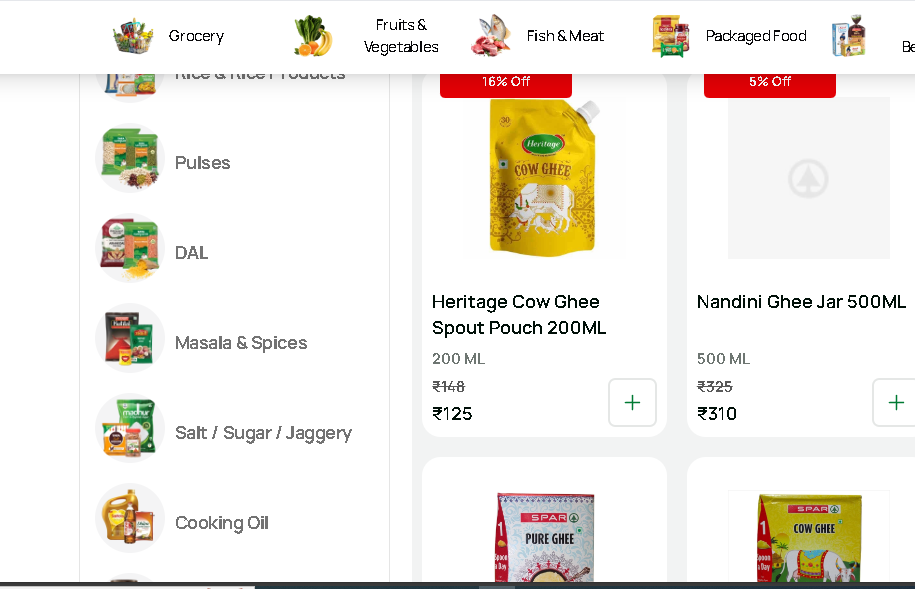
**CART PAGE :-**

****

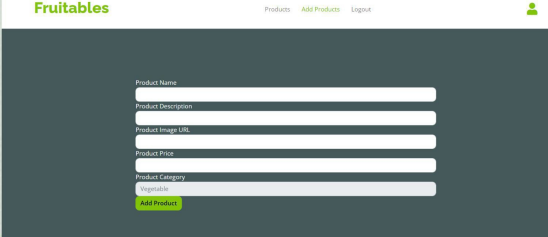
**PAYMENT PAGE :-**

****

**PRODUCT LIST :-**

****

**ADD PRODUCT :-**

****

**Conclusion :-**

\* The main theme is to build an online Grocery Store Web application with all three i.e., Front end,back end and Database.

\* This Web application is a fully pledged working web application right from the authentication,admin authorization,add to cart using payment gateway. login item

\* The web application is easy for them to access and without any effort categories can be created

**Reference :**

**1.React -** [**https://react.dev/**](https://react.dev/)

**2.NPM -** [**https://www.npmjs.com/**](https://www.npmjs.com/)

**3.Node JS -** [**https://nodejs.org/en**](https://nodejs.org/en)

**4.ExpressJS-** [**https://expressjs.com/**](https://expressjs.com/)

**5.Mongodb-**[**https://learn.mongodb.com/**](https://learn.mongodb.com/)

**“Thanks….!”**