Green Grocery

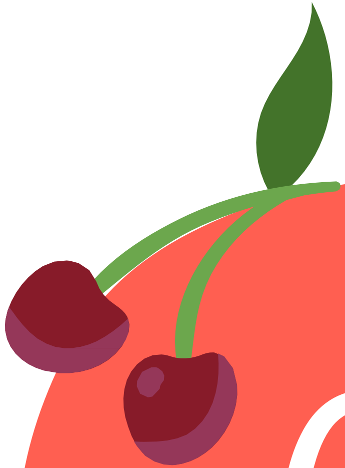
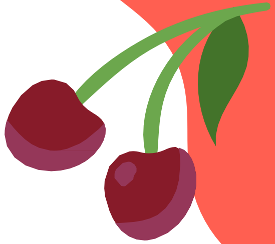
An Online Grocery Store

Project Business Cases:

Problem Motivation

01

The main motivation of the project is to help the farmers to sell their products at a fair price and to provide customers healthy and fresh vegetables, fruits and dairy products at low prices.

Problem Statement

In these present situations, it is difficult for farmers to sell their products at a fair price in markets. Customers are buying hybrid fruits and vegetables which are not good for health which contains harmful substances. Customers are buying products at high price than market price.

Problem Solution

Farmers can sell fruits, vegetables and dairy products at fair market prices on our website and customers can buy directly organic products from farmers through the website which are fresh and healthy for the body.

Project Overall Idea :

Farmers can contact GreenGrocery website and can sell their vegetables, fruits and dairy products at fair prices. The process includes, firstly, the delivery guy pick their products and the management staff are responsible for updating these products on the website. Users can login into the website and can buy organic fruits, vegetables and dairy products which are very fresh and healthy that came directly from farmers. Users can then select their required products and respective quantity of each product and add them to the cart. After adding to the cart, User can checkout and fill in the shipping details and enter payment details. After successful transaction, user will get order confirmation and tracking link. Users can resolve their queries in the Contact Us section.

Unique features of website & How this website is different from Other Websites?

- *Unique and friendly UI / UX*
- *There is no website with similar idea or similar UI / UX*
- *Many websites sell the Hybrid products, but we are selling natural organic healthy products.*
- *This website directly contact with farmers and take products directly from them with no intermediate. No other website is currently doing it.*
- *Free of bugs*
- *Give Farmer a gift every month who donates the more quality of vegetables,fruits and dairy products*
- *Rapid and safe COVID preventive measures taken care while delivery*
- *Management team will personally go and make dealership with farmers for wide range of varieties of products*
- *This website will provide funds for farmers by giving some share of profit collected through the website.*

Function Description

Modules : 8

Module 1: UI/UX Design

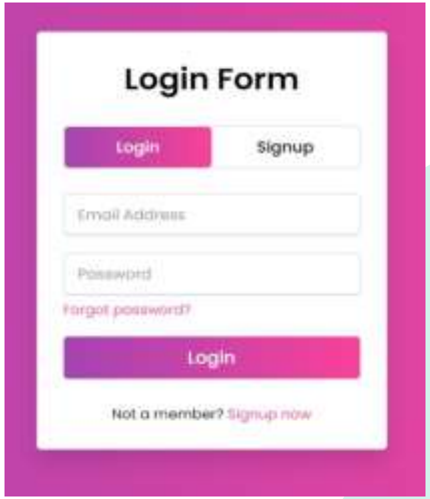
This is the first and important module in developing our website. An interactive UI/UX makes customer satisfied.

Module 2: Login/SignUp

The existing users of our website can Log In through username and password in this section. Now, If you are a new user around here, you can sign up by providing name, mail-id and username. The user-id and password will be added to database and will be maintained securely through some hash and salt techniques

Module 3: Authentication

We have to use certain API's for connecting to the application and providing an Interface for Authentication Purpose.


Login Form

Login Signup

Email Address

Password

Forgot password?

Login

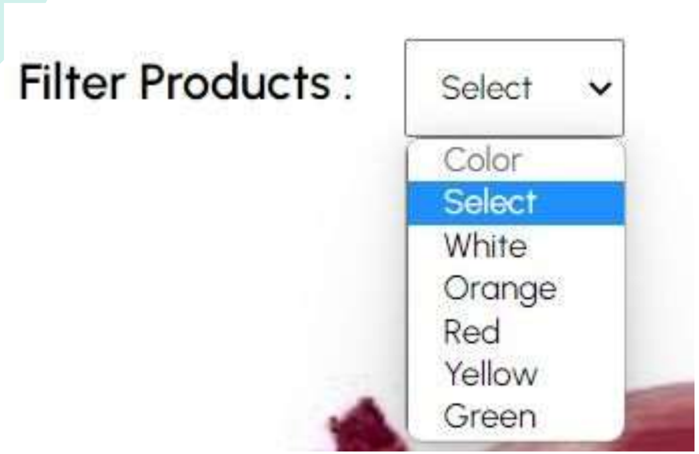
Not a member? Signup now

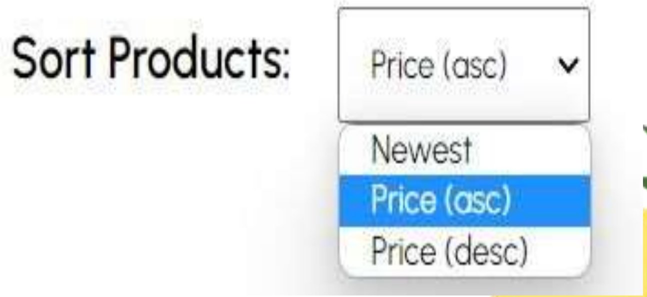
Module 4: Database Management

We used MongoDB for maintaining the back-end of our website. By this, we connected to database and perform CRUD operations and maintain user records and farmer products and contact us queries.

Module 5: Product Categories and Navigation

In this module, color and prices are displayed using a drop down menu and the user can navigate through the category they want and filter/sort the products.





Module 6: User Needs

On Choosing:

- ˙**Shop Vegetables**” will lead to where there will be vegetable products along with filter. The users can now select the desired vegetables and their quantity and add them to the cart
- “**Shop Fruits**” will lead to where there will be Fruit products along with filter. The users now can now select the desired vegetables and their quantity and add them to the cart
- “**Shop Dairy Products**” will lead to where there will be Dairy products along with filter. The users can now select the desired vegetables and their quantity and add them to the cart.

Module 7: Product Page

- ❑ In this page, one can find all particular category products like vegetables or fruits or dairy products along with filters.

- ❑ For each vegetable ,there are details of the product in this section with product name, description, price and selection of quantity.

- ❑ The products can be added to the cart proceeding with a checkout form including shipping address and then the payment of the order can be done.

Module 8: Deployment and Hosting

- ➢ We will test our website for 2-3 weeks before deploying it for finding any bugs in the functionality of website.

- ➢ We currently have hosted our website on GitHub and will deploy it on cloud based services such as Heroku.

Preliminary Work

Before Starting Project

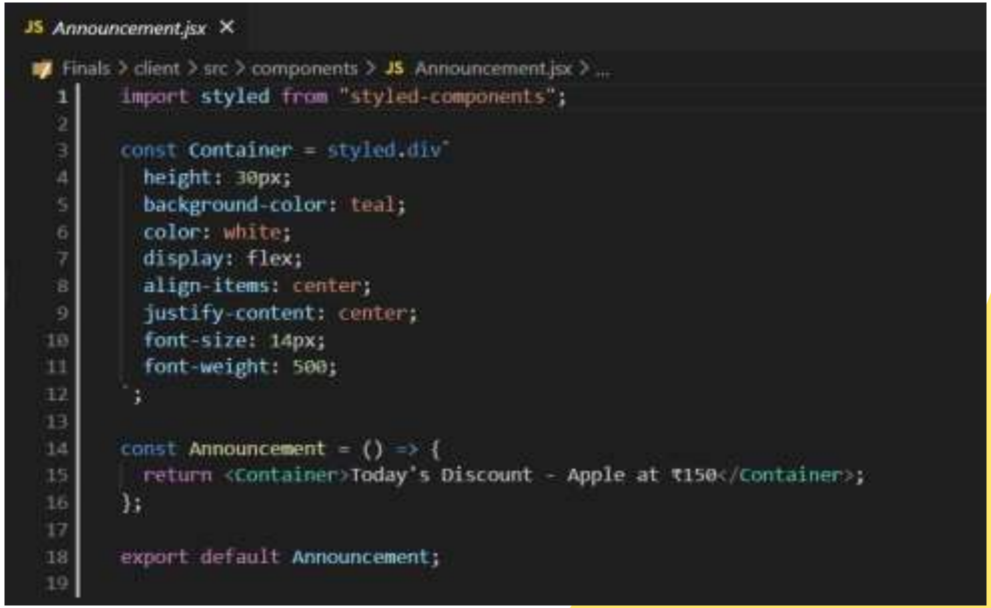
- ❖ We drew an outline of our website
- ❖ We searched for places where people require the organic food most
- ❖ We will tie-up with delivery companies
- ❖ We will collaborate with other websites
- ❖ We will initially collect few products and provide them as a catalog to the users

React Components Used

This is an example of Functional Component used. By writing a javascript function, we can create a functional components in React App.

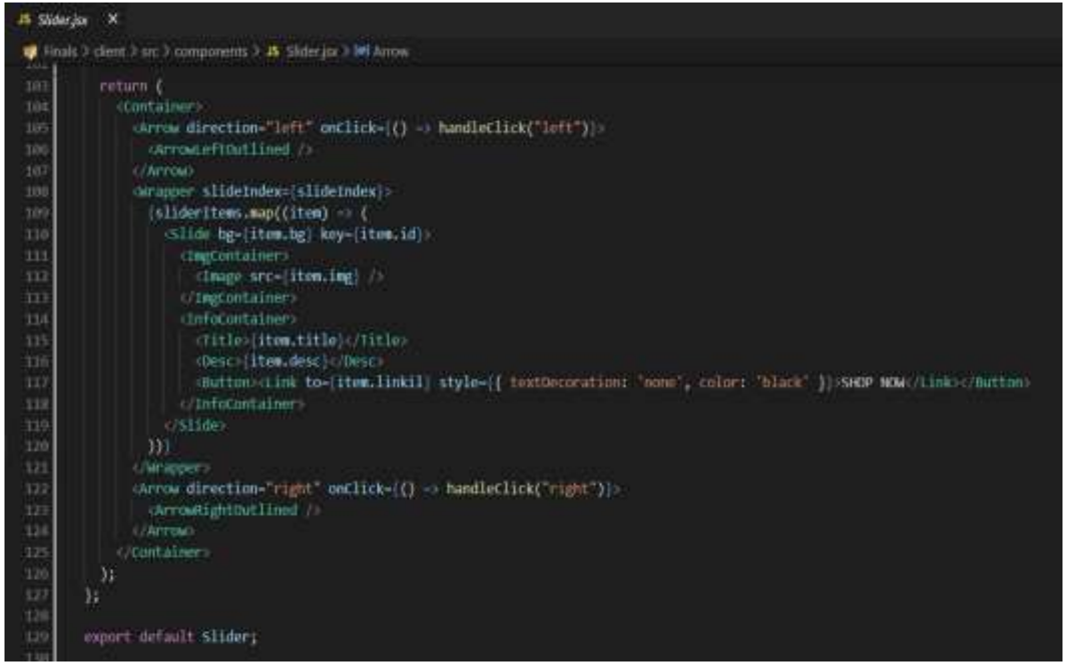
Functional Components used :

Announcement
Navbar
Slider
Categories
Products
Newsletter
Footer



```
import styled from "styled-components";

const Container = styled.div`
  height: 30px;
  background-color: teal;
  color: white;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 14px;
  font-weight: 500;
`;

const Announcement = () => {
  return <Container>Today's Discount - Apple at ₹150</Container>;
};

export default Announcement;
```

- React Higher-Order Components are popular advanced React pattern to deploy reusable logic and functionality across React components.
- Here, in the slider component we used the function takes an array, usually from state which was retrieved from backend with ***fetch*** or from props passed down from parent component, and ***maps*** each element in the array, transforming each element in the array into a React component.


Slider.jsx

Finals > client > src > components > Slider.jsx > Arrow

```
    return (
      <Container>
        <Arrow direction="left" onClick={() => handleClick("left")}>
          <ArrowLeftOutlined />
        </Arrow>
        <Wrapper slideIndex={slideIndex}>
          {sliderItems.map((item) => (
            <Slide bg={item.bg} key={item.id}>
              <ImgContainer>
                <Image src={item.img} />
              </ImgContainer>
              <InfoContainer>
                <Title>{item.title}</Title>
                <Desc>{item.desc}</Desc>
                <Button><Link to={item.linkil} style={{ textDecoration: 'none', color: 'black' }}>SHOP NOW</Link></Button>
              </InfoContainer>
            </Slide>
          ))}
        </Wrapper>
        <Arrow direction="right" onClick={() => handleClick("right")}>
          <ArrowRightOutlined />
        </Arrow>
      </Container>
    );
  };

export default Slider;
```

React-UI Elements used:

React Material -UI

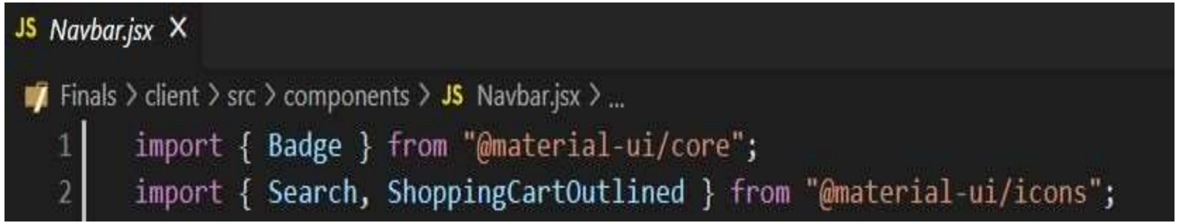
❑ React Material-UI is a set of React components that implements Google's Material Design.

❑ Material-UI is a library that allows us to import and use different components to create a user interface in our React application

❑ Material UI comprises lots of configurable and accessible UI widgets and saves a lot of time.

We used Material-ui icons in many components. For example, In Navbar component we used for Search, Badges at the top right corner in Navbar.

Imports :

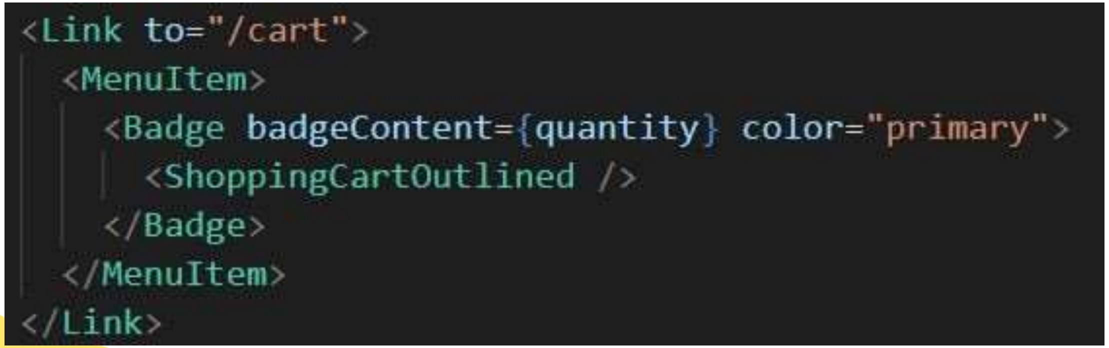


Navbar.jsx

Finals > client > src > components > Navbar.jsx > ...

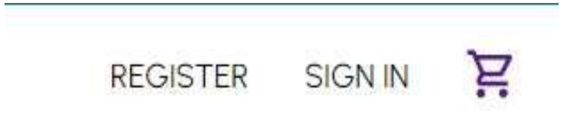
```
import { Badge } from "@material-ui/core";
import { Search, ShoppingCartOutlined } from "@material-ui/icons";
```

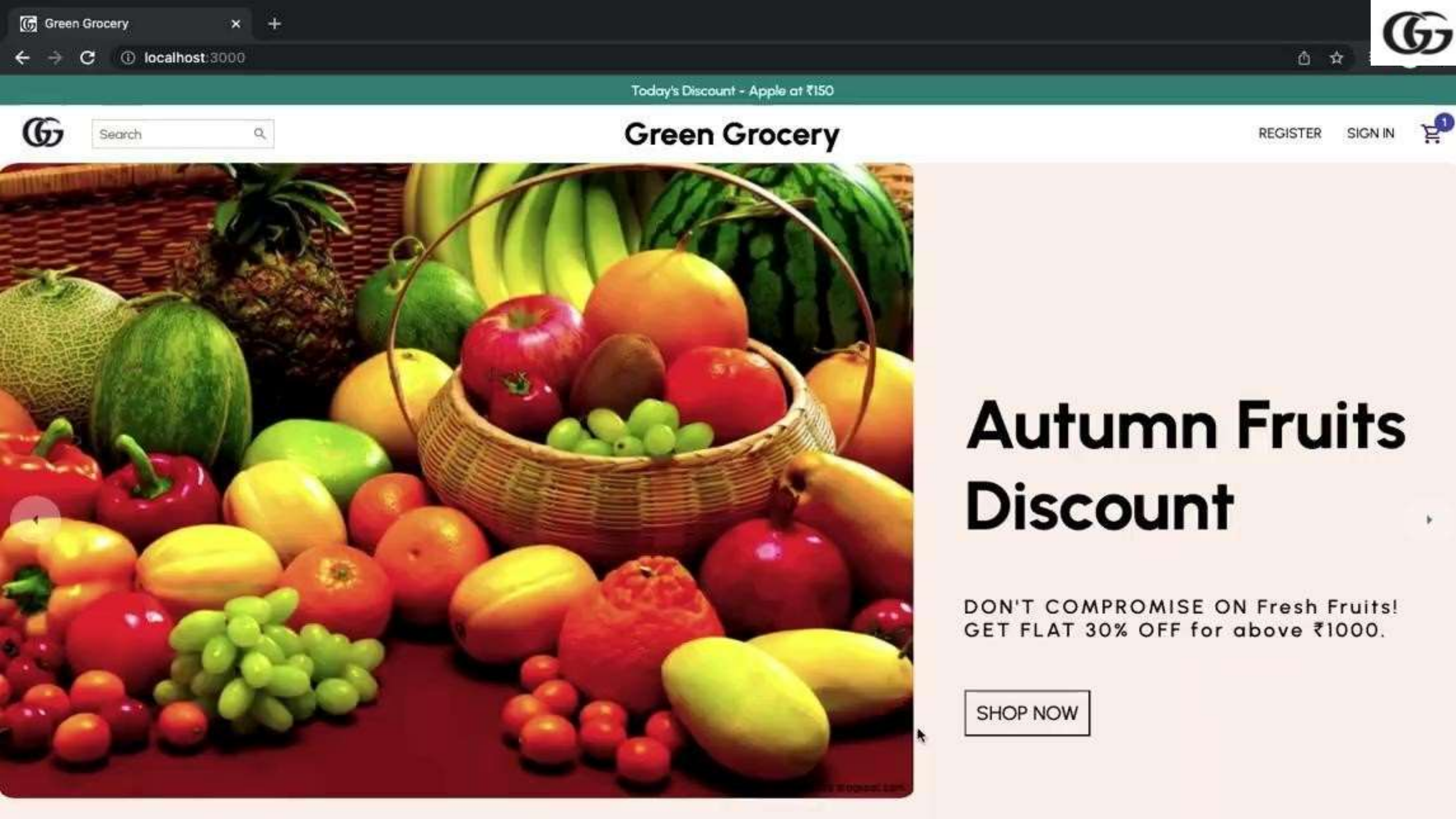
In code customization :



```
<Link to="/cart">
  <MenuItem>
    <Badge badgeContent={quantity} color="primary">
      <ShoppingCartOutlined />
    </Badge>
  </MenuItem>
</Link>
```

Rendered on Website :





Today's Discount - Apple at ₹150

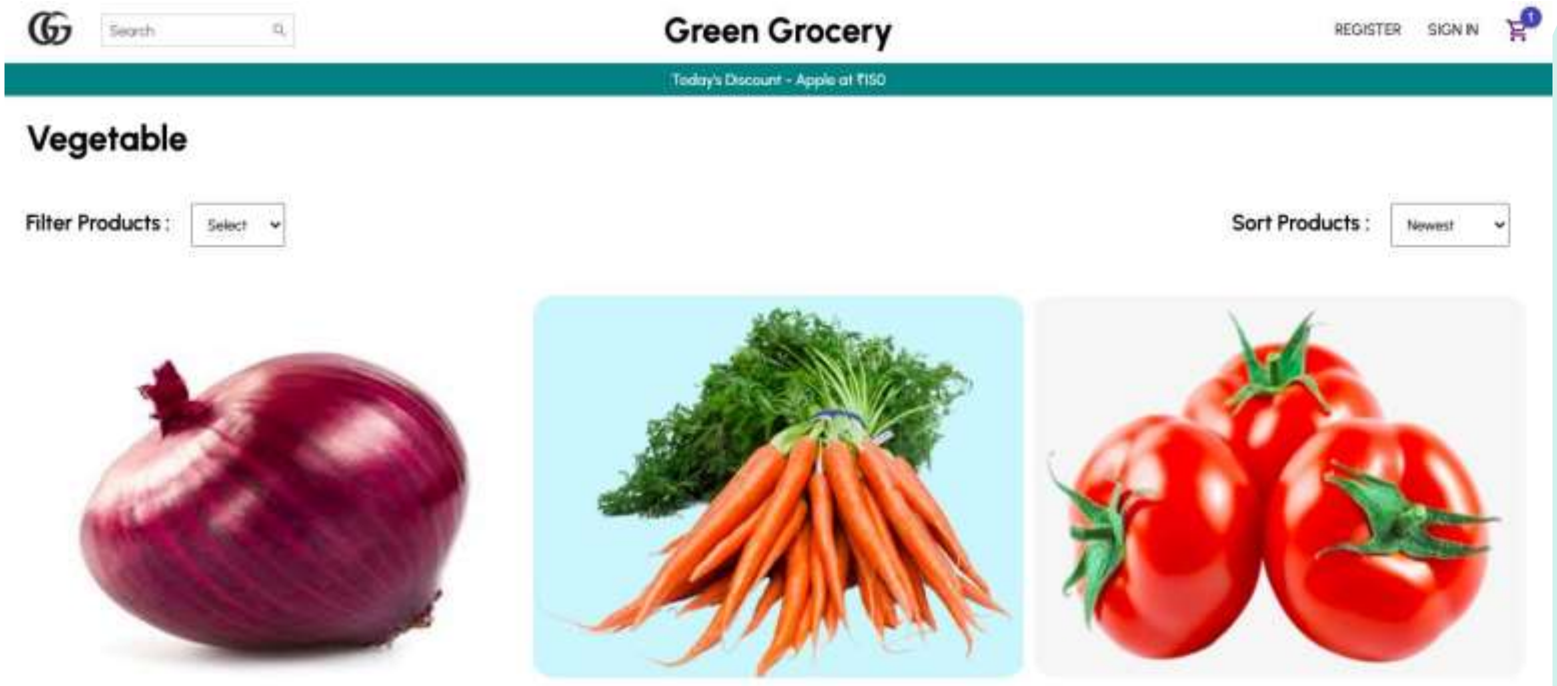
Green Grocery

Autumn Fruits Discount

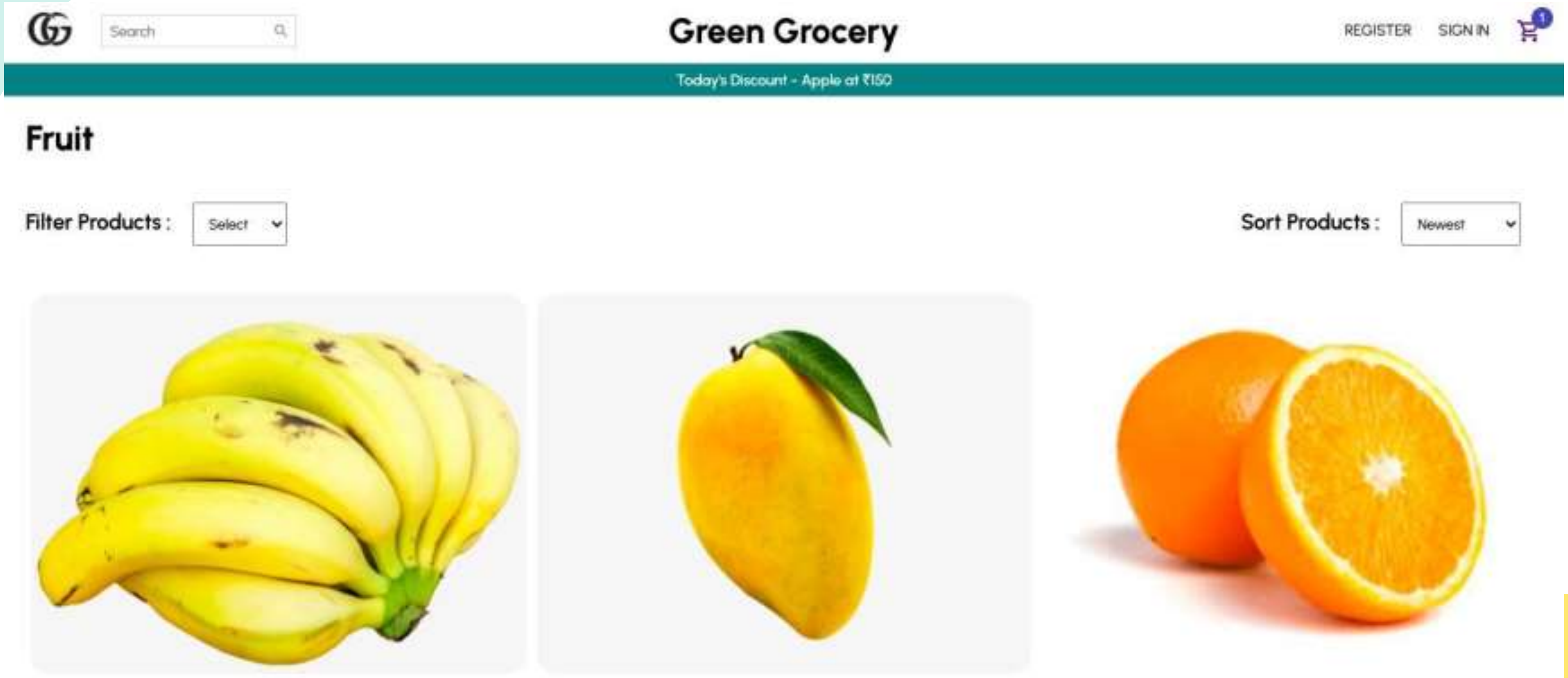
DON'T COMPROMISE ON Fresh Fruits!
GET FLAT 30% OFF for above ₹1000.

Views - Involves calls to backend to fetch data and display it to the user

Vegetables View Page



Fruits View Page



Green Grocery

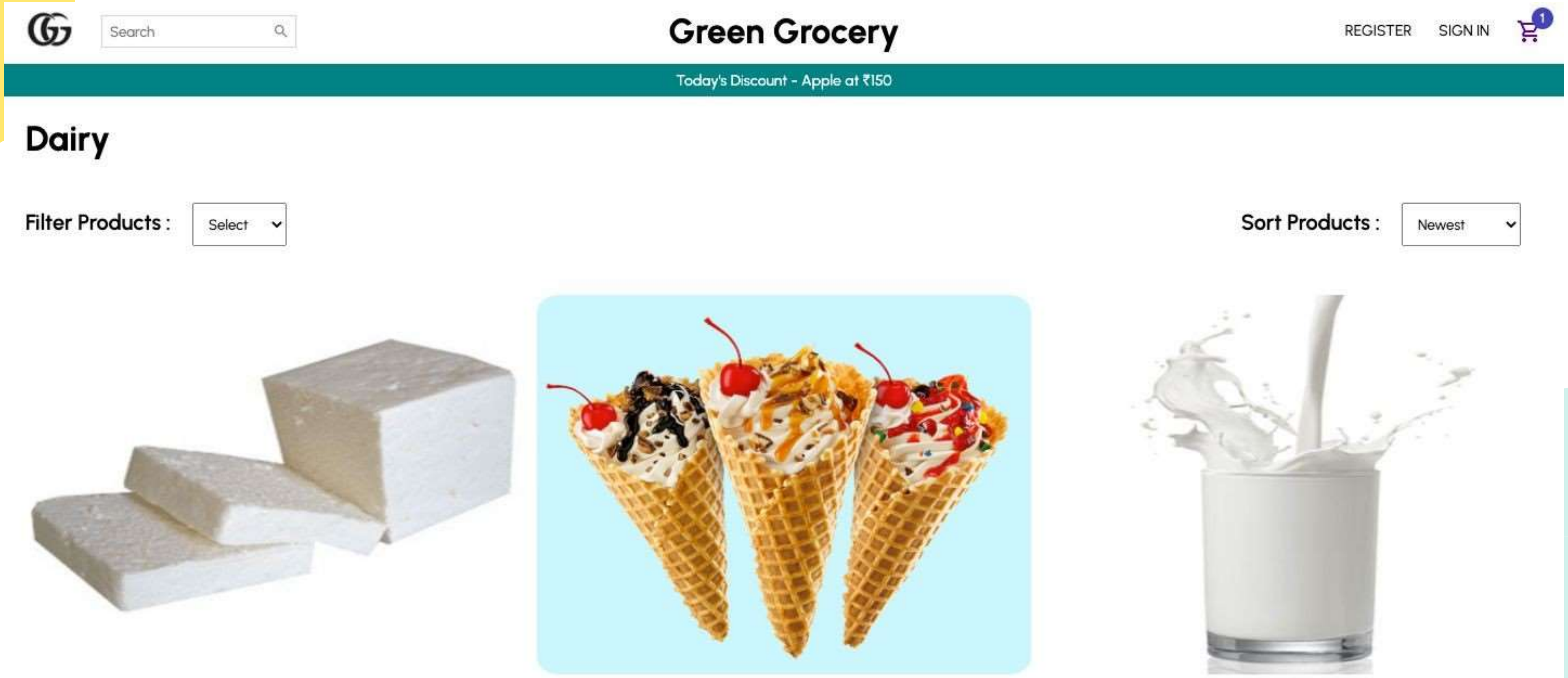
Today's Discount - Apple at ₹150

Fruit

Filter Products : Select

Sort Products : Newest

Dairy Product View Page



Green Grocery

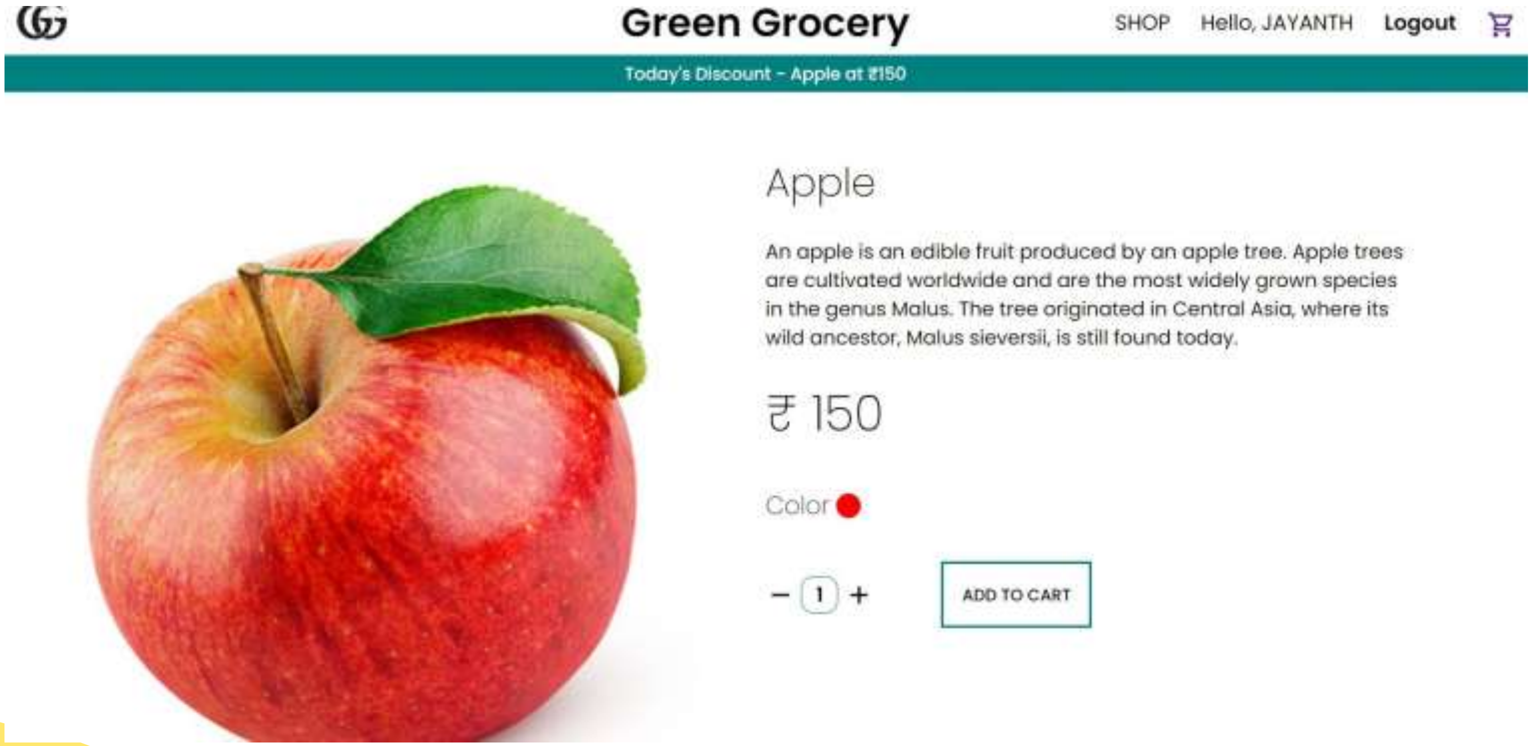
Today's Discount - Apple at ₹150

Dairy

Filter Products : Select

Sort Products : Newest

Detailed Product View Page



Green Grocery

SHOP Hello, JAYANTH Logout

Today's Discount - Apple at ₹150

Apple

An apple is an edible fruit produced by an apple tree. Apple trees are cultivated worldwide and are the most widely grown species in the genus Malus. The tree originated in Central Asia, where its wild ancestor, Malus sieversii, is still found today.

₹ 150

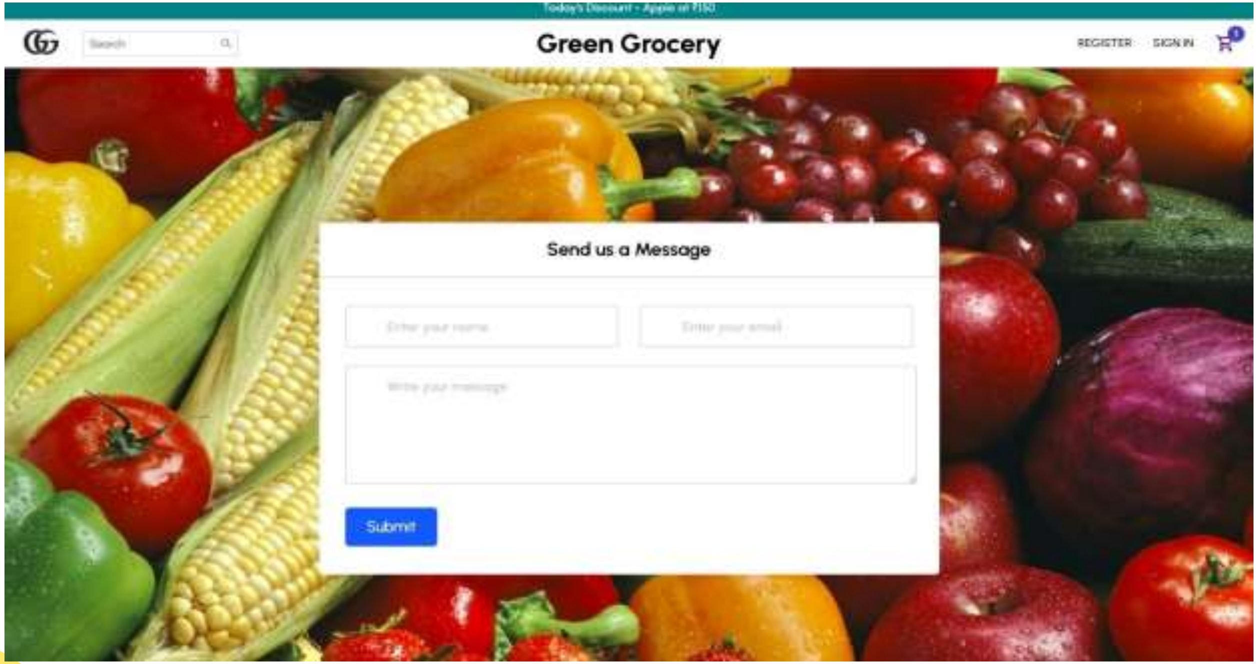
Color ●

− 1 +

ADD TO CART

Form -Involves presenting a form to collect user-input, validate, process and post to the backend server

Contact Form Page



Today's Discount - Apple at ₹150

Green Grocery

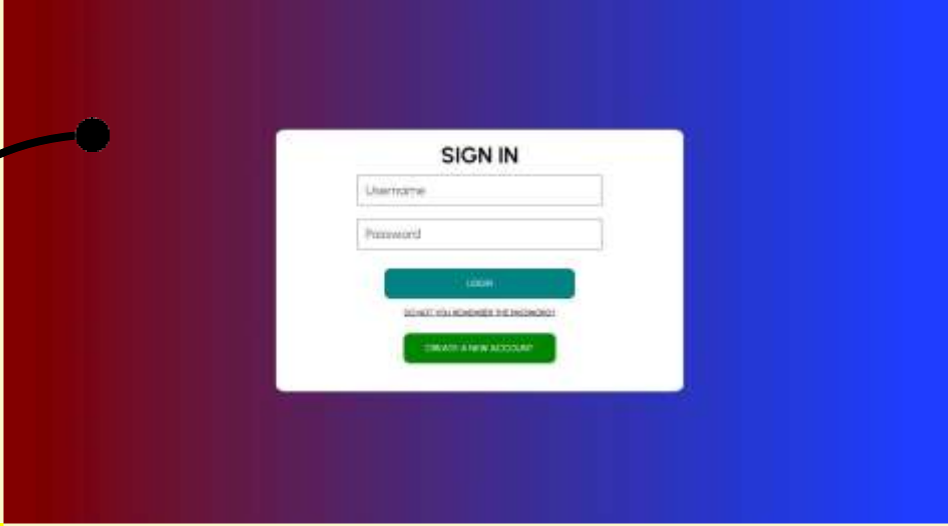
Send us a Message

Enter your name

Enter your email

Write your message

Log In


SIGN IN

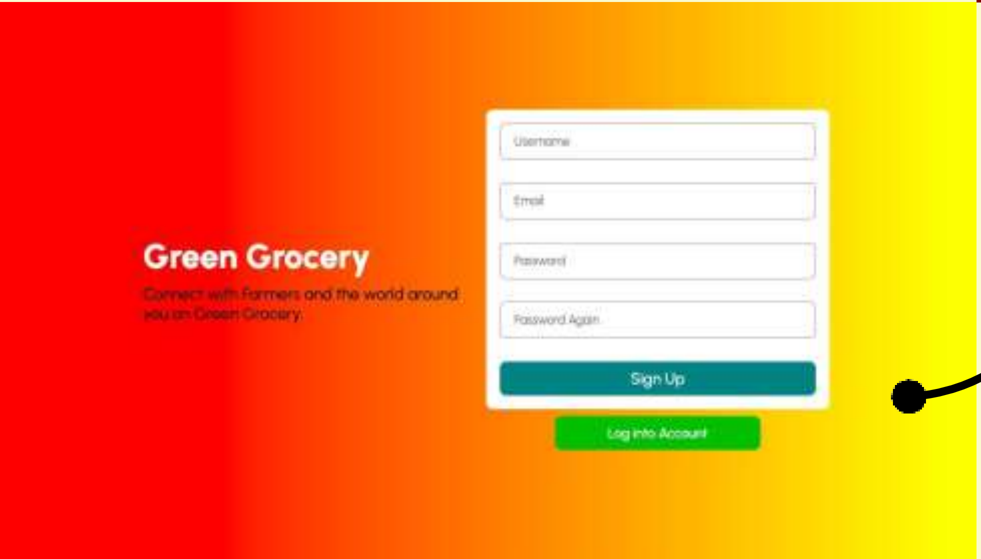
Username

Password

LOGIN

DO NOT YOU REMEMBER THE PASSWORD?

CREATE A NEW ACCOUNT


Green Grocery

Connect with farmers and the world around you on Green Grocery.

Username

Email

Password

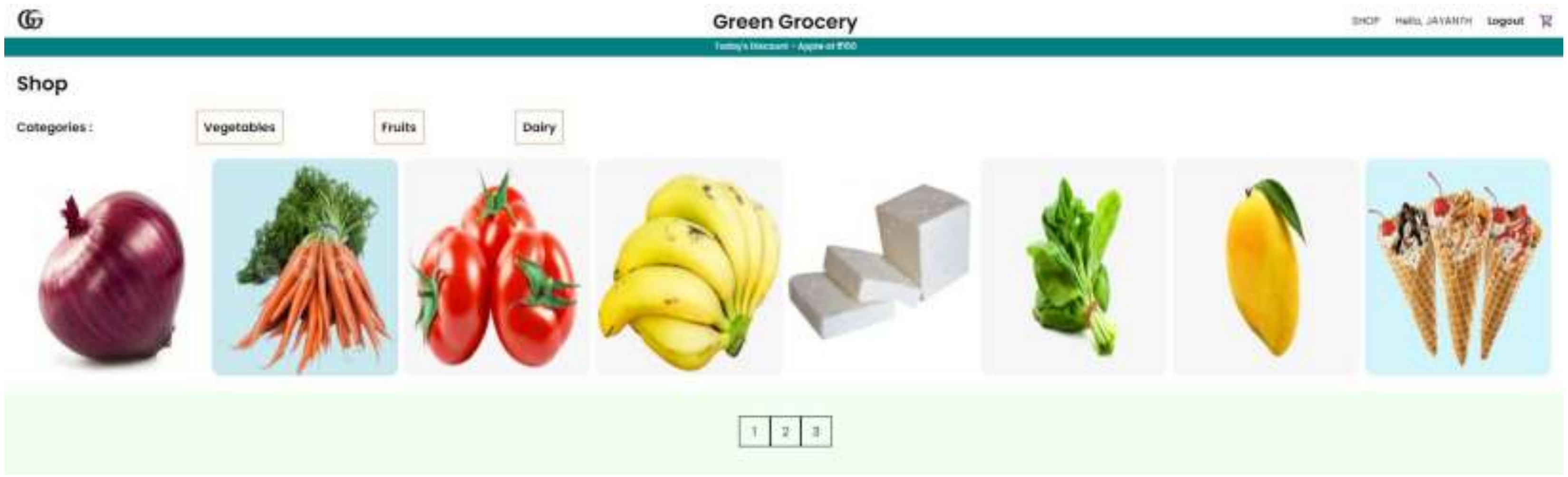
Password Again

Sign Up

Log into Account

Sign Up

Shop page with pagination



Green Grocery

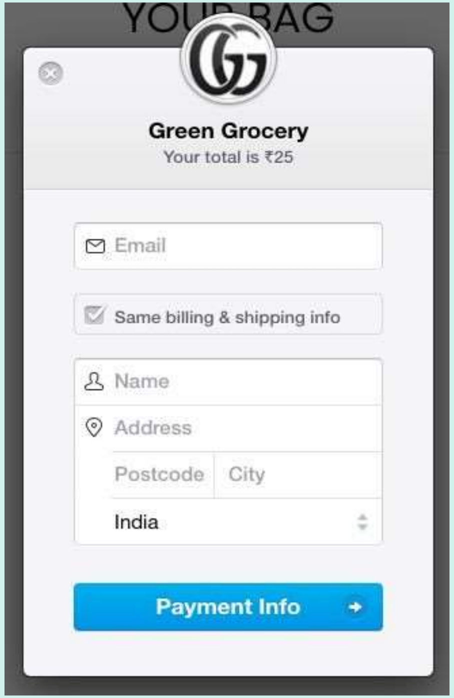
SHOP Hello, JAYANTH Logout

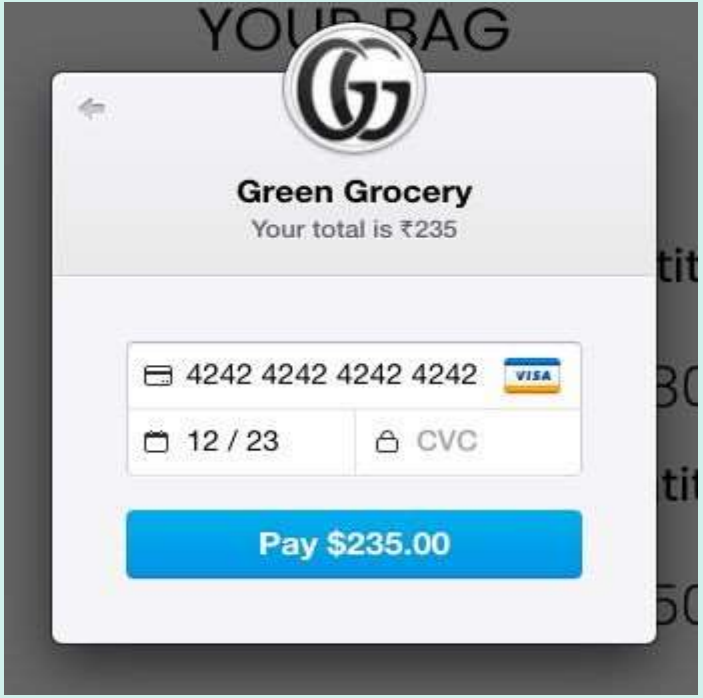
Today's Discount - Apple at ₹100

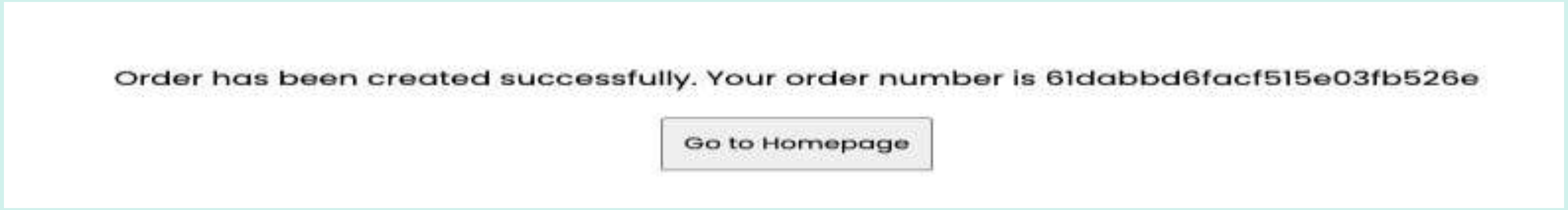
Shop

Categories :

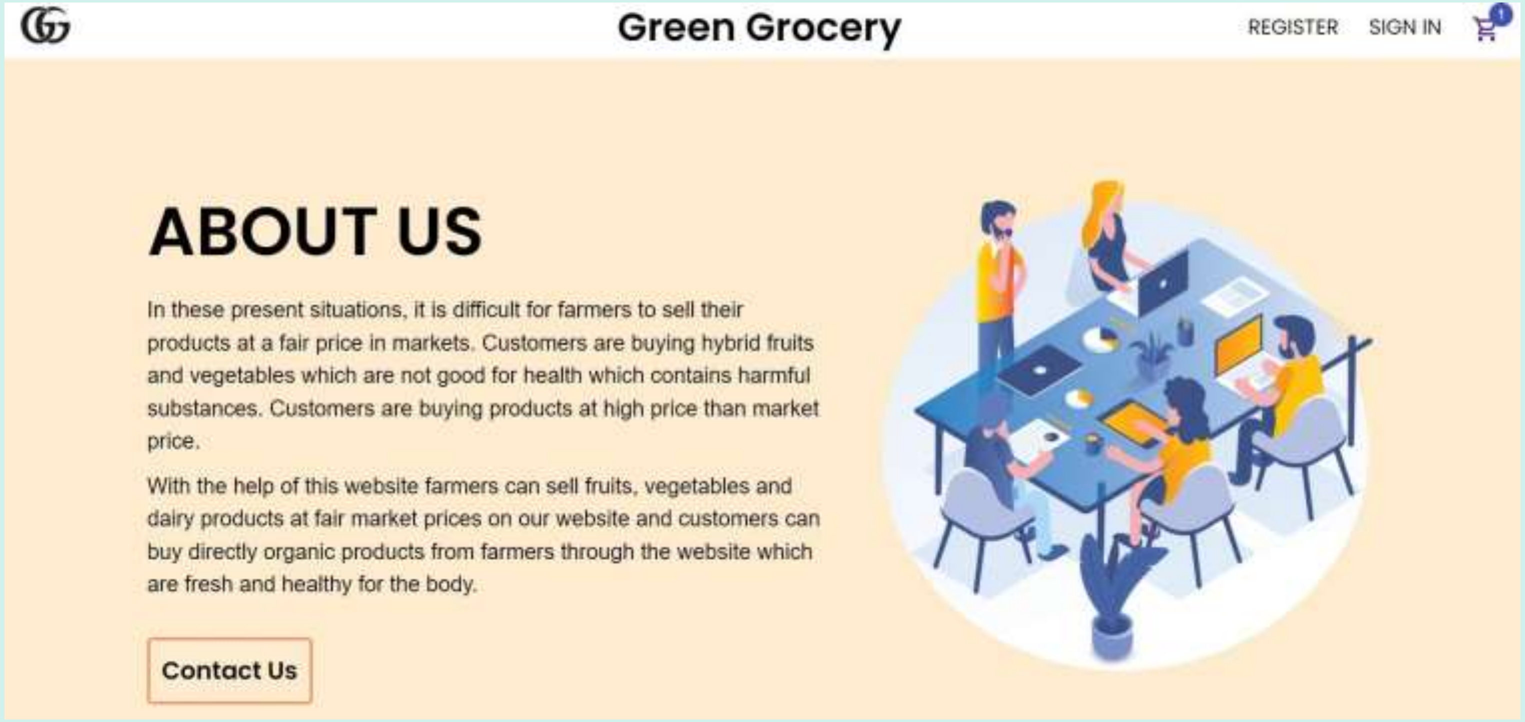
Payment and success






Order has been created successfully. Your order number is 61dabbd6facf515e03fb526e

About us

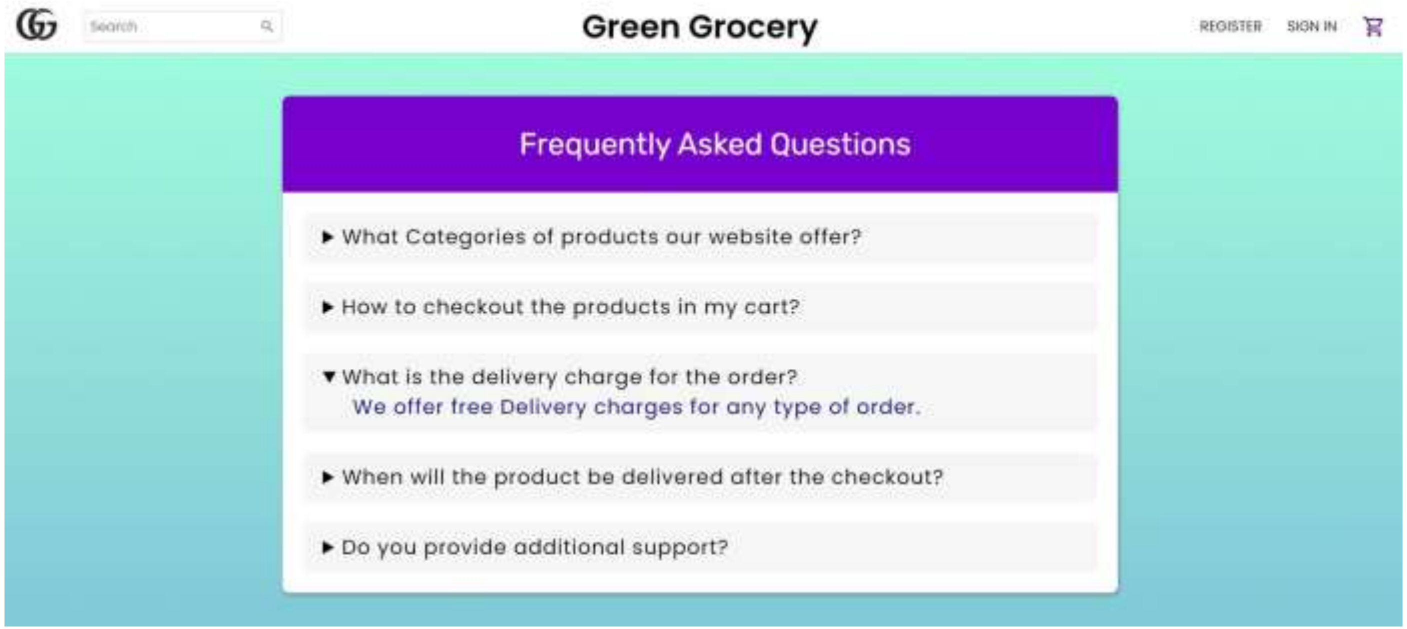


ABOUT US

In these present situations, it is difficult for farmers to sell their products at a fair price in markets. Customers are buying hybrid fruits and vegetables which are not good for health which contains harmful substances. Customers are buying products at high price than market price.

With the help of this website farmers can sell fruits, vegetables and dairy products at fair market prices on our website and customers can buy directly organic products from farmers through the website which are fresh and healthy for the body.

Frequently Asked Questions (FAQ)



Frequently Asked Questions

▸ What Categories of products our website offer?

▸ How to checkout the products in my cart?

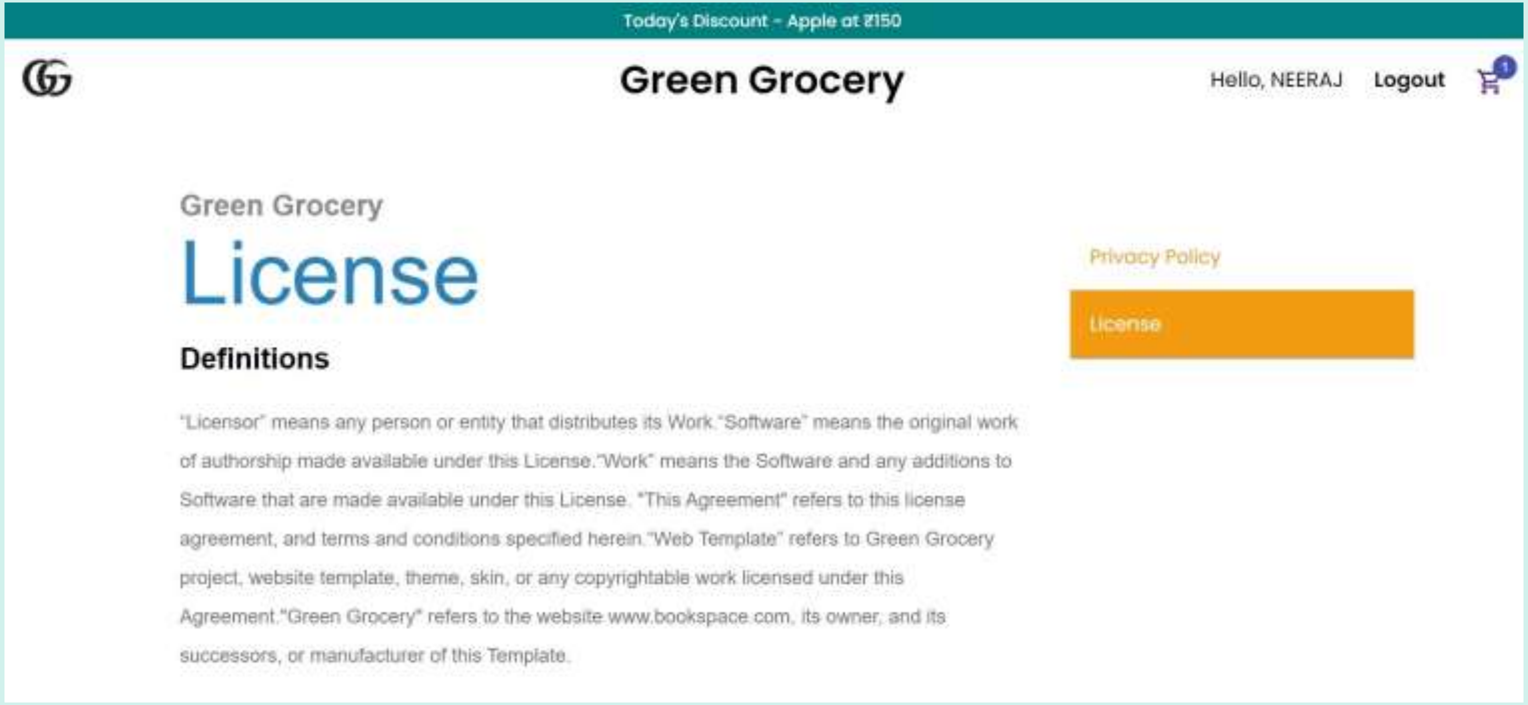
▾ What is the delivery charge for the order?

We offer free Delivery charges for any type of order.

▸ When will the product be delivered after the checkout?

▸ Do you provide additional support?

License



Today's Discount - Apple at ₹150

Green Grocery

Hello, NEERAJ Logout

Green Grocery

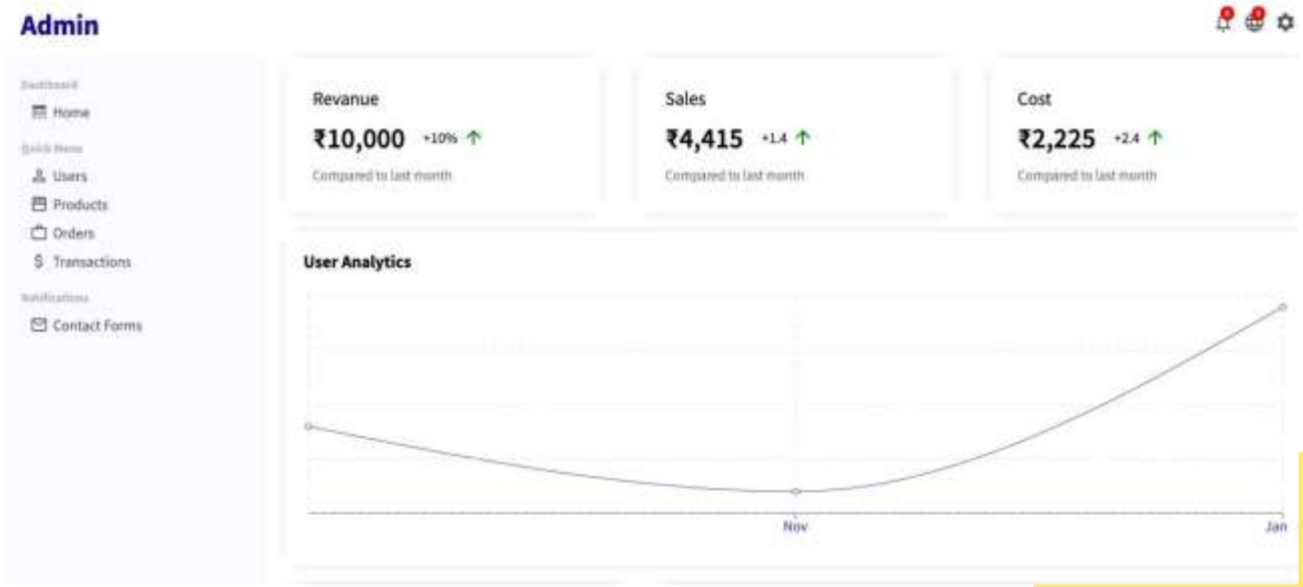
License

Definitions

"Licensor" means any person or entity that distributes its Work."Software" means the original work of authorship made available under this License."Work" means the Software and any additions to Software that are made available under this License. "This Agreement" refers to this license agreement, and terms and conditions specified herein."Web Template" refers to Green Grocery project, website template, theme, skin, or any copyrightable work licensed under this Agreement."Green Grocery" refers to the website www.bookspace.com, its owner, and its successors, or manufacturer of this Template.

Admin

Admin can monitor all the user activities and have full access to manage the site including adding, deleting and editing all pages and modules

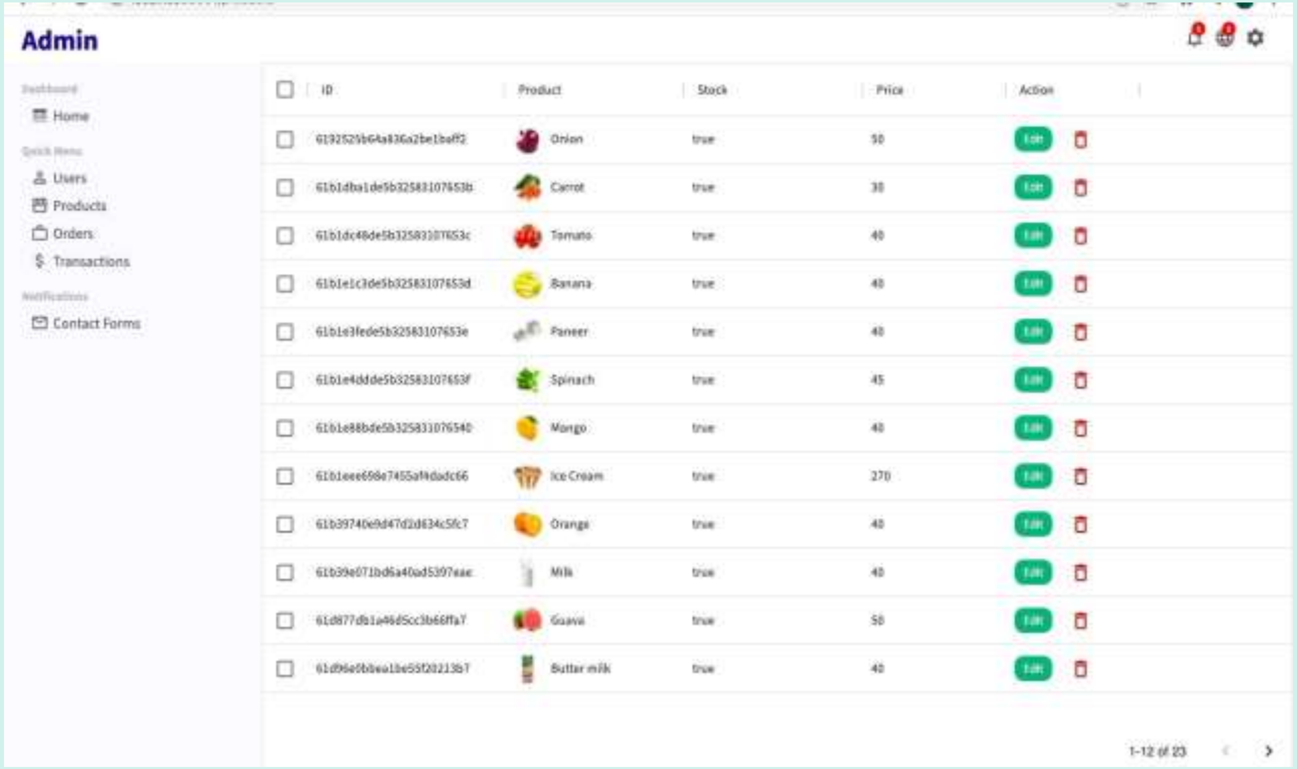


Revanue
₹10,000 +10% ↑
Compared to last month

Sales
₹4,415 +1.4 ↑
Compared to last month

Cost
₹2,225 +2.4 ↑
Compared to last month

Product page



Admin

Dashboard
- Home

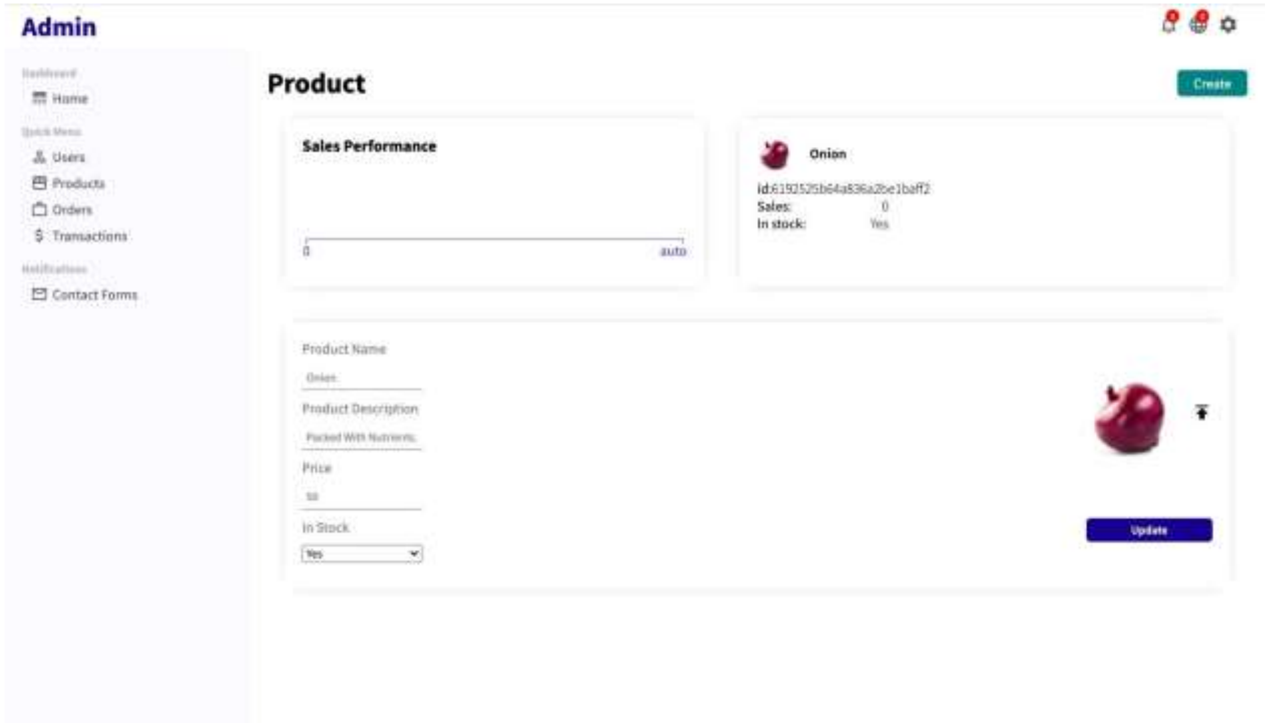
Quick Menu
- Users
- Products
- Orders
- Transactions

Notifications
- Contact Forms

ID	Product	Stock	Price	Action
6192525b64a836a2be1baff2	Onion	true	50	Edit
61b1dba1de5b32583107653b	Carrot	true	30	Edit
61b1dc48de5b32583107653c	Tomato	true	40	Edit
61b1e1c3de5b32583107653d	Banana	true	40	Edit
61b1e3fede5b32583107653e	Paneer	true	40	Edit
61b1e4ddde5b32583107653f	Spinach	true	45	Edit
61b1e88bde5b325831076540	Mango	true	40	Edit
61b1eee698e7455af4dadc66	Ice Cream	true	270	Edit
61b39740e9d47d2d634c5fc7	Orange	true	40	Edit
61b39e071bd6a40ad5397eae	Milk	true	40	Edit
61d877db1a46d5cc3b66ffa7	Guava	true	50	Edit
61d96e0bbea1be55f20213b7	Butter milk	true	40	Edit

1-12 of 23

Each Product page



Admin

Dashboard

Home

Quick Menu

Users

Products

Orders

Transactions

Notifications

Contact Forms

Product

Create

Sales Performance

0 auto

Onion

id:6192525b64a836a2be1baff2

Sales:	0
In stock:	Yes

Product Name

Onion

Product Description

Packed With Nutrients.

Price

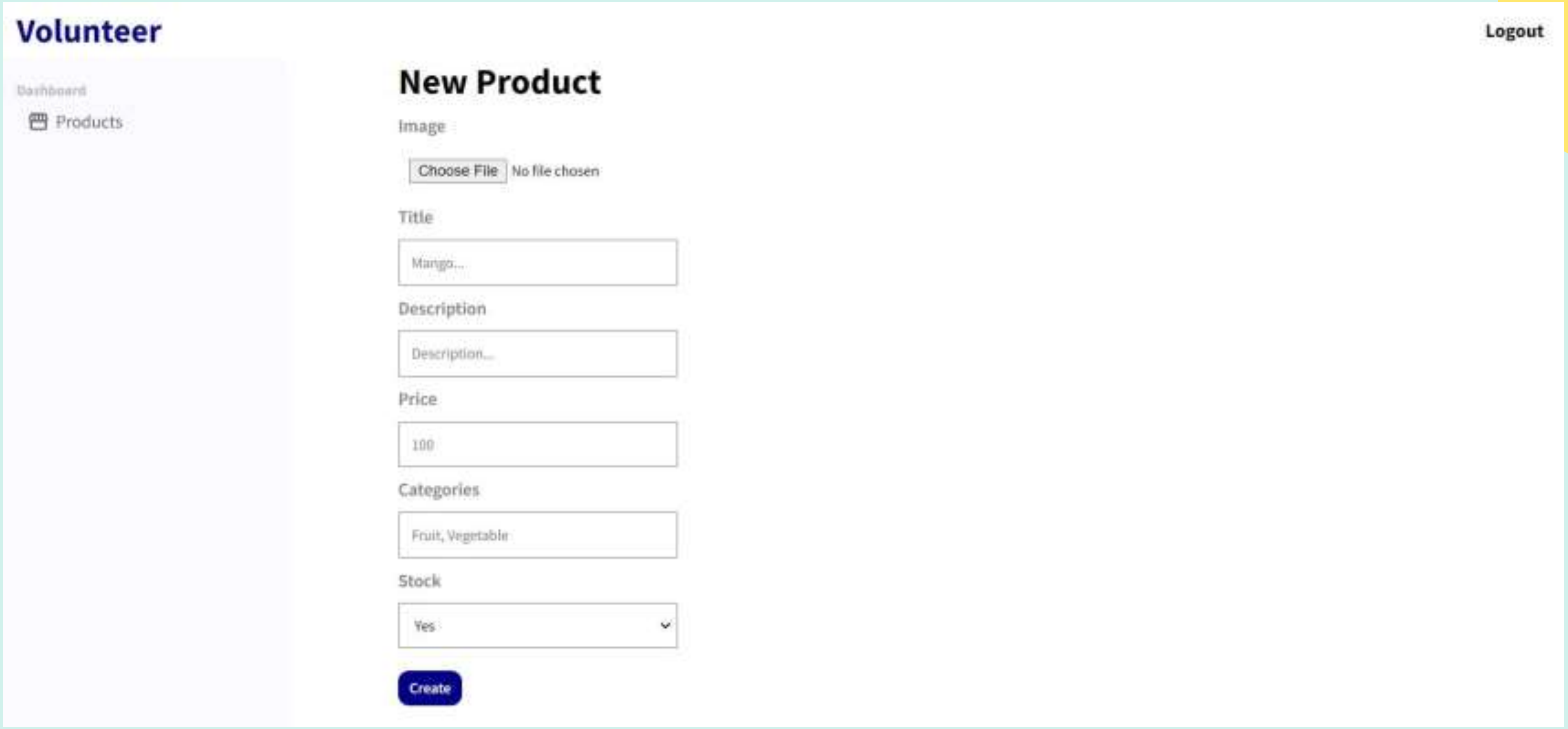
10

In Stock

Yes

Update

Volunteer



Volunteer

Logout

Dashboard

Products

New Product

Image

Choose File No file chosen

Title

Mango...

Description

Description...

Price

100

Categories

Fruit, Vegetable

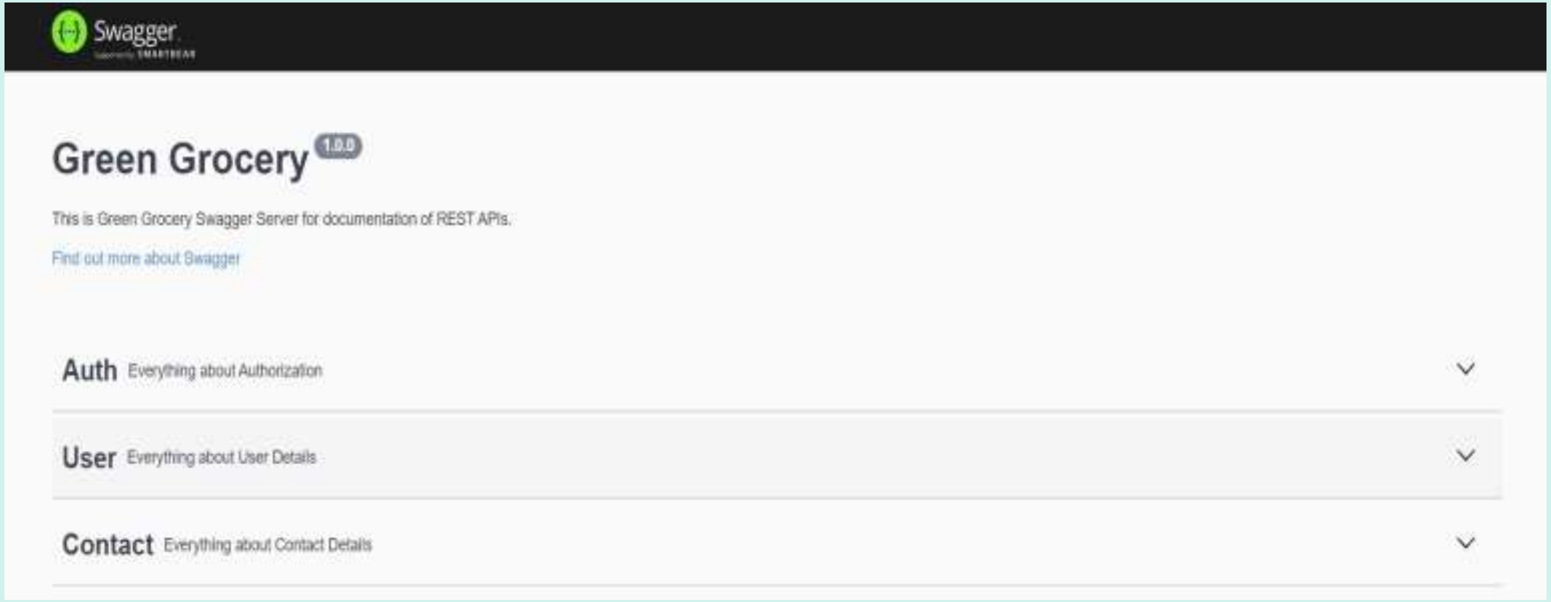
Stock

Yes

Create

Swagger U I

Swagger UI is a collection of HTML, JavaScript, and CSS assets that dynamically generate beautiful documentation from a Swagger-compliant API



Green Grocery 1.0.0

This is Green Grocery Swagger Server for documentation of REST APIs.

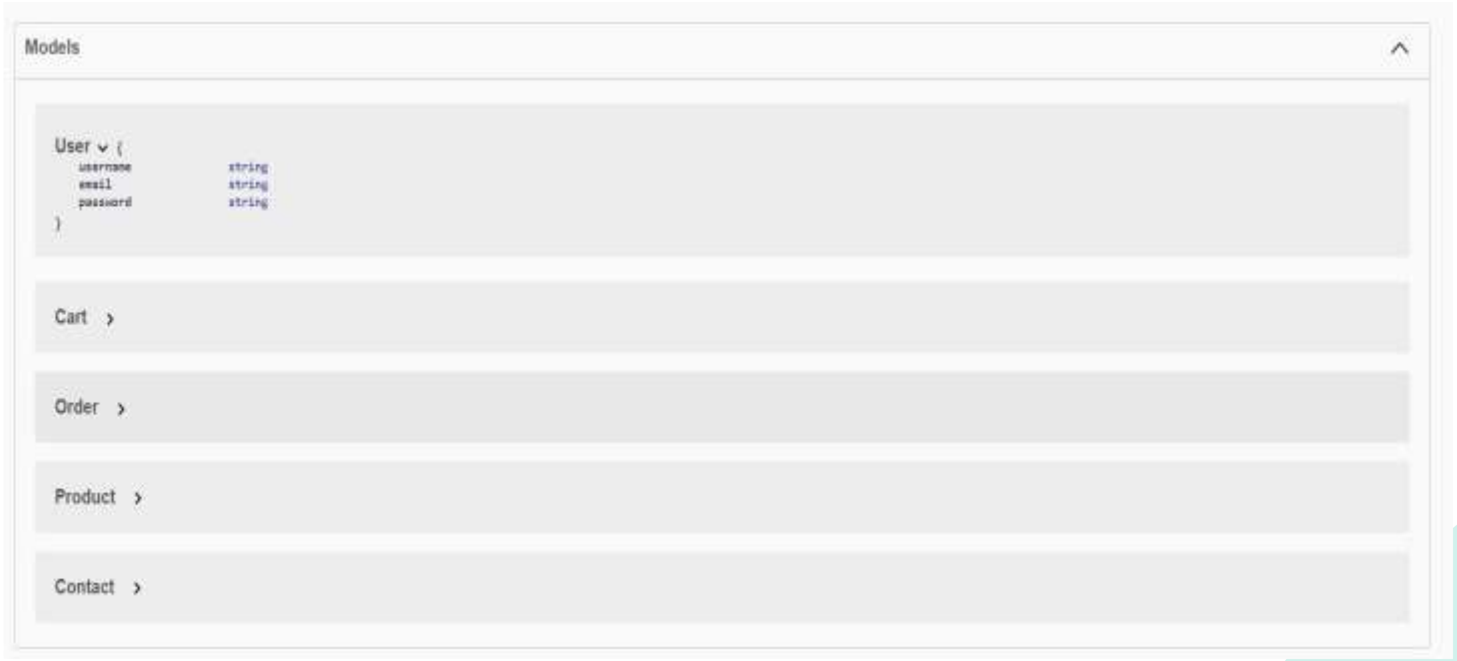
Find out more about Swagger

Auth Everything about Authorization

User Everything about User Details

Contact Everything about Contact Details

Models that are defined in the Swagger UI are as shown. A Model Object holds the definition of a new model for this API Declaration.


Models

```
User ∨ {
    username          string
    email             string
    password          string
}
```

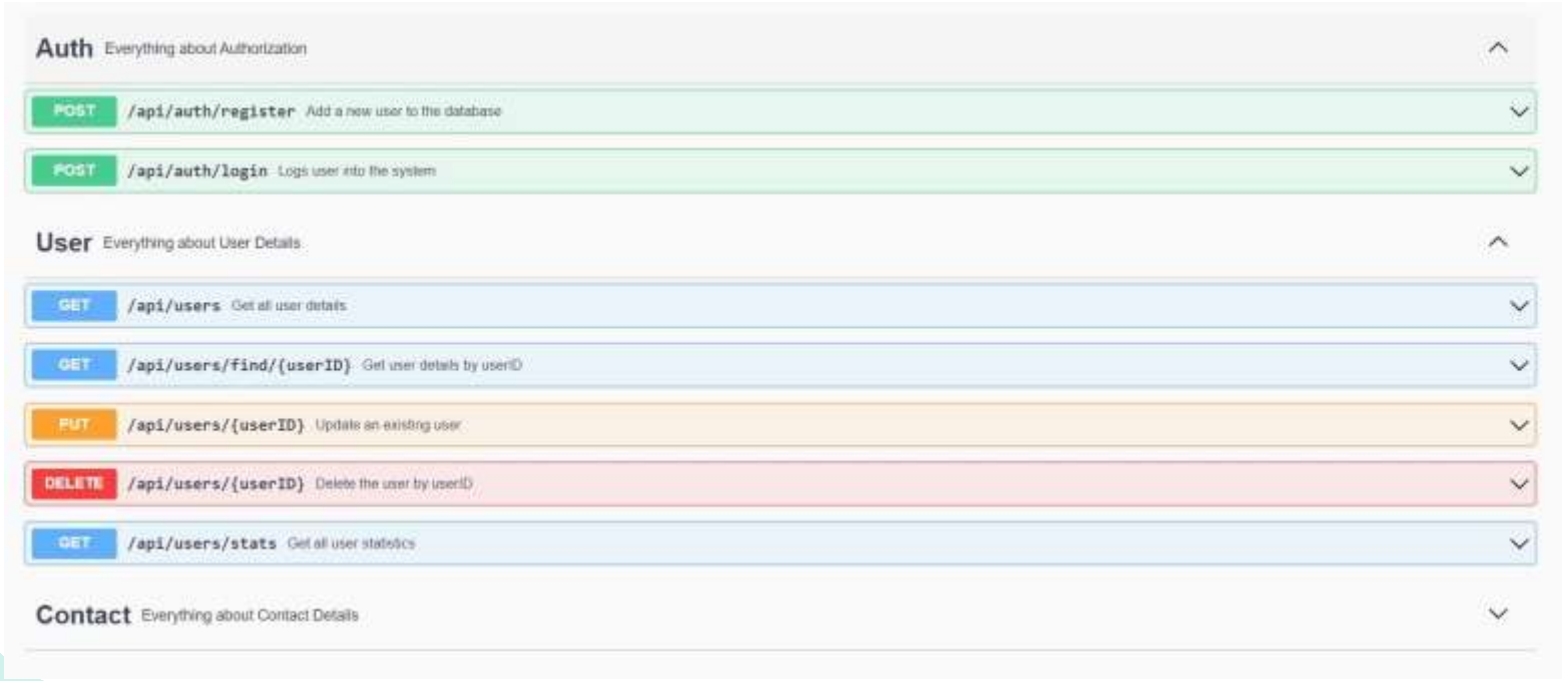
Cart ›

Order ›

Product ›

Contact ›

These are various routes defined for each schema model with individual endpoints (paths) in the API, and the HTTP methods (operations) supported by these endpoints.


Auth Everything about Authorization

POST /api/auth/register Add a new user to the database

POST /api/auth/login Logs user into the system

User Everything about User Details

GET /api/users Get all user details

GET /api/users/find/{userID} Get user details by userID

PUT /api/users/{userID} Update an existing user

DELETE /api/users/{userID} Delete the user by userID

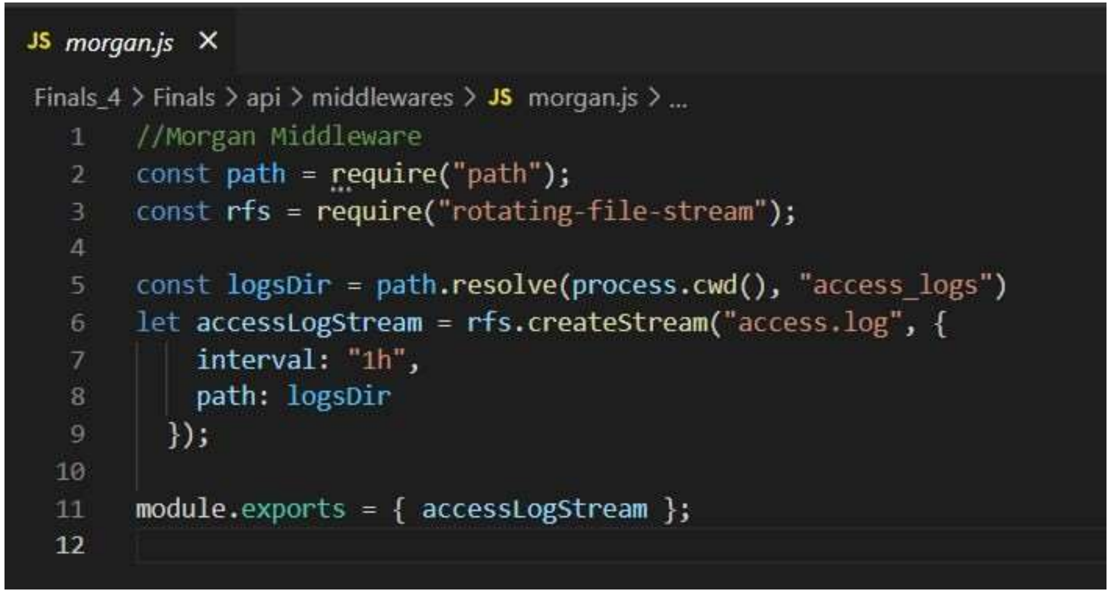
GET /api/users/stats Get all user statistics

Contact Everything about Contact Details

Express Middlewares

Morgan Middleware

Morgan is a Node. js and Express middleware to log HTTP requests and errors, and simplifies the process.

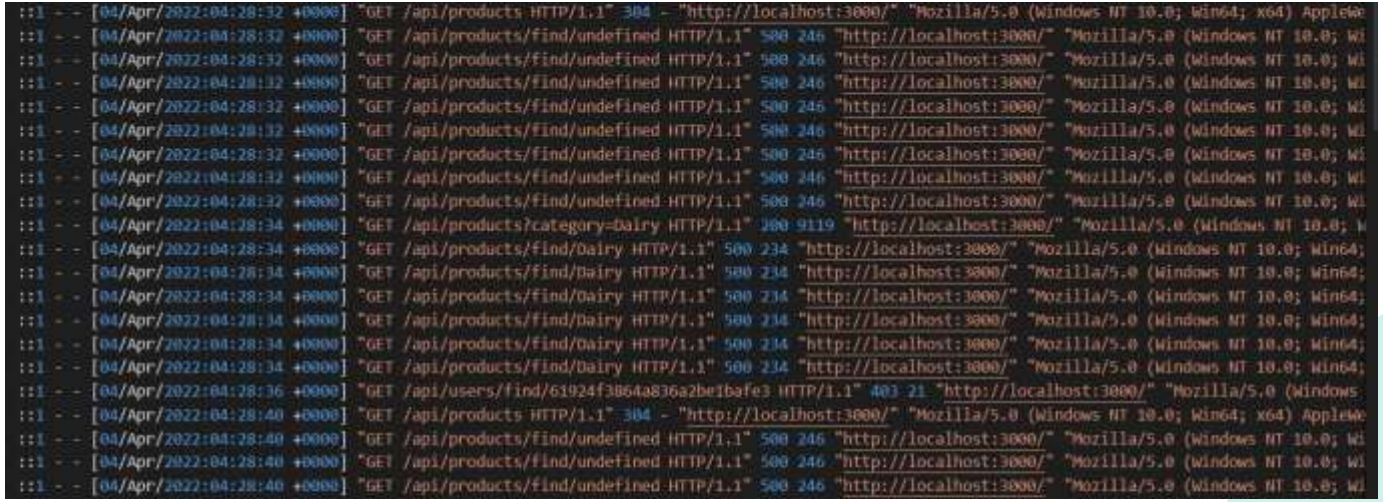


```javascript
//Morgan Middleware
const path = require("path");
const rfs = require("rotating-file-stream");

const logsDir = path.resolve(process.cwd(), "access_logs")
let accessLogStream = rfs.createStream("access.log", {
    interval: "1h",
    path: logsDir
  });

module.exports = { accessLogStream };
```

There are two ways to format your logs:

1) Pre-defined formats: Morgan already provides with a simple pre-configured set of items to log, we need to pick the combination that suits our needs.
2) Manually by using tokens: If the pre-defined formats aren't enough, we can easily create new ones.

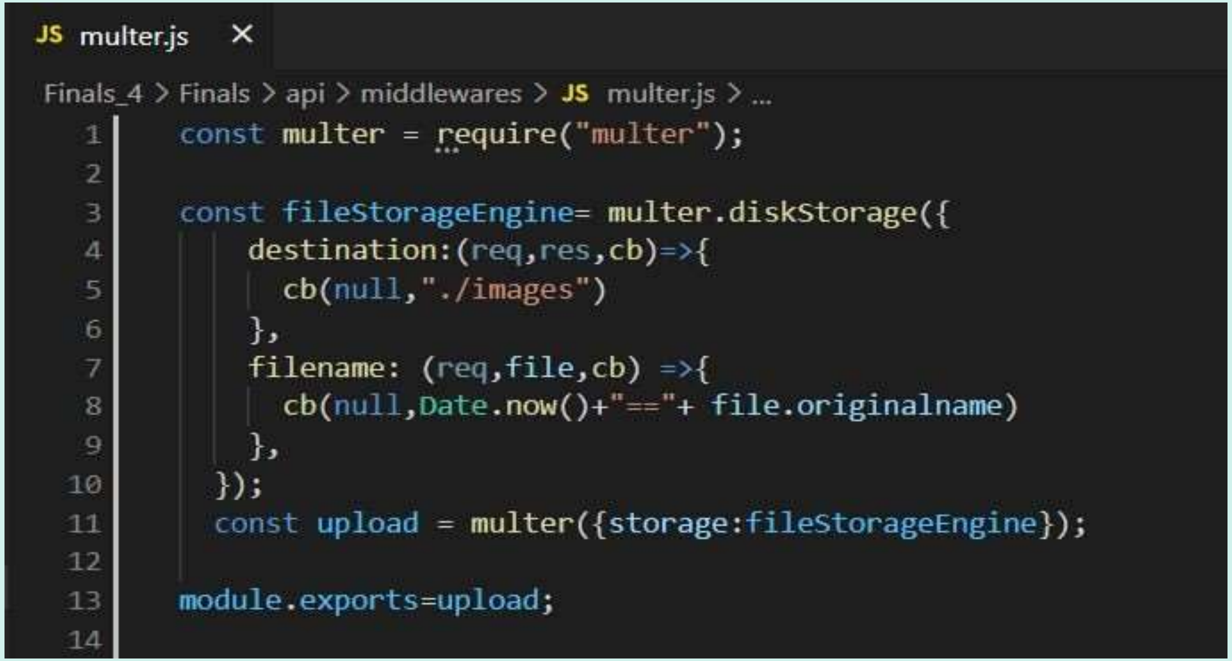
We have used "combined" format which gives the Apache standard combined format for the logs which can be as shown below.



```
::1 - - [04/Apr/2022:04:28:32 +0000] "GET /api/products HTTP/1.1" 304 - "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
::1 - - [04/Apr/2022:04:28:32 +0000] "GET /api/products/find/undefined HTTP/1.1" 500 246 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Wi
::1 - - [04/Apr/2022:04:28:32 +0000] "GET /api/products/find/undefined HTTP/1.1" 500 246 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Wi
::1 - - [04/Apr/2022:04:28:32 +0000] "GET /api/products/find/undefined HTTP/1.1" 500 246 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Wi
::1 - - [04/Apr/2022:04:28:32 +0000] "GET /api/products/find/undefined HTTP/1.1" 500 246 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Wi
::1 - - [04/Apr/2022:04:28:32 +0000] "GET /api/products/find/undefined HTTP/1.1" 500 246 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Wi
::1 - - [04/Apr/2022:04:28:32 +0000] "GET /api/products/find/undefined HTTP/1.1" 500 246 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Wi
::1 - - [04/Apr/2022:04:28:32 +0000] "GET /api/products/find/undefined HTTP/1.1" 500 246 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Wi
::1 - - [04/Apr/2022:04:28:32 +0000] "GET /api/products/find/undefined HTTP/1.1" 500 246 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Wi
::1 - - [04/Apr/2022:04:28:34 +0000] "GET /api/products?category=Dairy HTTP/1.1" 200 9119 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; W
::1 - - [04/Apr/2022:04:28:34 +0000] "GET /api/products/find/Dairy HTTP/1.1" 500 234 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Win64;
::1 - - [04/Apr/2022:04:28:34 +0000] "GET /api/products/find/Dairy HTTP/1.1" 500 234 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Win64;
::1 - - [04/Apr/2022:04:28:34 +0000] "GET /api/products/find/Dairy HTTP/1.1" 500 234 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Win64;
::1 - - [04/Apr/2022:04:28:34 +0000] "GET /api/products/find/Dairy HTTP/1.1" 500 234 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Win64;
::1 - - [04/Apr/2022:04:28:34 +0000] "GET /api/products/find/Dairy HTTP/1.1" 500 234 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Win64;
::1 - - [04/Apr/2022:04:28:34 +0000] "GET /api/products/find/Dairy HTTP/1.1" 500 234 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Win64;
::1 - - [04/Apr/2022:04:28:36 +0000] "GET /api/users/find/61924f3864a836a2be1bafe3 HTTP/1.1" 403 21 "http://localhost:3000/" "Mozilla/5.0 (Windows
::1 - - [04/Apr/2022:04:28:40 +0000] "GET /api/products HTTP/1.1" 304 - "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
::1 - - [04/Apr/2022:04:28:40 +0000] "GET /api/products/find/undefined HTTP/1.1" 500 246 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Wi
::1 - - [04/Apr/2022:04:28:40 +0000] "GET /api/products/find/undefined HTTP/1.1" 500 246 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Wi
::1 - - [04/Apr/2022:04:28:40 +0000] "GET /api/products/find/undefined HTTP/1.1" 500 246 "http://localhost:3000/" "Mozilla/5.0 (Windows NT 10.0; Wi
```

Multer Middleware

Multer is a node. js middleware for handling multipart/form-data , which is primarily used for uploading files.

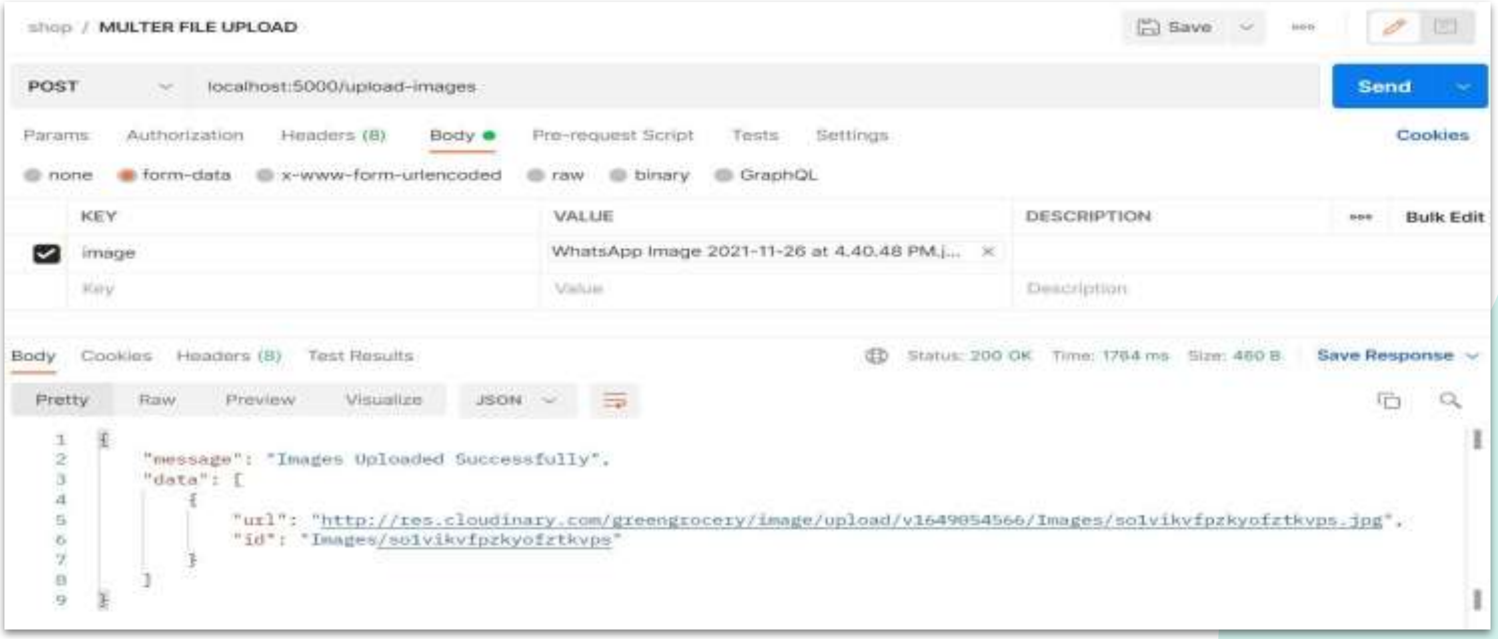


multer.js ×

Finals_4 > Finals > api > middlewares > multer.js > ...

```
const multer = require("multer");

const fileStorageEngine= multer.diskStorage({
    destination:(req,res,cb)=>{
      cb(null,"./images")
    },
    filename: (req,file,cb) =>{
      cb(null,Date.now()+"=="+ file.originalname)
    },
  });
  const upload = multer({storage:fileStorageEngine});

module.exports=upload;

```

- We used cloudinary API service to upload the images to the cloud.
- We created a cloudinary.js file that contains the configurations to upload files to Cloudinary service.
- Now, in the postman on the body section, select form-data and enter the field name used in the multer configuration file which in our case is image. Now, postman would like this when sending.



POST localhost:5000/upload-images

	KEY	VALUE	DESCRIPTION
✓	image	WhatsApp Image 2021-11-26 at 4.40.48 PM.j...	
	Key	Value	Description

Status: 200 OK Time: 1784 ms Size: 460 B

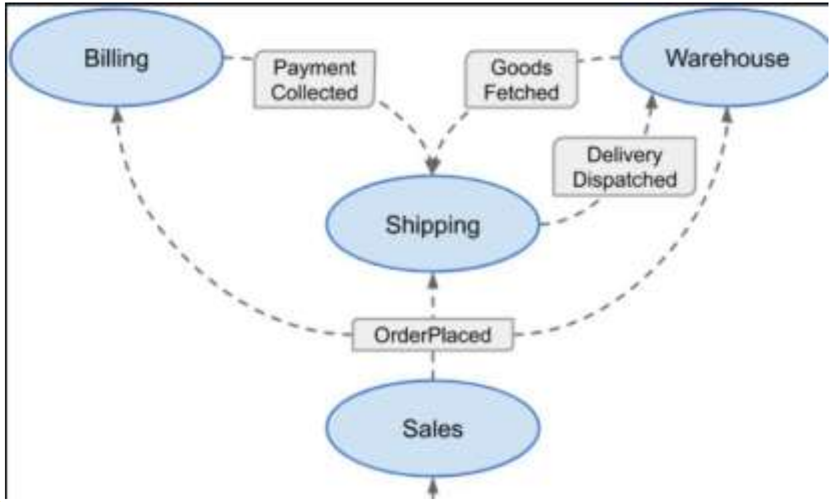
```
{
    "message": "Images Uploaded Successfully",
    "data": [
        {
            "url": "http://res.cloudinary.com/greengrocery/image/upload/v1649054566/Images/so1vikvfpzkyofztkvps.jpg",
            "id": "Images/so1vikvfpzkyofztkvps"
        }
    ]
}
```

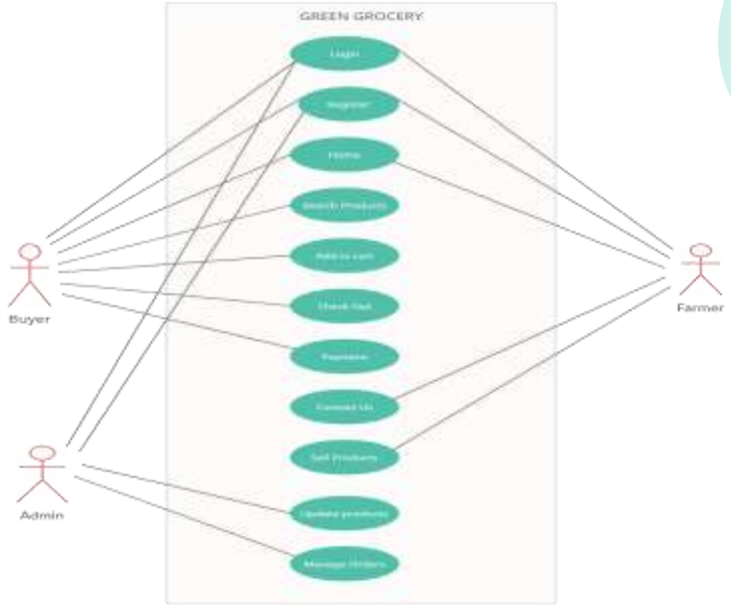
Domain Driven Designs

- ❖ Domain-Driven Design is an approach to software design that our focus should not be primarily on technology, rather it should be primarily on business.
- ❖ It is a software development technique that centers the development on programming a domain model that has a rich understanding of the processes and rules of a domain.
- ❖ Here, we design and create highly expressive models. It highly aims to create models that are understandable by everyone involved in the software development.
- ❖ A model of a domain can be depicted as a UML sketch, as code, and in the language of the domain.

As we are building an e-Commerce System, our Domain is the e-Commerce business.

Domain-driven design talks about two kinds of design tools, first one is Strategic design tools and another one is Tactical design tools.





Use Case Diagram of the Green Grocery System.

1) **Strategic Design**

The common terms used in Strategic Design of DDD are:

- Model – It acts as a core logic and describes aspects of domain.
- Ubiquitous Language – A common language using common verbs and nouns for classes, methods, services, and objects while talking with domain experts and team members.
- Bounded Context –It is a description of a boundary and acts as a particular domain model.

2) **Tactical Design**

- Tactical design talks about implementation details i.e., modeling domain. It generally takes care of components inside a bounded context.
- Entity, Value Objects, Services, Aggregates, Factories and Repositories tools are high-level concepts that can be used to create and modify domain models.

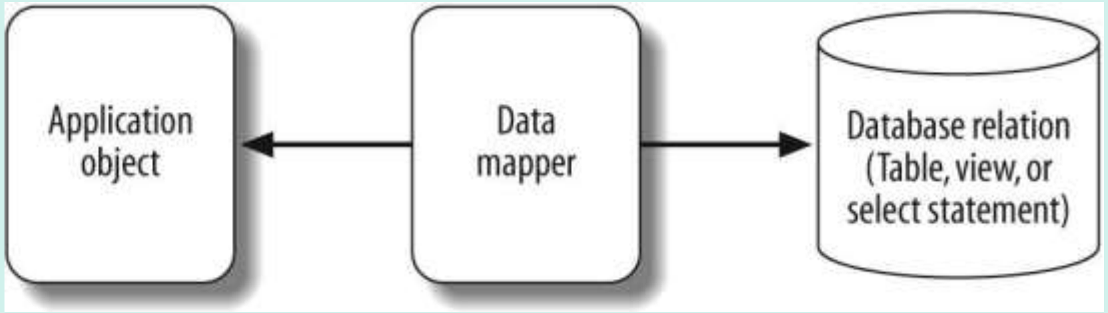
ORM VS ODM

- ❑ ORM or Object-Relational Mapper maps objects to records in relational database tables. Oracle, MySQL, and PostgreSQL are well-known relational databases examples.
- ❑ ODM or Object-Document Mapper does the same for non-relational, document-based databases like MongoDB -it maps objects to documents.
- ❑ The main difference between ORM and ODM is that ORM is for MySQL databases, while ODM does the mapping for document representation of data
- ❑ In our project, we are using ODM i.e. MongoDB.

Object Document Mapping (O D M)

- It is an object associated with NoSQL database. As the name suggests, it maps to the document related module.

- MongoDB expresses data to be saved in a JSON-like format and saves it as a document. The function of associating such a document with an object in a programming language is called Object-Document Mapping (O/D mapping).



Future Work :

redis

Using Redis to Optimize MongoDB Queries

- ❑ Redis 10 - 30% faster when the dataset fits within working memory of a single machine than mongoDB database.
- ❑ We use Redis package from npm in our application for the purpose of fast querying.
- ❑ Firstly, we will add a cache configuration and connect our application to the Redis server.
- ❑ Secondly, we need to add the logic that queries Redis, and if there is no answer it will query our MongoDB.
- ❑ Lastly, After we fetch new data from MongoDB, we'll store it in Redis cache database for future queries.