



Green Grocery

An Online Grocery Store



Group No: 30



Group Members :

Jayanth Sri Ram Kanduri	- S20190010084
Neeraj Dusa	- S20190010047
Harish Mullagura	- S20190010124
Anirudh Jakhotia	- S20190010007
Rakesh Ganeshula	- S20190010052



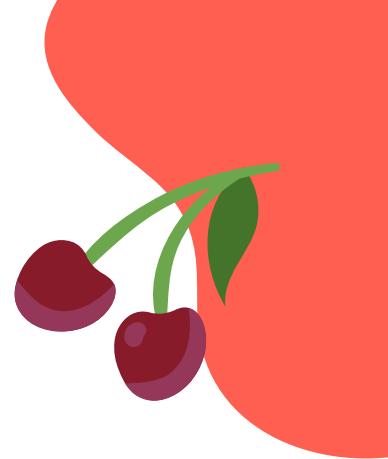


01

Project Business Cases:

Problem Motivation

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.

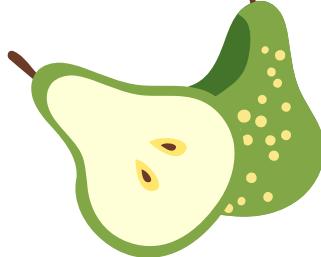
Problem Solution

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 :



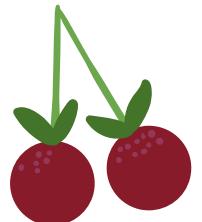
Farmers can contact GreenGrocery website and can sell their vegetables, fruits and dairy products at fair prices. The process includes, firstly, the delivery guy pick their products and the management staff are responsible for updating these products on the website. Users can login into the website and 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.

The form is titled "Login Form" at the top center. It features two buttons: "Login" (highlighted in purple) and "Signup" (in white). Below these are two input fields: "Email Address" and "Password". There is also a "Forgot password?" link. At the bottom is a large purple "Login" button. A small note at the very bottom right says "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.

Filter Products :

Select

Color

Select

White

Orange

Red

Yellow

Green

Sort Products:

Price (asc)

Newest

Price (asc)

Price (desc)





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

```
JS Announcement.jsx X
Finals > client > src > components > JS Announcement.jsx > ...
1 import styled from "styled-components";
2
3 const Container = styled.div`
4   height: 30px;
5   background-color: teal;
6   color: white;
7   display: flex;
8   align-items: center;
9   justify-content: center;
10  font-size: 14px;
11  font-weight: 500;
12`;
13
14 const Announcement = () => {
15   return <Container>Today's Discount - Apple at ₹150</Container>;
16 };
17
18 export default Announcement;
19
```



- 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.

The screenshot shows a code editor window with the file 'Slider.jsx' open. The file contains JSX code for a slider component. The code uses a map operation on 'sliderItems' to generate multiple 'Slide' components for each item in the array. Each slide contains an image and some descriptive text. A button labeled 'SHOP NOW' is also present. The code editor has syntax highlighting and shows line numbers from 102 to 130.

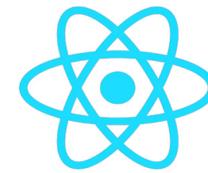
```
JS Slider.jsx ×
Finals > client > src > components > JS Slider.jsx > [e] Arrow
102
103 return (
104   <Container>
105     <Arrow direction="left" onClick={() => handleClick("left")}>
106       <ArrowLeftOutlined />
107     </Arrow>
108     <Wrapper slideIndex={slideIndex}>
109       {sliderItems.map((item) => (
110         <Slide bg={item.bg} key={item.id}>
111           <ImgContainer>
112             <Image src={item.img} />
113           </ImgContainer>
114           <InfoContainer>
115             <Title>{item.title}</Title>
116             <Desc>{item.desc}</Desc>
117             <Button><Link to={item.link}>SHOP NOW</Link></Button>
118           </InfoContainer>
119         </Slide>
120       )));
121     </Wrapper>
122     <Arrow direction="right" onClick={() => handleClick("right")}>
123       <ArrowRightOutlined />
124     </Arrow>
125   </Container>
126 );
127
128
129 export default slider;
130
```



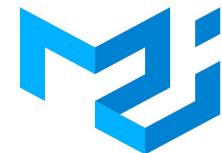
React-UI Elements used:

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.



React





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 :

```
JS Navbar.jsx X
Finals > client > src > components > JS Navbar.jsx > ...
1 import { Badge } from "@material-ui/core";
2 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 :

REGISTER SIGN IN 



Today's Discount - Apple at ₹150



Search



Green Grocery

REGISTER

SIGN IN



Autumn Fruits Discount

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

[SHOP NOW](#)



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

Vegetables View Page

Green Grocery

REGISTER SIGN IN

Today's Discount - Apple at ₹150

Vegetable

Filter Products : Select

Sort Products : Newest





Fruits View Page



Search



REGISTER

SIGN IN



Green Grocery

Today's Discount - Apple at ₹150

Fruit

Filter Products :

Select

Sort Products :

Newest





Dairy Product View Page



Search



Green Grocery

REGISTER

SIGN IN



Today's Discount - Apple at ₹150

Dairy

Filter Products :

Select ▾

Sort Products :

Newest ▾





Detailed Product View Page



Green Grocery

[SHOP](#)[Hello, JAYANTH](#)[Logout](#)

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



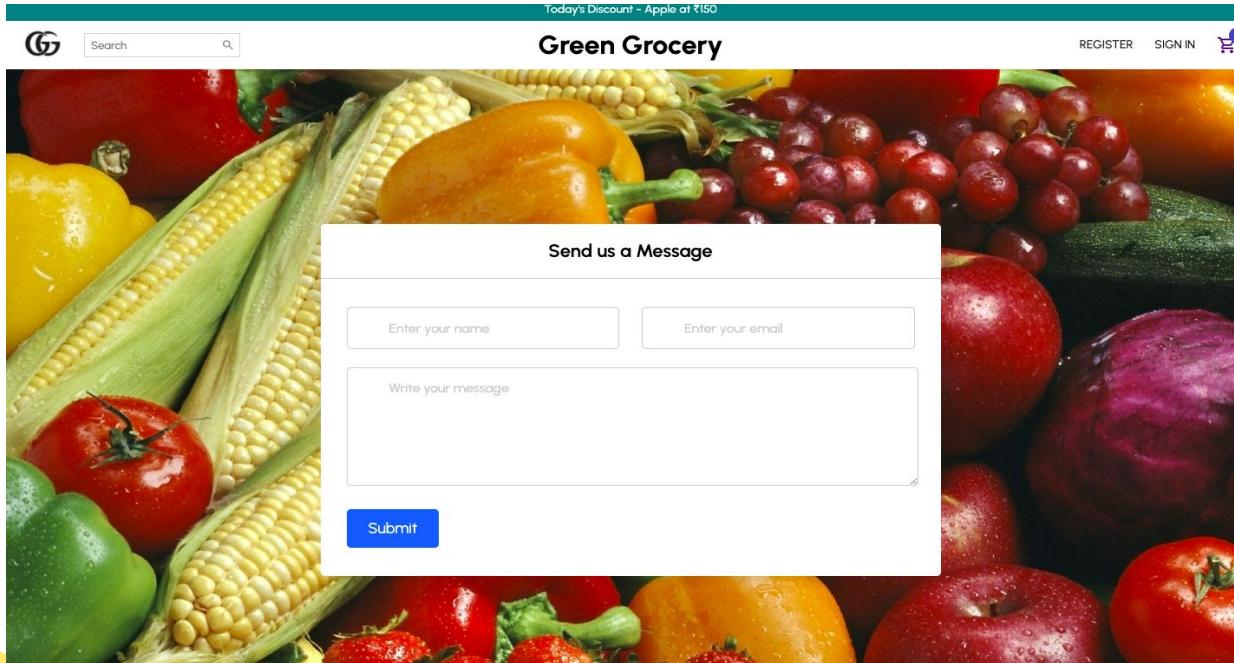
- 1 +

[ADD TO CART](#)



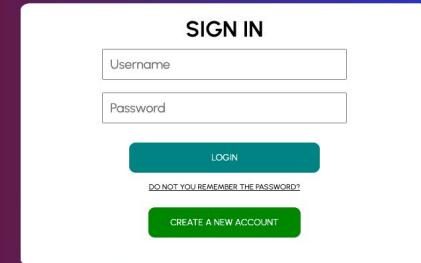
Form -Involves presenting a form to collect user-input, validate, process and post to the backend server

Contact Form Page





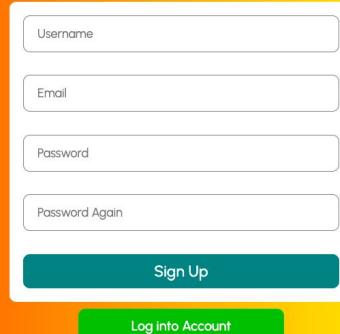
Log In



A sign-in form titled "SIGN IN". It contains fields for "Username" and "Password", a "LOGIN" button, a "DO NOT YOU REMEMBER THE PASSWORD?" link, and a "CREATE A NEW ACCOUNT" button.

Green Grocery

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



A sign-up form with fields for "Username", "Email", "Password", and "Password Again". It includes a "Sign Up" button, a "Log into Account" button, and a "Forgot Password" link.

Sign Up



Account Details



Green Grocery

Hello, NEERAJ [Logout](#)



Today's Discount - Apple at ₹150

Account Details



neeraj

User

👤 neeraj

📅 2021-11-15T12:14:48.974Z

Contact Details

📞 +1123 456 67

✉️ yguvy@gmail.com

📍 India

Account Details

Username

neeraj

Email

yguvy@gmail.com

Phone

+1 123 456 67

Address

India





Shop page with pagination



Green Grocery

SHOP Hello, JAYANTH Logout

Shop

Categories :

[Vegetables](#)

[Fruits](#)

[Dairy](#)



1 2 3



Cart

localhost:3000/cart

The screenshot shows a shopping cart interface for "Green Grocery". At the top right, there are links for "SHOP", "Hello, NEERAJ", and "Logout", along with a shopping cart icon showing a count of 3. The main content area is titled "YOUR BAG". It displays three items: Mango (Quantity: 2, ₹ 80), Guava (Quantity: 1, ₹ 50), and Tomato (Quantity: 2, ₹ 80). To the left of the items is a "CONTINUE SHOPPING" button. On the right is an "ORDER SUMMARY" section detailing the subtotal, shipping, discount, and total.

Green Grocery

Today's Discount - Apple at ₹150

YOUR BAG

CONTINUE SHOPPING

Product: Mango



Quantity: 2
₹ 80

Product: Guava



Quantity: 1
₹ 50

Product: Tomato



Quantity: 2
₹ 80

ORDER SUMMARY

Subtotal	₹ 210
Estimated Shipping	₹ 30.90
Shipping Discount	₹ -5.90
Total	₹ 235

CHECKOUT NOW



Payment and success

YOUR BAG

Green Grocery
Your total is ₹25

Email

Same billing & shipping info

Name

Address

Postcode City

India

Payment Info

YOUR BAG

Green Grocery
Your total is ₹235

4242 4242 4242 4242

12 / 23

Pay \$235.00

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

[Go to Homepage](#)



About us

The screenshot shows the Green Grocery website's About Us page. At the top, there is a navigation bar with the GG logo, the brand name "Green Grocery", and links for "REGISTER", "SIGN IN", and a shopping cart icon with a notification count of 1. The main content area has a light orange background. A large, bold heading "ABOUT US" is centered at the top of the page. Below it, there is a paragraph of text describing the challenges farmers face in selling their products at fair prices and the benefits of buying organic products through the website. At the bottom of the page, there is a button labeled "Contact Us". To the right of the text, there is a circular illustration depicting a group of people working together around a table with laptops and documents.

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.

[Contact Us](#)



Frequently Asked Questions (FAQ)

The screenshot shows a website header with the 'Green Grocery' logo, a search bar, and links for 'REGISTER', 'SIGN IN', and a shopping cart icon. Below the header is a purple box containing the title 'Frequently Asked Questions'. The main content area contains a list of questions and answers:

- 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

 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.

[Privacy Policy](#)

[License](#)



Private Policy

Today's Discount - Apple at ₹150



Green Grocery

Hello, NEERAJ [Logout](#)



Green Grocery

Privacy Policy

Privacy Statement

Green Grocery is committed to protecting your privacy and developing technology that gives you the most powerful and safe online experience. This Statement of Privacy applies to the Green Grocery website and governs data collection and usage. By using our website, you consent to the data practices described in this statement.

[Privacy Policy](#)

[License](#)

Collection of your Personal Information

When you visit our Website, and make a purchase or attempt to make a purchase through our



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

Admin

Dashboard

- Home

Quick Menu

- Users
- Products
- Orders
- Transactions

Notifications

- Contact Forms

Revenue

₹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

User Analytics

Month	Value
Oct	Low
Nov	Medium-Low
Dec	Medium-High
Jan	High



Users

Admin

- Dashboard
- Home
- Quick Menu
 - Users
 - Products
 - Orders
 - Transactions
- Notifications
- Contact Forms

<input type="checkbox"/>	ID	User	Email	Volunteer	Created Time	Action
<input type="checkbox"/>	61924f3864a836a2be1bafe3	 neeraj	yguvy@gmail.com	true	2021-11-15T12:14:48.974Z	<button>View</button>
<input type="checkbox"/>	61935fb84561e45852902dd5	 jakhotia	efwnj@gmail.com	false	2021-11-16T07:37:28.708Z	<button>View</button>
<input type="checkbox"/>	61b1d5d3d8da3f7b362b3aed	 rakesh	rakesh@gmail.com	false	2021-12-09T10:09:23.407Z	<button>View</button>
<input type="checkbox"/>	61b1d749d8da3f7b362b3b00	 harish1	harish1@gmail.com	false	2021-12-09T10:15:37.094Z	<button>View</button>
<input type="checkbox"/>	61b1de26d8da3f7b362b3b7b	 admin	admin@gmail.com	false	2021-12-09T10:44:54.853Z	<button>View</button>
<input type="checkbox"/>	61b387622d104facc51efbf8	 Kanduri Jayanth Sri Ram	jayanthsriram@gmail.com	false	2021-12-10T16:59:14.515Z	<button>View</button>
<input type="checkbox"/>	61b387f22d104facc51efc11	 Jayanth	jayanth@gmail.com	false	2021-12-10T17:01:38.069Z	<button>View</button>
<input type="checkbox"/>	61ba2e9255800d704c55917c	 FSD	fsd@gmail.com	false	2021-12-15T18:06:10.824Z	<button>View</button>
<input type="checkbox"/>	61bb67fb2f44d2e323287eb5	 FSD2	fsd2@gmail.com	false	2021-12-16T16:23:23.087Z	<button>View</button>
<input type="checkbox"/>	61bc52e80958993f16238733	 Himangshu Sir	himanshusir@gmail.com	false	2021-12-17T09:05:44.523Z	<button>View</button>
<input type="checkbox"/>	61d49341ed4ebc5e7114b515	 anirudhjak06	anirudh.jakhotia6@gmail.c...	false	2022-01-04T18:34:41.134Z	<button>View</button>
<input type="checkbox"/>	61d49341ed4ebc5e7114b517	 har	har@gmail.com	false	2022-01-04T18:34:41.541Z	<button>View</button>



Each user

Admin

Dashboard
Home
Quick Menu
Users
Products
Orders
Transactions
Notifications
Contact Forms

Username: neeraj
Email: yguvy@gmail.com

Edit User



neeraj

User

& neeraj

2021-11-15T12:14:48.974Z

Contact Details

+1 123 456 67

yguvy@gmail.com

India



Create

Edit

Username

neeraj

Email

yguvy@gmail.com

Phone

+1 123 456 67

Address

India





Product page

Admin

Dashboard

- Home
- Users
- Products
- Orders
- Transactions

Notifications

- Contact Forms

ID	Product	Stock	Price	Action
6192525b64a836a2be1baff2	Onion	true	50	<button>Edit</button> <button>Delete</button>
61b1dba1de5b32583107653b	Carrot	true	30	<button>Edit</button> <button>Delete</button>
61b1dc48de5b32583107653c	Tomato	true	40	<button>Edit</button> <button>Delete</button>
61b1e1c3de5b32583107653d	Banana	true	40	<button>Edit</button> <button>Delete</button>
61b1e3fede5b32583107653e	Paneer	true	40	<button>Edit</button> <button>Delete</button>
61b1e4ddde5b32583107653f	Spinach	true	45	<button>Edit</button> <button>Delete</button>
61b1e88bde5b325831076540	Mango	true	40	<button>Edit</button> <button>Delete</button>
61b1eee698e7455af4dad66	Ice Cream	true	270	<button>Edit</button> <button>Delete</button>
61b39740e9d47d2d634c5fc7	Orange	true	40	<button>Edit</button> <button>Delete</button>
61b39e071bd6a40ad5397eae	Milk	true	40	<button>Edit</button> <button>Delete</button>
61d877db1a46d5cc3b66ffa7	Guava	true	50	<button>Edit</button> <button>Delete</button>
61d96e0bbe1be55f20213b7	Butter milk	true	40	<button>Edit</button> <button>Delete</button>

1-12 of 23 < >



Each Product page

Admin

Dashboard
Home

Quick Menu
Users
Products
Orders
Transactions

Notifications
Contact Forms

Product

Sales Performance

0 auto

Onion

id:6192525b64a836a2be1baff2
Sales: 0
In stock: Yes

Product Name
Onion

Product Description
Packed With Nutrients.

Price
50

In Stock
Yes

Update



Contact

Admin

NotificationsQuick MenuSettings

ID	User	Email	Message	Action
61d86cec8f9faac68c01a540	Anirudh Jakhotia	anirudh.jakhotia6@gmail.com...	The website UI is real...	<button>View</button>
61d86d118f9faac68c01a542	John	john@gmail.com	This is a really uniqu...	<button>View</button>
61d873e38f9faac68c01a590	Jayanth sriram	jayanthsriram@gmail.com	Testing	<button>View</button>
61d8740f8f9faac68c01a594	Neeraj	neeraj@gmail.com	Testing 2.0	<button>View</button>
61d874368f9faac68c01a599	Harish	harish@gmail.com	The website UI is real...	<button>View</button>
61d874718f9faac68c01a59f	Rakesh	rakesh@gmail.com	The UI is really amazi...	<button>View</button>

1-6 of 6<>



Each contact

Admin

Dashboard
Home

Quick Menu
Users
Products
Orders
Transactions

Notifications
Contact Forms

View Form

Anirudh Jakhotia

User
 Anirudh Jakhotia
 anirudh.jakhotia6@gmail.com

Contact Details
 +1 123 456 67
 The website UI is really cool!
 India

User Details

Username
 Anirudh Jakhotia

Email
 anirudh.jakhotia6@gmail.com

Message
 The website UI is really cool!

Address
 India



Orders

Admin

Dashboard

Home

Quick Menu

Users

Products

Orders

Transactions

Notifications

Contact Forms

ID	Address	Amount	Created Date	Status
61d875841a46d5cc3b66ff21	Hyderabad Amberpet 500013	245	2022-01-07T17:16:52.327Z	pending
61d87b0c1a46d5cc3b66ffd4	4-22-38/A Gowthami Nagar 1st lane,...	925	2022-01-07T17:40:28.758Z	pending
61d87bce1a46d5cc3b66ffe5	Brodipet 4th Lane Guntur 522007	565	2022-01-07T17:43:42.362Z	pending
61d87bfe1a46d5cc3b66fff3	Lakshmpuram 1st lane, Koritepadu ...	265	2022-01-07T17:44:30.404Z	pending
61d9dc73b04e1896a7b468c5	Hyderabad Amberpet 500013	475	2022-01-08T18:48:19.179Z	pending
61d9dcbb6dd825fdee3b6f2f8	Hyderabad Amberpet 500013	105	2022-01-08T18:49:26.385Z	pending
61d9dd94703958a51cbfffd1	Hyderabad Amberpet 500013	85	2022-01-08T18:53:08.763Z	pending
61d9deac08be2632ddacf16a	Hyderabad Amberpet 500013	55	2022-01-08T18:57:48.817Z	pending
61d9e01761c25423c13add57	Hyderabad Amberpet 500013	65	2022-01-08T19:03:51.665Z	pending
61daa2915153360fa1d0bd26	Hyderabad Krishnaraopalem 521214	205	2022-01-09T08:53:37.861Z	pending
61daa4246f785fb9c4789c34	Mern Ashoknagar (Hyderabad) 500020	125	2022-01-09T09:00:20.349Z	pending
61daae9f21da566367574391	testing 3234 Lingampalli 500019	75	2022-01-09T09:45:03.534Z	pending

1-12 of 14 < >



Transactions

Admin

- Dashboard
- Home
- Quick Menu
 - Users
 - Products
 - Orders
 - Transactions
- Notifications
- Contact Forms



<input type="checkbox"/>	Payment ID	User ID	Address	Amount	Created Date
<input type="checkbox"/>	61d875841a46d5cc3b66ff21	61d8749f1a46d5cc3b66ff12	Hyderabad Amberpet 500013	245	2022-01-07T17:16:52.327Z
<input type="checkbox"/>	61d87b0c1a46d5cc3b66ffd4	61924f3864a836a2be1bafe3	4-22-38/A Gowthami Nagar 1st lane, ...	925	2022-01-07T17:40:28.758Z
<input type="checkbox"/>	61d87bce1a46d5cc3b66ffe5	61924f3864a836a2be1bafe3	Brodipet 4th Lane Guntur 522007	565	2022-01-07T17:43:42.362Z
<input type="checkbox"/>	61d87bfe1a46d5cc3b66fff3	61924f3864a836a2be1bafe3	Lakshmiapuram 1st lane, Koritepadu ...	265	2022-01-07T17:44:30.404Z
<input type="checkbox"/>	61d9dc73b04e1896a7b468c5	61924f3864a836a2be1bafe3	Hyderabad Amberpet 500013	475	2022-01-08T18:48:19.179Z
<input type="checkbox"/>	61d9dcbb6dd825fdee3b6f2f8	61924f3864a836a2be1bafe3	Hyderabad Amberpet 500013	105	2022-01-08T18:49:26.385Z
<input type="checkbox"/>	61d9dd94703958a51cbfffd1	61924f3864a836a2be1bafe3	Hyderabad Amberpet 500013	85	2022-01-08T18:53:08.763Z
<input type="checkbox"/>	61d9deac08be2632ddacf16a	61d96a96983986b54ccb451a	Hyderabad Amberpet 500013	55	2022-01-08T18:57:48.817Z
<input type="checkbox"/>	61d9e01761c25423c13add57	61d96a96983986b54ccb451a	Hyderabad Amberpet 500013	65	2022-01-08T19:03:51.665Z
<input type="checkbox"/>	61daaa2915153360fa1d0bd26	61924f3864a836a2be1bafe3	Hyderabad Krishnaraopalem 521214	205	2022-01-09T08:53:37.861Z
<input type="checkbox"/>	61daa4246f785fb9c4789c34	61924f3864a836a2be1bafe3	Mern Ashoknagar (Hyderabad) 500020	125	2022-01-09T09:00:20.349Z
<input type="checkbox"/>	61daae9f21da566367574391	61924f3864a836a2be1bafe3	testing 3234 Lingampalli 500019	75	2022-01-09T09:45:03.534Z



Volunteer

Volunteer

Logout

Dashboard

Products

New Product

Image

No file chosen

Title

Description

Price

Categories

Stock

Create

Swagger UI

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

The screenshot shows the Swagger UI interface for a "Green Grocery" API. At the top left is the Swagger logo (a green circle with white curly braces) and the word "Swagger". Below it, a smaller line of text says "Supported by SMARTBEAR". The main title "Green Grocery" is displayed in large, bold, dark font, with a small "1.0.0" badge next to it. A descriptive subtitle below the title states: "This is Green Grocery Swagger Server for documentation of REST APIs." A link "Find out more about Swagger" is also present. The interface features three expandable sections: "Auth" (Everything about Authorization), "User" (Everything about User Details), and "Contact" (Everything about Contact Details). Each section has a small downward arrow icon to its right, indicating it can be collapsed.

Swagger. Supported by SMARTBEAR

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.

The screenshot shows the 'Models' section of a Swagger UI interface. At the top left is a 'Models' button, and at the top right is a collapse/expand arrow (^). Below the header, there is a list of 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.

```
JS morgan.js ×  
Finals_4 > Finals > api > middlewares > JS morgan.js > ...  
1 //Morgan Middleware  
2 const path = require("path");  
3 const rfs = require("rotating-file-stream");  
4  
5 const logsDir = path.resolve(process.cwd(), "access_logs")  
6 let accessLogStream = rfs.createStream("access.log", {  
7   interval: "1h",  
8   path: logsDir  
9 });  
10  
11 module.exports = { accessLogStream };  
12
```

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.

Multer Middleware

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

```
JS multer.js  X
Finals_4 > Finals > api > middlewares > JS multer.js > ...
1  const multer = require("multer");
2
3  const fileStorageEngine= multer.diskStorage({
4      destination:(req,res,cb)=>{
5          cb(null,"./images")
6      },
7      filename: (req,file,cb) =>{
8          cb(null,Date.now()+"=="+ file.originalname)
9      },
10 });
11 const upload = multer({storage:fileStorageEngine});
12
13 module.exports=upload;
14
```

- 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 look like this when sending.

The screenshot shows the Postman interface for a POST request to `localhost:5000/upload-images`. The 'Body' tab is selected, showing a table with one row. The 'image' field is selected, and its value is `WhatsApp Image 2021-11-26 at 4.40.48 PM.jpeg`. The response tab shows a JSON object indicating successful file upload.

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> image	WhatsApp Image 2021-11-26 at 4.40.48 PM.jpeg			
Key	Value	Description		

```

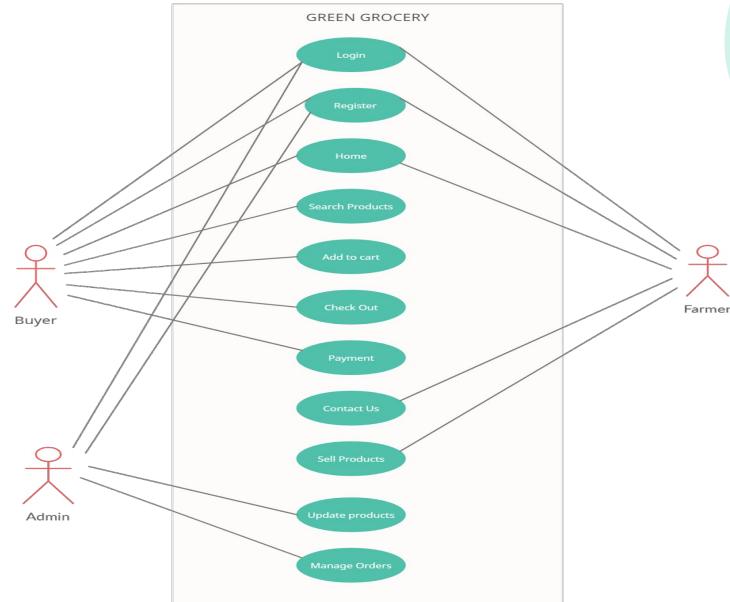
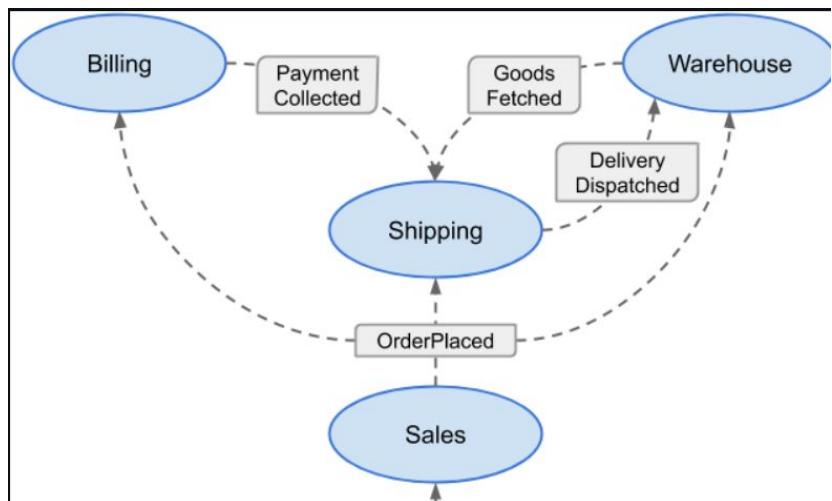
1 "message": "Images Uploaded Successfully",
2 "data": [
3   {
4     "url": "http://res.cloudinary.com/greengrocery/image/upload/v1649054566/Images/so1vikvfpzkyofztkvps.jpg",
5     "id": "Images/so1vikvfpzkyofztkvps"
6   }
7 ]
8
9
  
```

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

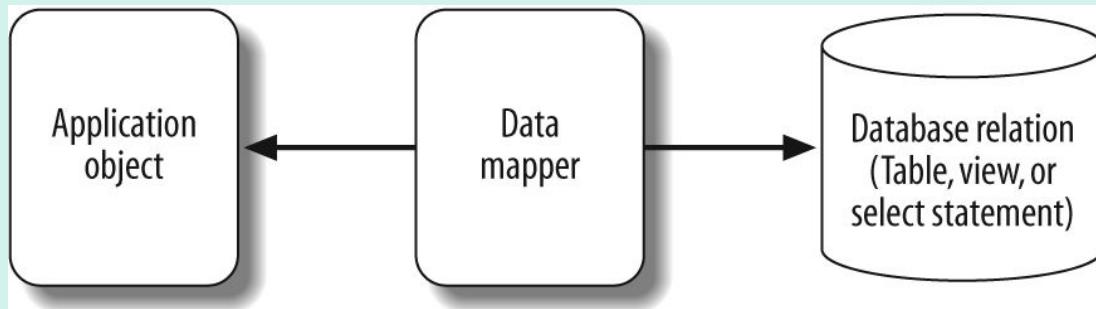
- Tactical design talks about implementation details i.e., modeling domain. It generally takes care of components inside a bounded context.
- Entity, Value Objects, Services, Aggregates, Factories and Repositories tools are high-level concepts that can be used to create and modify domain models.

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 (ODM)

- ❑ 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 :



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.



Task Assigned

Jayanth Sri Ram Kanduri - Each Product Categories, Navbar, About Us, Volunteer (Frontend and Backend), Cloudinary API, Multer Middleware.

Rakesh Ganeshula - Contact us, Contact (Backend), Account Details, License and Privacy Policy, Stripe, Swagger API, Multer Middleware.

Harish Mullagura - Each product details, Contact(FrontEnd) Products (Backend), Order (Backend), Filters, Morgan Middleware, Multer Middleware.

Anirudh Jakhotia - Slider, Footer, Users (Backend), Shop, Pagination, Faq, Admin Home, Graphs, Swagger API, Morgan Middleware, DDD.

Neeraj Dusa - Login, Sign-Up, Payments, Success, Checkout (Backend), Volunteer (Frontend and Backend), Domain Driven Designs, Cloudinary API.





Thanks!

**DO YOU HAVE ANY
QUESTIONS?**

fsdproj30@gmail.com
+91 98765 43210
GreenGrocery.com