BSc (Hons) in Software Engineering

SE3020 -Distributed Systems - Semester 1, 2022

Assignment 2 – REST API

Version 03

A collaborative Agri product online purchasing platform

Name	IT number
C. D. Adhihetty	IT20133672
Gunasekara A.M.P.P.	IT20135966
R.A.E. Wijenayake	IT20133122
Perera M.P.M.	IT20133818

Table of Contents

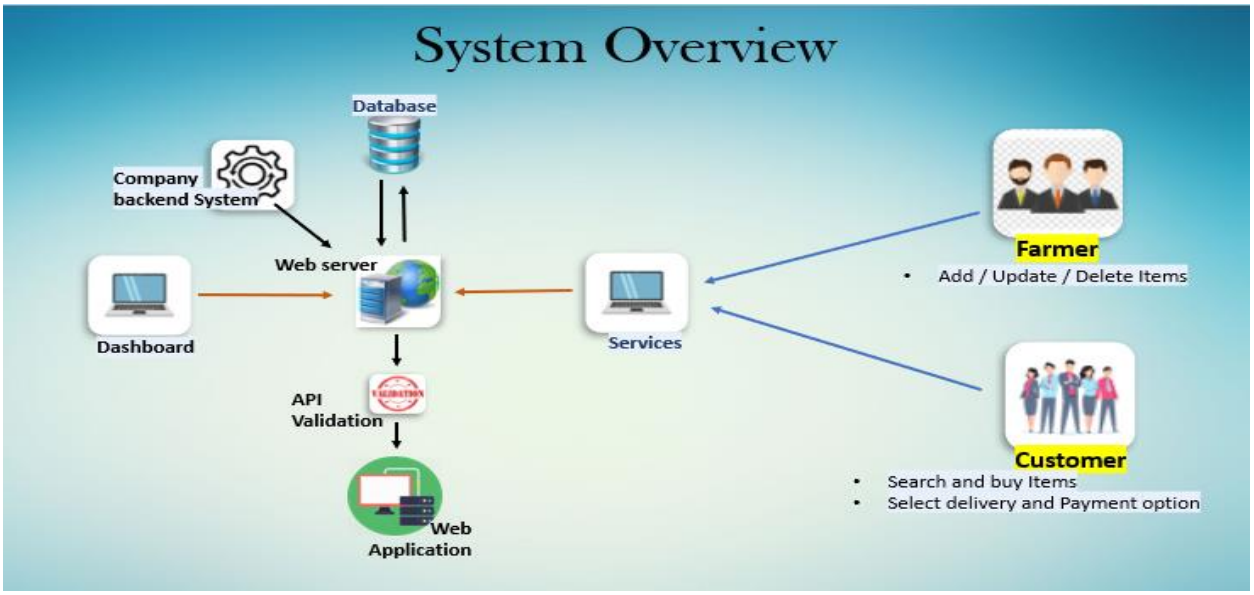
1. Introduction

1.1 Purpose

Through this website, farmers are able to add their different types of Agri products to the online store. And also, they have access to update previously added product details and remove those products if they want. When customers are visiting this web app, they can see a variety of Agri products. Furthermore, this platform provides a feature to search for the Agri product as customers wish. Through this, customers can find out their requirements easily. The system has provided another facility for their customers to make many purchases at once of various items. For this customers can use the shopping cart feature. And also, customers are able to be delivered the purchased items easily using a third party companies such as Pick me and Uber. For customer convenience, the system provides a card payment method for the process of paying bills.

1.2 System Overview

High level architectural diagram



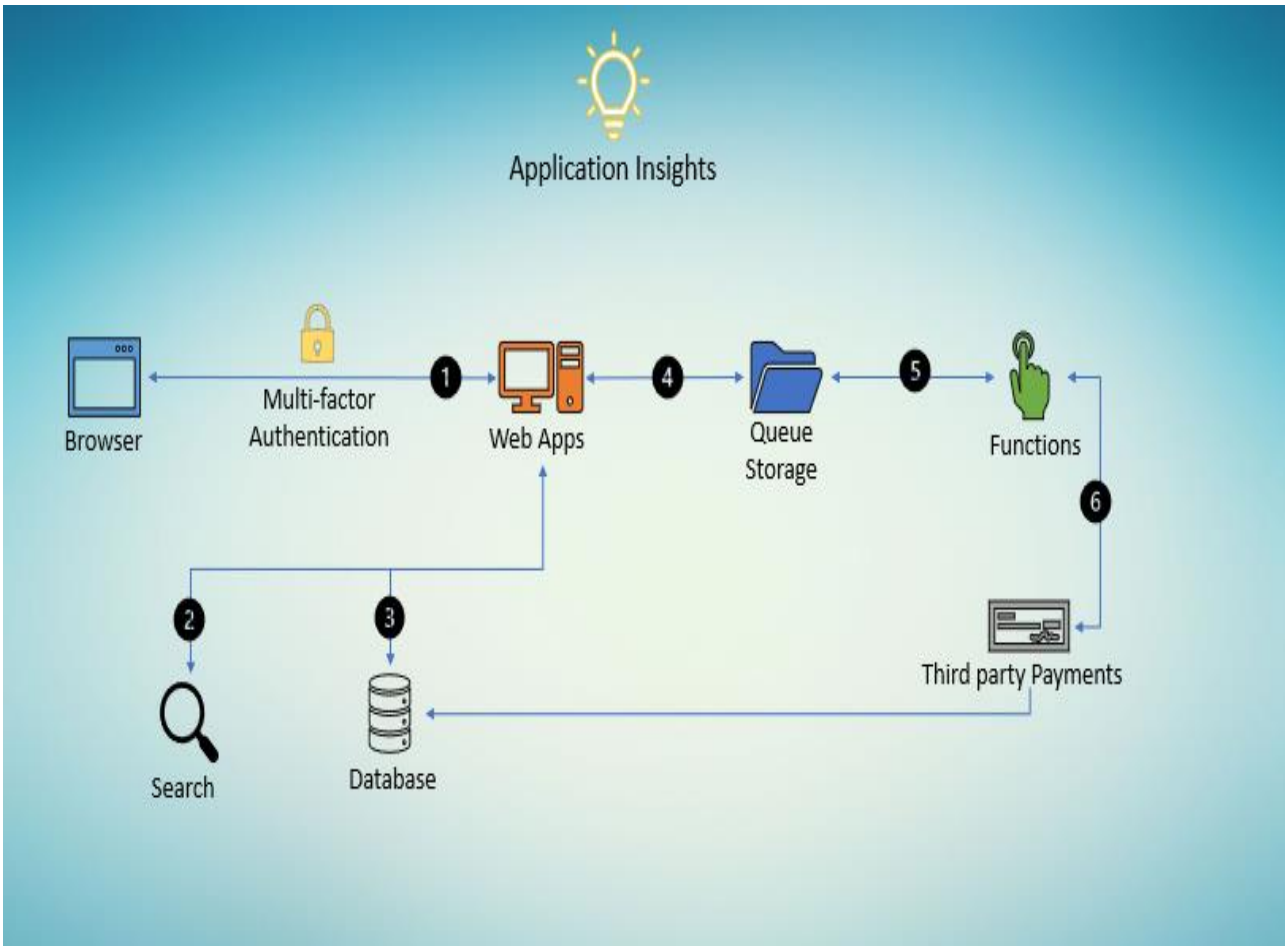
2. Overall Description

2.1 Product Perspective

Through a web interface that is provided by the platform, customers have the ability to go shopping for products that have been published by farmers. Customers have access to services that allow them to search for products and purchase them, while farmers have access to services that allow them to add, edit, and remove items from their online store. The customers are able to make many purchases at once of a variety of goods. Once an item or combination of items has been added to the shopping cart, the consumer has the flexibility to select the desired delivery option from the available options. There are two distinct options available for the customer to choose from when it comes to paying the bill.

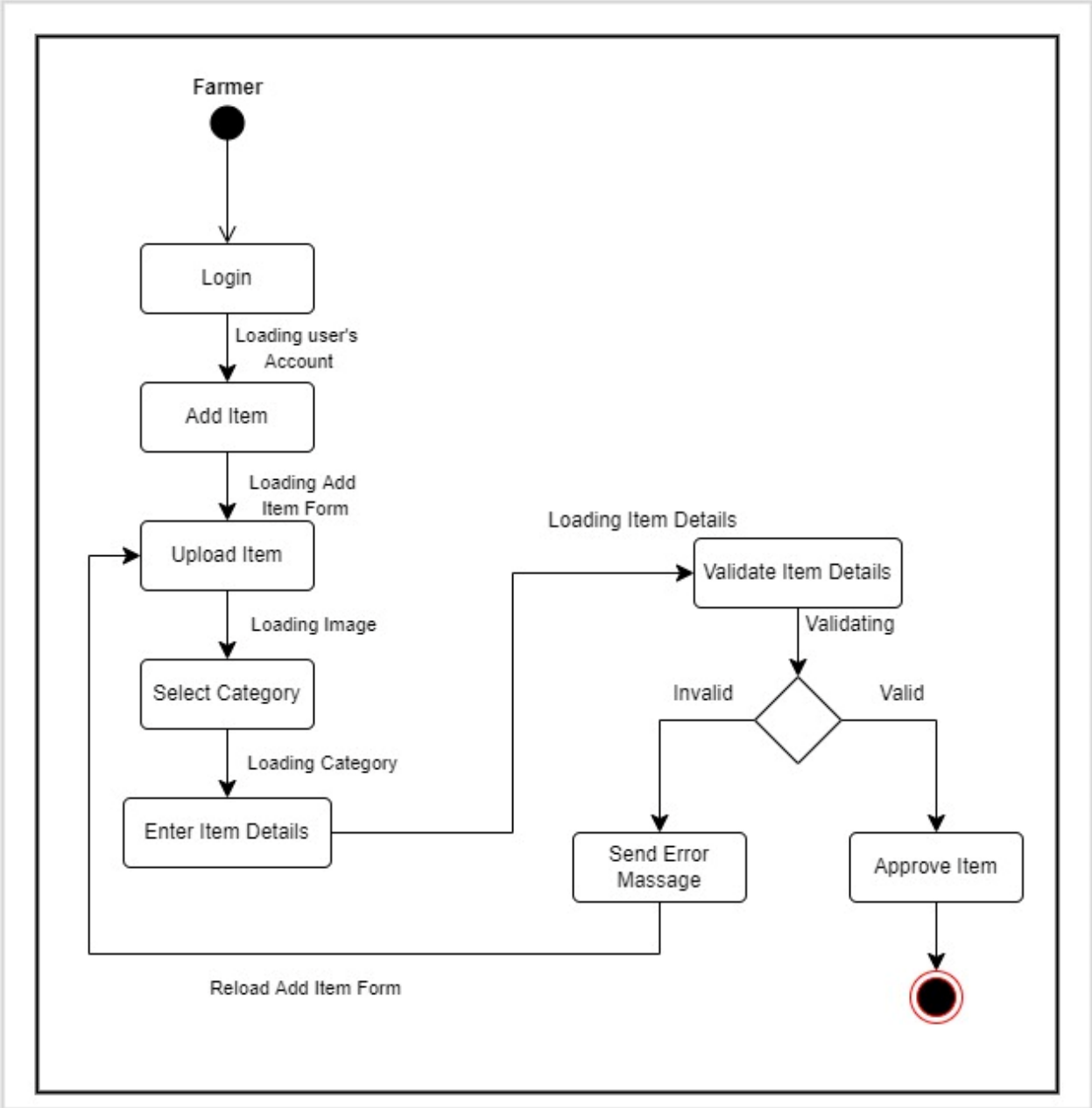
2.2 Product Features

3. System Feature



3.1 Item management

Mainly, there are four types of tasks are done by farmers. For that, firstly farmers need to log in to the system. Then farmers can add their Agri items by filling out the adding items form. Here farmers have to add an image of the product, select the item category and enter item details. When the entered item details are the invalid system will show an error message. Otherwise, the item will be approved by the system.

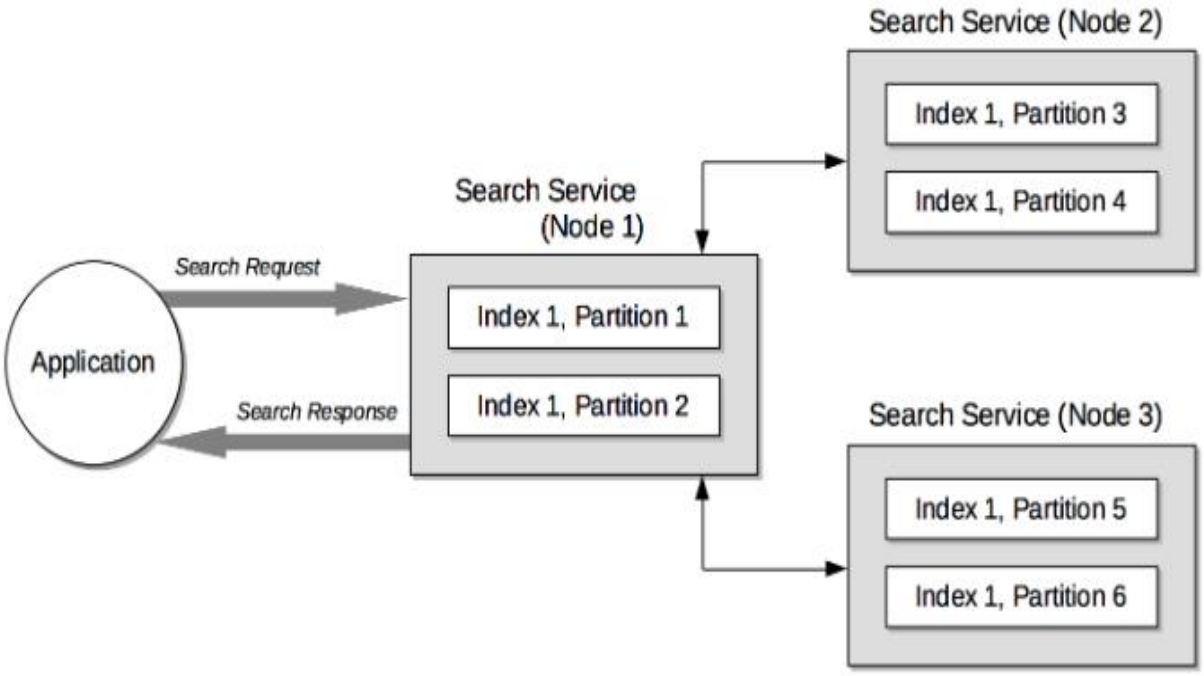

Farmer
Login
Loading user's Account
Add Item
Loading Add Item Form
Upload Item
Loading Image
Select Category
Loading Category
Enter Item Details
Loading Item Details
Validate Item Details
Validating
Invalid
Valid
Send Error Massage
Approve Item
Reload Add Item Form

3.2 Search Items

Individually searchable are all available Search Service nodes in the cluster. When a specific Search Service node is selected for a search request, it assumes the function of coordinator and is responsible for applying the search request to the other Search Service nodes and delivering results.

The following diagram depicts how the search service operates.

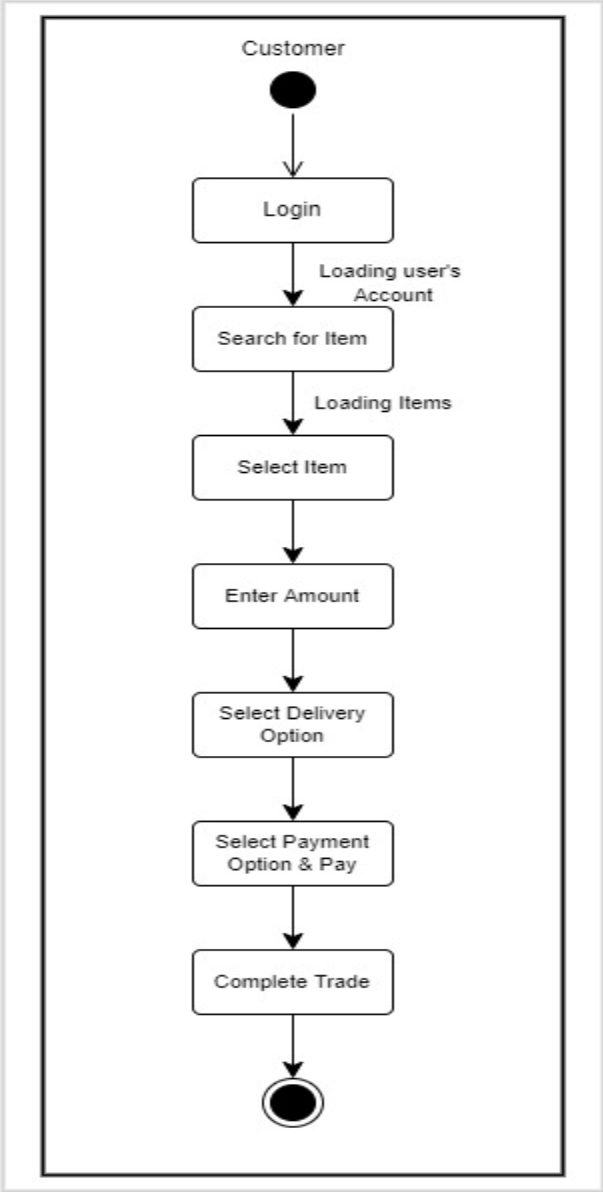
Following is depicted in the diagram:



Method by which an application sends a search request to a specified Search Service node (here, Node 1). This node is the coordinator of the network. How the coordinator distributes search requests to all other search-index partitions (in this case, Nodes 2 and 3) in the cluster. After collecting all returned data, the coordinator applies the required filters and sends the user the final results.

3.3 Buy Items (select delivery, payment option and & buy)

As a user, the customer needs to log in to the system. Then they can search Agri items by item categories and add those Agri items to the shopping cart. Then the system will display the total amount that customer should pay. As the next step customer has to select the delivery option. After that system will navigate to the payment options page. By entering the valid card details, the customer can be able to succeed in the payment process.



4. Tools and Technology

MongoDB

A document-oriented database that does not use the SQL standard for storage of the application's data.

ExpressJS

Express is the most widely used online application framework that is built on top of NodeJS.

NodeJS

It uses the methods and structures that NodeJS gives it to build the back end of the web application.

ReactJS

It is used to build the front end of a web application, which makes it easier to make user interfaces that are dynamic and interactive.

NodeJS

It is used to run JavaScript on a machine rather than in a browser so that code can be executed more quickly.

Visual Studio Code

The integrated development environment (IDE) that we rely on to create the web application is called Visual Studio Code.

REST API

A Representational State Transfer (REST) interface is a set of operations that can be called by a remote client (which could be another service) over the HTTP protocol and a network. Usually, the client will give parameters like a search string or the name of a resource that needs to be deleted.

There is a rule about which HTTP method to use for what kind of operation:

GET: Fetching data

POST: Brings new data

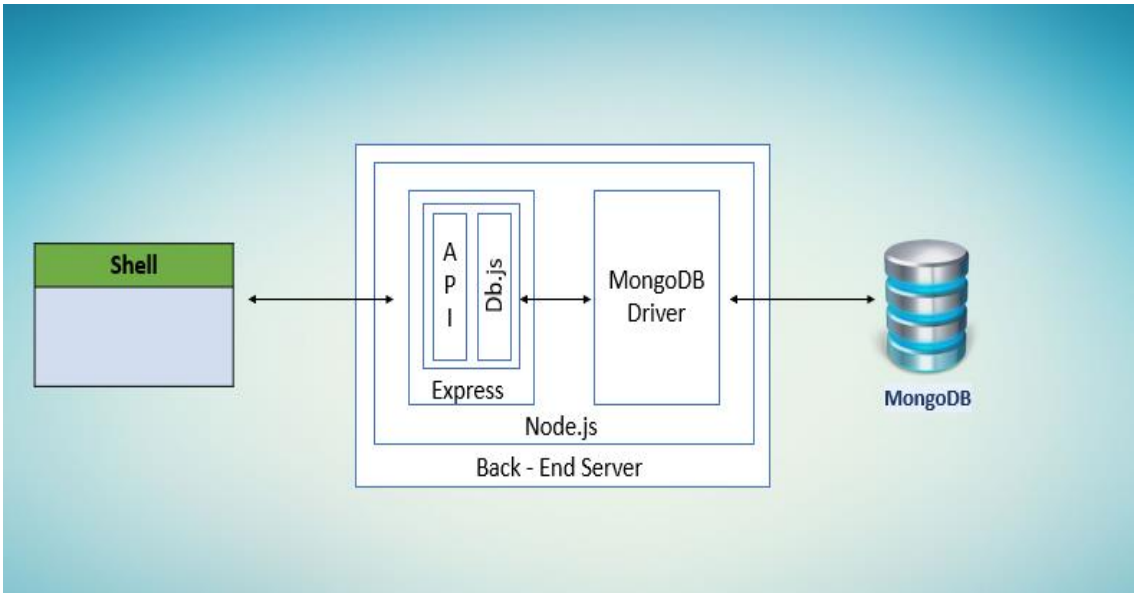
PUT: This method updates data.

DELETE: Removes data

This rule is broken by Mongopop's REST API, which uses POST for some read requests instead of GET because it is easier to pass arguments with POST. Express is executed within the Node.js environment as a separate module. Express can control how requests move through our app and send them to the right functions (or to different apps running in the same environment).

Express allows us to run the entire app's business logic, and we can even utilize an optional view engine to build the final HTML that will be rendered by the user's browser. Both of these features are available to users. On the other hand, Express can be used to merely provide a REST API, which grants the front-end app access to the resources to which it requires access, such as the database. This is an extreme example. Express is used to accomplish two different functions for the Mongopop application: When a user navigates to our application, have the code for the front end of the application sent to the remote client. Make available a REST application programming interface (API) that the front end can use to access the database through HTTP network calls.

4.1 Architecture



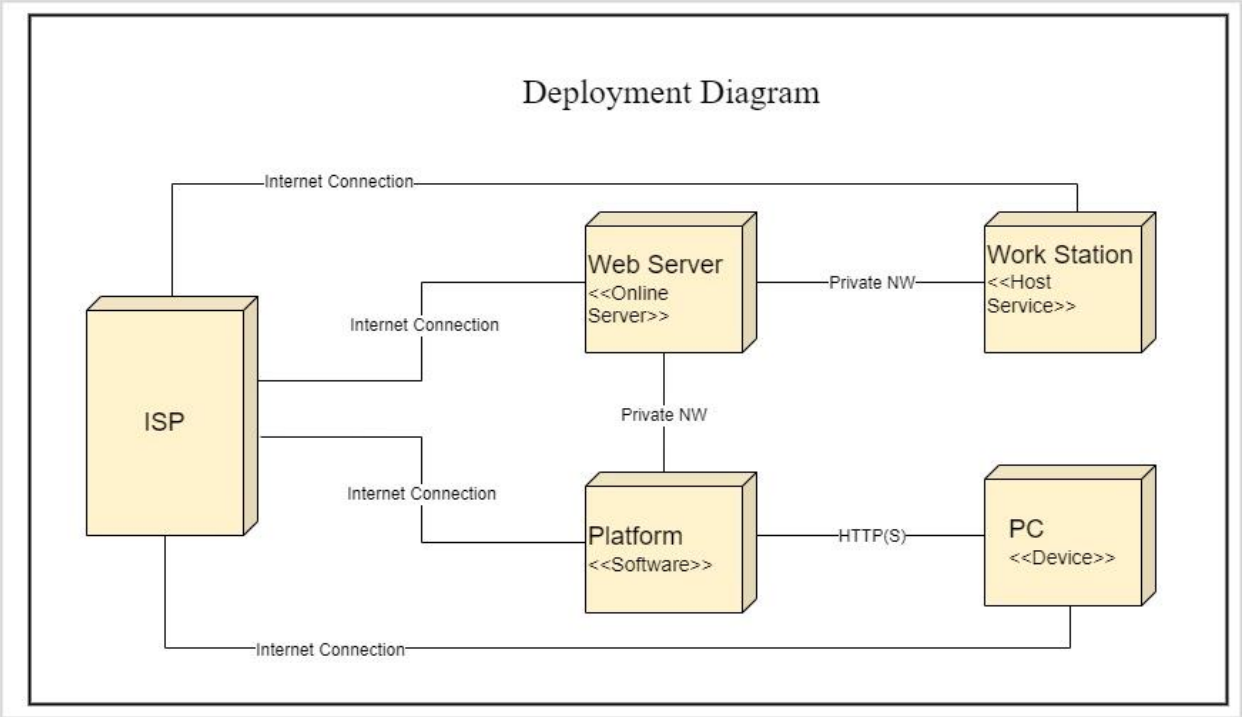
All files

- package.json is a file that instructs the Node.js package manager (npm) on what it should be doing, such as installing the required dependencies packages. This file is located in the Node.js installation directory.
- When packages are obtained, npm will install them in the directory known as "node modules," which is the default location for this directory.
- The Node.js driver for MongoDB is referred to as the node modues/mongodb package.
- node modules/mongodb-core is a low-level MongoDB driver library that is made available for framework developers (application developers should avoid using it directly)
- javascripts/db.js is a JavaScript module that we've created in order for our Node.js applications (for the reasons of this series, it will be Express) to use in order to access MongoDB; this module,

in turn, utilizes the MongoDB Node.js driver. For the purposes of this series, Express will be the application that will access MongoDB.

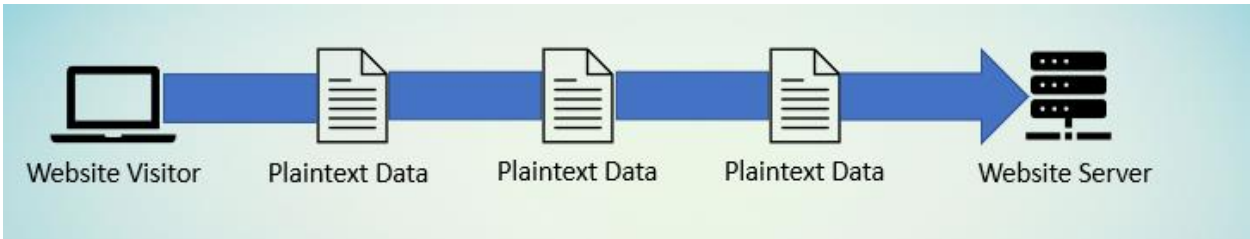
- The following is a list of additional directories and files that are necessary for our Express application:
- The application-specific application depends are saved in the file referred to as config.js, which is the name of the file.
- bin/www is the script that initiates the execution of an Express application. The npm start script that is included in the package.json file is the one that is responsible for triggering the execution of this script. The HTTP server is created, and it is instructed to use the app module that is specified in app.js. app.js is the name of the primary application module that is specified (app). It is possible to configure the program so that it will be executed using Express.
- Which of the routes will be accessible, in addition to the precise locations of those routes within the file system (routes directory)
- views: a directory containing the templates that will be used by the Jade view engine to assemble the HTML for any pages generated by the Express application (for this application, this is only the error page that is used in circumstances such as mistyped routes ("404 Page not found"); for example,
- routes: This is the execution of the Mongopop REST API, which is a directory containing one JavaScript file for each Express route. The directory is organized in a way that makes it easy to navigate. Here, users will discover all of the procedures that are associated with each of the supported routes.
- public: This directory contains all of the static files that a distant client must be able to read in order for the directory to function properly (React apps).

4.2 Hardware Interfaces



4.3 Communications Interfaces

HTTP is an application layer protocol and stands for hypertext transfer protocol. In our platform's security, it is a protocol that enables backend and client internet communication. It is regarded as a stateless system since it offers connections on demand to browsers that initiate them. HTTP was developed before HTTPS.
HTTP utilizes TCP (transmission control protocol) and port 80 to exchange data packets across the Internet. After the first handshake between the client and server, the HTTP server will employ several request messages.



5. Other Nonfunctional Requirements

5.1 Performance

Farmers can add their different kinds of Agri products to the online store by using this website. They can also change the information about products they've already added and remove those products if they want to. Customers can see a range of Agri products when they use this web app. Also, this platform lets customers search for Agri products in any way they want. This makes it easy for customers to find out what they need. The system has given their customers another way to buy a lot of different things at once. Customers can do this by using the "shopping cart" feature. And third-party companies like Pick me and Uber make it easy for customers to get their purchases delivered. The system makes it easy for customers to pay their bills by letting them use card payments.

5.2 Safety & Security

When a customer tries to purchase an item from this store, the system will request to log in first. So, the customer has to register to the system. After the registration process, login credentials will be generated for each customer. It will be unique for each customer account. So, the personal details of the customer and the payment details will be secured from other parties. This method further ensures the safety of the customers.

Appendix

[1] Backend Part

Sever Js

```
const express = require('express');
let bodyParser = require('body-parser');
const farmerRoutes = require('./routes/farmer');
const customerRoutes = require('./routes/customer');
const authRoutes = require('./routes/auth');
const deliveryRoutes = require('./routes/delivery');
const mongoose = require('mongoose');
require('dotenv').config();

const app = express();
const port = process.env.PORT || "5000";
app.use(bodyParser.json({limit: '10MB', extended: true}))
app.use(bodyParser.urlencoded({limit: '10MB', extended: true}))

mongoose.connect(`mongodb+srv://user:user123@bigheartagriproducts.2nbik.mongodb.n
et/BigHeartAgriProducts?retryWrites=true&w=majority`, { useNewUrlParser:
true,  useUnifiedTopology: true })
  .then(() => {
    console.log("Connected to database!");
  })
  .catch(() => {
    console.log("Connection failed!");
  });

app.use((req,res,next) => {
  res.setHeader("Access-Control-Allow-Origin","*");
  res.setHeader("Access-Control-Allow-Headers","Origin, X-Requested-With, Accept,
Content-type");
  res.setHeader("Access-Control-Allow-Methods","GET, POST, PUT, PATCH, DELETE,
OPTIONS");
  next();
});

app.use('/farmer', farmerRoutes)
app.use('/customer', customerRoutes);;
app.use('/delivery', deliveryRoutes);
app.use('/auth', authRoutes);
app.listen(port, () => {
```

```
  console.log(`Listening to requests on http://localhost:${port}`);
});
module.exports = app;
```

Model classes

customerAuthData.js

```
const mongoose = require('mongoose');

const customerAuthSchema = mongoose.Schema({
    name: {type: String, required: true},
    email: {type: String, required: true},
    pwd: {type: String, required: true},
    phone: {type: String, required: true},
    type: {type: String, required: true}
});

module.exports = mongoose.model('CustomerAuthList', customerAuthSchema);
```

farmerAuthData.js

```
const mongoose = require('mongoose');

const farmerAuthSchema = mongoose.Schema({
    name: {type: String, required: true},
    email: {type: String, required: true},
    pwd: {type: String, required: true},
    phone: {type: String, required: true},
    type: {type: String, required: true}
});

module.exports = mongoose.model('FarmerAuthList', farmerAuthSchema);
```

productData.js

```
const mongoose = require('mongoose');

const prodSchema = mongoose.Schema({
    ptype: {type: String, required: true},
    pcategory: {type: String, required: true},
    pname: {type: String, required: true},
    pimg: {type: String, required: true},
```

```
    pno: {type: String, required: true},
    pprice: {type: String, required: true},
    pdiscount: {type: String, required: false},
    pdesc: {type: String, required: true}
});

module.exports = mongoose.model('ProductList', prodSchema);
```

forageSeeds.js

```
const mongoose = require('mongoose');

const forageSeedsSchema = mongoose.Schema({
    ptype: {type: String, required: true},
    pcategory: {type: String, required: true},
    pname: {type: String, required: true},
    pimg: {type: String, required: true},
    pno: {type: String, required: true},
    pprice: {type: String, required: true},
    pdiscount: {type: String, required: false},
    pdesc: {type: String, required: true},
    preview: [{
        pratings: {type: String, required: false},
        pcomments: {type: String, required: false}
    }]
});

module.exports = mongoose.model('forageSeedsList', forageSeedsSchema);
```

horticultureSeeds.js

```
const mongoose = require('mongoose');

const horticultureSeedsSchema = mongoose.Schema({
    ptype: {type: String, required: true},
    pcategory: {type: String, required: true},
    pname: {type: String, required: true},
    pimg: {type: String, required: true},
    pno: {type: String, required: true},
    pprice: {type: String, required: true},
    pdiscount: {type: String, required: false},
    pdesc: {type: String, required: true},
    preview: [{
        pratings: {type: String, required: false},
```

```
        pcomments: {type: String, required: false}
    }]
});

module.exports = mongoose.model('horticultureSeedsList',
horticultureSeedsSchema);
```

plantNuritions.js

```
const mongoose = require('mongoose');

const plantNutritionsSchema = mongoose.Schema({
    ptype: {type: String, required: true},
    pcategory: {type: String, required: true},
    pname: {type: String, required: true},
    pimg: {type: String, required: true},
    pno: {type: String, required: true},
    pprice: {type: String, required: true},
    pdiscount: {type: String, required: false},
    pdesc: {type: String, required: true},
    preview: [{
        pratings: {type: String, required: false},
        pcomments: {type: String, required: false}
    }]
});

module.exports = mongoose.model('plantNutritionsList', plantNutritionsSchema);
```

tools.js

```
const mongoose = require('mongoose');

const toolsSchema = mongoose.Schema({
    ptype: {type: String, required: true},
    pcategory: {type: String, required: true},
    pname: {type: String, required: true},
    pimg: {type: String, required: true},
    pno: {type: String, required: true},
    pprice: {type: String, required: true},
    pdiscount: {type: String, required: false},
    pdesc: {type: String, required: true},
    preview: [{
        pratings: {type: String, required: false},
        pcomments: {type: String, required: false}
```

```
    }]
});

module.exports = mongoose.model('ToolsList', toolsSchema);
```

Routes

auth.js

```
const express = require('express');
const router = express.Router();
const CustomerAuthList = require('../models/customer/customerAuthData');
const FarmerAuthList = require('../models/Farmer/farmerAuthData');

router.post('/customer/signup', (req,res,next) => {
    console.log(req.body);
    CustomerAuthList.find({email: req.body.email})
        .then(data => {
            console.log(data);
            if(data.length>0) {
                res.sendStatus(404);
            } else {
                const user = new CustomerAuthList({name: req.body.name, email:
req.body.email, phone: req.body.phone, pwd: req.body.password, type:
req.body.type});
                user.save().then(result => {
                    console.log(result._id);
                    res.status(201).json({message: "User registered
successfully!!"});
                });
            }
        })

})

router.post('/customer/signin', (req,res,next) => {
    console.log(req.body);
    CustomerAuthList.find({email: req.body.email})
        .then(data => {
            console.log(data);
            if(data.length>0) {
                if(data[0].pwd === req.body.password) {
                    res.status(201).json({message: "User signed in
successfully!!", token: data[0].type, custid: data[0].email});
```

```
                } else {
                    res.sendStatus(404);
                }
            } else {
                res.sendStatus(404);
            }
        })

})
router.post('/farmer/signup', (req,res,next) => {
    console.log(req.body);
    FarmerAuthList.find({email: req.body.email})
        .then(data => {
            console.log(data);
            if(data.length>0) {
                res.sendStatus(404);
            } else {
                const user = new FarmerAuthList({name: req.body.name, email:
req.body.email, phone: req.body.phone, pwd: req.body.password, type:
req.body.type});
                user.save().then(result => {
                    console.log(result._id);
                    res.status(201).json({message: "User registered successfully
as Farmer!!"});
                });
            }
        })

})


router.post('/farmer/signin', (req,res,next) => {
    console.log(req.body);
    FarmerAuthList.find({email: req.body.email})
        .then(data => {
            console.log(data);
            if(data.length>0) {
                if(data[0].pwd === req.body.password) {
                    res.status(201).json({message: "User signed in
successfully!!", token: data[0].type, custid: data[0].email});
                } else {
                    res.sendStatus(404);
                }
            } else {
                res.sendStatus(404);
```

```
            }
        })

})

module.exports = router;
```

customer.js

```
const express = require('express');
const router = express.Router();
const plantNutritionsList = require('../models/customer/plantNutritions');
const ToolsList = require('../models/customer/tools');
const forageSeedsList = require('../models/customer/forageSeeds');
const horticultureSeedsList = require('../models/customer/horticultureSeeds');
const ObjID = require('mongoose').ObjectID ;

//plantNutritions routes
router.post('/productList/plantNutritions/add', (req,res,next) => {
    //console.log('Backend', req.body[0].pname);
    plantNutritionsList.remove({})
        .then((response) => {
            for(let i=0;i<req.body.length;i++) {
                const product = new plantNutritionsList({pimg:
req.body[i].pimg,ptype: req.body[i].ptype,pcategory: req.body[i].pcategory,pname:
req.body[i].pname,pno: req.body[i].pno,pprice: req.body[i].pprice,pdiscount:
req.body[i].pdiscount,pdesc: req.body[i].pdesc,preview:[]});
                product.save().then(result => {
                    console.log(result._id);
                  });
                if(i === req.body.length-1) {
                    res.status(201).json({message: "Plant Nutritions List added
successfully!!"});
                }
            }
        })
})


router.get('/plantNutritions/data', (req,res,next) => {
    plantNutritionsList.find({})
        .then((response) => {
            res.status(201).json({message: "Plant Nutritions List received
successfully!!", data: response});
```

```
        })
})
router.post('/productList/plantNutritions/update', (req,res,next) => {
    const edevice = req.body;
    for(let i=0;i<edevice.length;i++) {
        plantNutritionsList.findById(edevice[i]._id)
            .then(obj => {
            if(Number(obj.pno) > Number(edevice[i].pno)){
                obj.pno = (Number(obj.pno) - Number(edevice[i].pno)).toString();
            } else if(Number(obj.pno) === Number(edevice[i].pno)) {
                obj.pno = "0";
            }
            console.log(obj._id, obj.pno);
            plantNutritionsList.updateOne({_id: obj._id}, obj)
                .then(result => {
                    console.log("Update Successful!!");
                })
        });
        if(i === edevice.length-1) {
            res.status(201).json({message: "Plant Nutritions List updated
successfully!!"});
        }
    }
})
//Tools routes

router.post('/productList/tools/add', (req,res,next) => {
    //console.log('Backend', req.body[0].pname);
    ToolsList.remove({})
        .then((response) => {
            for(let i=0;i<req.body.length;i++) {
                const product = new ToolsList({pimg: req.body[i].pimg,ptype:
req.body[i].ptype,pcategory: req.body[i].pcategory,pname: req.body[i].pname,pno:
req.body[i].pno,pprice: req.body[i].pprice,pdiscount:
req.body[i].pdiscount,pdesc: req.body[i].pdesc,preview:[]});
                product.save().then(result => {
                    console.log(result._id);
                  });
                if(i === req.body.length-1) {
                    res.status(201).json({message: "tools list added
successfully!!"});
                }
            }
        })
})
```

```
router.get('/tools/data', (req,res,next) => {
    ToolsList.find({})
        .then((response) => {
            res.status(201).json({message: "tools list received successfully!!",
data: response});
        })
})

router.post('/productList/tools/update', (req,res,next) => {
    const clothes = req.body;
    for(let i=0;i<clothes.length;i++) {
        ToolsList.findById(clothes[i]._id)
            .then(obj => {
            if(Number(obj.pno) > Number(clothes[i].pno)){
                obj.pno = (Number(obj.pno) - Number(clothes[i].pno)).toString();
            } else if(Number(obj.pno) === Number(clothes[i].pno)) {
                obj.pno = "0";
            }
            console.log(obj._id, obj.pno);
            ToolsList.updateOne({_id: obj._id}, obj)
                .then(result => {
                    console.log("Update Successful!!");
                })
        });
        if(i === clothes.length-1) {
            res.status(201).json({message: "tools list updated successfully!!"});
        }
    }
})

//forage Seeds routes

router.post('/productList/forageSeeds/add', (req,res,next) => {
    //console.log('Backend', req.body[0].pname);
    forageSeedsList.remove({})
        .then((response) => {
            for(let i=0;i<req.body.length;i++) {
                const product = new forageSeedsList({pimg:
req.body[i].pimg,ptype: req.body[i].ptype,pcategory: req.body[i].pcategory,pname:
req.body[i].pname,pno: req.body[i].pno,pprice: req.body[i].pprice,pdiscount:
req.body[i].pdiscount,pdesc: req.body[i].pdesc,preview:[]});
                product.save().then(result => {
                    console.log(result._id);
```

```
                });
                if(i === req.body.length-1) {
                    res.status(201).json({message: "forageSeeds list added
successfully!!"});
                }
            }
        })
})
router.get('/forageSeeds/data', (req,res,next) => {
    forageSeedsList.find({})
        .then((response) => {
            res.status(201).json({message: "forageSeeds list received
successfully!!", data: response});
        })
})

router.post('/productList/forageSeeds/update', (req,res,next) => {
    const food = req.body;
    for(let i=0;i<food.length;i++) {
        forageSeedsList.findById(food[i]._id)
            .then(obj => {
            if(Number(obj.pno) > Number(food[i].pno)){
                obj.pno = (Number(obj.pno) - Number(food[i].pno)).toString();
            } else if(Number(obj.pno) === Number(food[i].pno)) {
                obj.pno = "0";
            }
            console.log(obj._id, obj.pno);
            forageSeedsList.updateOne({_id: obj._id}, obj)
                .then(result => {
                    console.log("Update Successful!!");
                })
        });
        if(i === food.length-1) {
            res.status(201).json({message: "forageSeeds list updated
successfully!!"});
        }
    }
})
//Horticulture seeds routes


router.post('/productList/horticultureSeeds/add', (req,res,next) => {
    //console.log('Backend', req.body[0].pname);
    horticultureSeedsList.remove({})
        .then((response) => {
```

```
            for(let i=0;i<req.body.length;i++) {
                const product = new horticultureSeedsList({pimg:
req.body[i].pimg,ptype: req.body[i].ptype,pcategory: req.body[i].pcategory,pname:
req.body[i].pname,pno: req.body[i].pno,pprice: req.body[i].pprice,pdiscount:
req.body[i].pdiscount,pdesc: req.body[i].pdesc,preview:[]});
                product.save().then(result => {
                    console.log(result._id);
                  });
                if(i === req.body.length-1) {
                    res.status(201).json({message: "HorticultureSeeds list added
successfully!!"});
                }
            }
        })
})

router.get('/horticultureSeeds/data', (req,res,next) => {

    horticultureSeedsList.find({})
        .then((response) => {
            res.status(201).json({message: "HorticultureSeeds list received
successfully!!", data: response});
        })
})

router.post('/productList/horticultureSeeds/update', (req,res,next) => {

    const furniture = req.body;
    for(let i=0;i<furniture.length;i++) {
        horticultureSeedsList.findById(furniture[i]._id)
            .then(obj => {
            if(Number(obj.pno) > Number(furniture[i].pno)){
                obj.pno = (Number(obj.pno) -
Number(furniture[i].pno)).toString();
            } else if(Number(obj.pno) === Number(furniture[i].pno)) {
                obj.pno = "0";
            }
            console.log(obj._id, obj.pno);
            horticultureSeedsList.updateOne({_id: obj._id}, obj)
                .then(result => {
                    console.log("Update Successful!!");
                })
        });
        if(i === furniture.length-1) {
```

```
            res.status(201).json({message: "HorticultureSeeds list updated
successfully!!"});
        }
    }
})

module.exports = router;
```

farmer.js

```
const express = require('express');
const router = express.Router();
const ProductList = require('../models/Farmer/productData');
var ObjectId = require('mongodb').ObjectID;


router.get('/', (req,res,next) => {
    ProductList.find({})
        .then(data => {
            res.status(201).json({message: "Product details fetched
successfully!!", data: data});
        })

})

router.post('/productList/add', (req,res,next) => {
    console.log('Backend', req.body);
    const product = new ProductList({pimg: req.body.pimg,ptype:
req.body.ptype,pcategory: req.body.pcategory,pname: req.body.pname,pno:
req.body.pno,pprice: req.body.pprice,pdiscount: req.body.pdiscount,pdesc:
req.body.pdesc});
    product.save().then(result => {
        console.log(result._id);
        res.status(201).json({message: "Product list received successfully!!"});
      });
})

router.delete('/productList/delete', (req,res,next) => {
    // console.log('Product Id to be deleted:', req.body.id);
    ProductList.deleteOne({'_id': ObjectId(req.body.id)})
        .then((result) => {
            res.status(201).json({message: "Product deleted successfully!!"});
        },
        (error) => {
            res.status(500).json({message: "Product deletion unsuccessful!!"})
```

```
        })
})

router.put('/productList/update', (req,res,next) => {
    console.log('Product Id to be updated:', req.body.data.data.pid);
    ProductList.updateOne({'_id': ObjectId(req.body.data.data.pid)}, {pimg:
req.body.data.data.pimg,ptype: req.body.data.data.ptype,pcategory:
req.body.data.data.pcategory,pname: req.body.data.data.pname,pno:
req.body.data.data.pno,pprice: req.body.data.data.pprice,pdiscount:
req.body.data.data.pdiscount,pdesc: req.body.data.data.pdesc}).then(result => {
        console.log(result._id);
        res.status(201).json({message: "Product updated successfully!!"});
      });
})

module.exports = router;
```

[2] Frontend Part

App.js

```
import { Fragment } from "react";
import { BrowserRouter as Router, Route } from "react-router-dom";
import "./App.css";
import Login from "../src/Components/auth/Login";

//Welcome Page
import welcomePage from "../src/Components/auth/welcomePage";

//Farmer View
import farmer from "./Components/Farmer View/farmer";
import record from "./Components/Farmer View/record";
import Success from "./Components/Farmer View/success";

//Customer VIew
import customer from "./Components/Customer View/customer/customer";
import plantNutritions from "./Components/Customer
View/customer/plantNutritions";
import tools from "./Components/Customer View/customer/tools";
import forageSeeds from "./Components/Customer View/customer/forageSeeds";
import horticultureSeeds from "./Components/Customer
View/customer/horticultureSeeds";
```

```
import cart from "./Components/Customer View/cart/cart";
import Payment from "./Components/Customer View/cart/payment";
import paymentSuccess from "./Components/Customer View/cart/paymentSuccess";
import recom from "./Components/Customer View/customer/recom";

//Authentication
import CustomerLogin from "./Components/auth/Login/CustomerLogin";
import FarmerLogin from "./Components/auth/Login/FarmerLogin";
import CustomerSignUp from "./Components/auth/SignUp/CustomerSignUp";
import FarmerSignup from "./Components/auth/SignUp/FarmerSignup";
import SignupAs from "./Components/auth/SignUp/SignupAs";

import React, { Component } from "react";

export default class App extends Component {
  render() {
    return (
      <div>
        <Router>
          <Fragment>
            <Route exact path="/" component={welcomePage} />
            <Route exact path="/Login" component={Login} />
            <Route exact path="/Login/Customer" component={CustomerLogin} />
            <Route exact path="/Login/Farmer" component={FarmerLogin} />
            <Route exact path="/SignUp/Customer" component={CustomerSignUp} />
            <Route exact path="/SignUp/Farmer" component={FarmerSignup} />
            <Route exact path="/SignUpAs" component={SignupAs} />
            <section className="container"></section>
            <Route exact path="/customer/dashboard" component={customer} />
            <Route exact path="/customer/recom" component={recom} />
            <Route
              exact
              path="/customer/dashboard/plantNutritions"
              component={plantNutritions}
            />
            <Route
              exact
              path="/customer/dashboard/tools"
              component={tools}
            />
            <Route exact path="/customer/dashboard/forageSeeds"
component={forageSeeds} />
            <Route
              exact
              path="/customer/dashboard/horticultureSeeds"
```

```
              component={horticultureSeeds}
            />
            <Route exact path="/customer/cart" component={cart} />
            <Route exact path="/payment" component={Payment} />
            <Route exact path="/paymentSuccess" component={paymentSuccess} />

            <Route exact path="/farmer/dashboard" component={farmer} />
            <Route exact path="/farmer/record" component={record} />
            <Route exact path="/farmer/Success" component={Success} />
          </Fragment>
        </Router>
      </div>
    );
  }
}
```

CustomerSignUp.js

```
import React, { Component, Fragment } from "react";
import { withRouter } from "react-router-dom";
import Navbar from "../../layout/navbar";
import Axios from 'axios';
import '../Login/Login.css';
class CustomerSignUp extends Component {
  constructor(props) {
    super(props);
    this.state = {
      Email: "",
      Name: "",
      Phone: "",
      Password: "",
      UserType: "Customer",
    };
  }

  EmailHandler = (event) => {
    this.setState({ Email: event.target.value });
  };
  NameHandler = (event) => {
    this.setState({ Name: event.target.value });
  };
  PhoneHandler = (event) => {
    this.setState({ Phone: event.target.value });
```

```
  };
  PassHandler = (event) => {
    this.setState({ Password: event.target.value });
  };

  SignEvent = (e) => {
    e.preventDefault();
    console.log(this.state);
    if(this.state.Email.length>0 && this.state.Name.length>0 &&
this.state.Password && this.state.Phone.length>0 && this.state.UserType.length>0)
{
      Axios.post("http://localhost:5000/auth/customer/signup", {email:
this.state.Email, name: this.state.Name, password: this.state.Password, phone:
this.state.Phone, type: this.state.UserType})
        .then(res => {
          console.log(res.data.message);
          this.props.history.push('/Login/Customer');
        })
        .catch(err => {
          console.log("User id already exists!!");
        })
    } else {
      console.log("Enter all fields correctly!!");
    }
  };
  render() {
    return (
      <Fragment>
        <Navbar type="signupAs" />
        <div class="wrapper">
          <form class="form-signin">
            <h2 class="form-signin-heading">Join As Customer</h2>
            <input type="text" class="form-control" name="name"
placeholder="Name" required="true" autofocus="" value={this.state.Name}
                   onChange={this.NameHandler} />
            <input type="text" class="form-control" name="phone"
placeholder="Phone" required="true" autofocus="" value={this.state.Phone}
                   onChange={this.PhoneHandler} />
            <input type="text" class="form-control" name="username"
placeholder="Email Address" required="true" autofocus="" value={this.state.Email}
                   onChange={this.EmailHandler} />
            <input type="password" class="form-control" name="password"
placeholder="Password" required="true"  value={this.state.Password}
                   onChange={this.PassHandler}/>
```

```
            <button class="btn btn-dark btn-sm" onClick={this.SignEvent}>Sign
Up</button>
          </form>
        </div>
      </Fragment>
    );
  }
}

export default withRouter(CustomerSignUp);
```

FarmerSignup.js

```
import React, { Component, Fragment } from "react";
import { withRouter } from "react-router-dom";
import Navbar from "../../layout/navbar";
import Axios from 'axios';
import '../Login/Login.css';
class FarmerSignUp extends Component {
  constructor(props) {
    super(props);
    this.state = {
      Email: "",
      Name: "",
      Phone: "",
      Password: "",
      UserType: "warehouse",
    };
  }

  EmailHandler = (event) => {
    this.setState({ Email: event.target.value });
  };
  NameHandler = (event) => {
    this.setState({ Name: event.target.value });
  };
  PhoneHandler = (event) => {
    this.setState({ Phone: event.target.value });
  };
  PassHandler = (event) => {
    this.setState({ Password: event.target.value });
  };
```

```
  SignEvent = (e) => {
    e.preventDefault();
    console.log(this.state);
    if(this.state.Email.length>0 && this.state.Name.length>0 &&
this.state.Password && this.state.Phone.length>0) {
      Axios.post("http://localhost:5000/auth/farmer/signup", {email:
this.state.Email, name: this.state.Name, password: this.state.Password, phone:
this.state.Phone, type: this.state.UserType})
        .then(res => {
          console.log(res.data.message);
          this.props.history.push('/Login/Farmer');
        })
        .catch(err => {
          console.log("User id already exists!!");
        })
    } else {
      console.log("Enter all fields correctly!!");
    }
  };
  render() {
    return (
      <Fragment>
        <Navbar type="signupAs" />
        <div class="wrapper">
          <form class="form-signin">
            <h2 class="form-signin-heading">Join As Farmer</h2>
            <input type="text" class="form-control" name="name"
placeholder="Name" required="true" autofocus="" value={this.state.Name}
                   onChange={this.NameHandler} />
            <input type="text" class="form-control" name="phone"
placeholder="Phone" required="true" autofocus="" value={this.state.Phone}
                   onChange={this.PhoneHandler} />
            <input type="text" class="form-control" name="username"
placeholder="Email Address" required="true" autofocus="" value={this.state.Email}
                   onChange={this.EmailHandler} />
            <input type="password" class="form-control" name="password"
placeholder="Password" required="true"  value={this.state.Password}
                   onChange={this.PassHandler}/>
            <button class="btn btn-dark btn-sm" onClick={this.SignEvent}>Sign
Up</button>
          </form>
        </div>
      </Fragment>
    );
  }
```

```
}

export default withRouter(FarmerSignUp);
```

CustomerLogin.js

```
import React, { Component, Fragment } from "react";
import { withRouter } from "react-router-dom";
import Navbar from '../../layout/navbar';
import Axios from 'axios';
import './Login.css';
class CustomerLogin extends Component {
  constructor(props) {
    super(props);
    this.state = {
      Email: "",
      Password: ""
    };
  }

  EmailHandler = (event) => {
    this.setState({ Email: event.target.value });
  };
  PassHandler = (event) => {
    this.setState({ Password: event.target.value });
  };

  LogEvent = (e) => {
    e.preventDefault();
    console.log(this.state);
    if(this.state.Email.length>0 && this.state.Password) {
      Axios.post("http://localhost:5000/auth/customer/signin", {email:
this.state.Email, password: this.state.Password})
        .then(res => {
          console.log(res.data.message, res.data.token, res.data.custid);
          localStorage.setItem('token', res.data.token);
          localStorage.setItem('id', res.data.custid);
          this.props.history.push('/customer/dashboard');
        })
        .catch(err => {
          console.log("User id or password is not valid!!");
        })
    } else {
      console.log("Enter all fields correctly!!");
    }
  }
```

```
  render() {
    return (
      <Fragment>
        <Navbar type='signupAs'/>
        <div class="wrapper">
          <form class="form-signin">
            <h2 class="form-signin-heading">Login</h2>
            <input type="text" class="form-control" name="username"
placeholder="Email Address" required="true" autofocus="" value={this.state.Email}
                   onChange={this.EmailHandler} />
            <input type="password" class="form-control" name="password"
placeholder="Password" required="true"  value={this.state.Password}
                   onChange={this.PassHandler}/>
            <button class="btn btn-dark btn-sm"
onClick={this.LogEvent}>Login</button>
          </form>
        </div>
      </Fragment>
    );
  }
}

export default withRouter(CustomerLogin);
```

FarmerLogin.js

```
import React, { Component, Fragment } from "react";
import { withRouter } from "react-router-dom";
import Navbar from '../../layout/navbar';
import Axios from 'axios';
import './Login.css';
class FarmerLogin extends Component {
  constructor(props) {
    super(props);
    this.state = {
      Email: "",
      Password: ""
    };
  }

  EmailHandler = (event) => {
    this.setState({ Email: event.target.value });
  };
```

```
  PassHandler = (event) => {
    this.setState({ Password: event.target.value });
  };

  LogEvent = (e) => {
    e.preventDefault();
    console.log(this.state);
    if(this.state.Email.length>0 && this.state.Password) {
      Axios.post("http://localhost:5000/auth/farmer/signin", {email:
this.state.Email, password: this.state.Password})
        .then(res => {
          console.log(res.data.message, res.data.token, res.data.custid);
          localStorage.setItem('token', res.data.token);
          localStorage.setItem('id', res.data.custid);
          this.props.history.push('/farmer/dashboard');
        })
        .catch(err => {
          console.log("User id or password is not valid!!");
        })
    } else {
      console.log("Enter all fields correctly!!");
    }
  }
  render() {
    return (
      <Fragment>
        <Navbar type='signupAs'/>
        <div class="wrapper">
          <form class="form-signin">
            <h2 class="form-signin-heading">Login</h2>
            <input type="text" class="form-control" name="username"
placeholder="Email Address" required="true" autofocus="" value={this.state.Email}
                   onChange={this.EmailHandler} />
            <input type="password" class="form-control" name="password"
placeholder="Password" required="true"  value={this.state.Password}
                   onChange={this.PassHandler}/>
            <button class="btn btn-dark btn-sm"
onClick={this.LogEvent}>Login</button>
          </form>
        </div>
      </Fragment>
    );
  }
}
```

```
export default withRouter(FarmerLogin);
```

Customer View

cart.js

```
import React, { Fragment } from 'react';
import {connect} from 'react-redux';
import { createStore } from 'redux';
import {withRouter} from 'react-router-dom';
import rootReducer from '../../../reducers/rootReducer';
import Axios from 'axios';
import Navbar from '../../layout/navbar';
import './cart.css';

class Cart extends React.Component {
    constructor(props){
        super(props);
        this.state = {
            productList: [],
            totalprice: 0
        };
        this.handleClick = this.handleClick.bind(this);
        this.buy = this.buy.bind(this);
        this.remove = this.remove.bind(this);
    }

    componentDidMount() {
        if(localStorage.getItem('token')) {
            if(this.props.productList.length > 0) {
                document.getElementById('topcart').hidden = false;
            } else {
                document.getElementById('topcart').hidden = true;
            }
            let tempList = this.props.productList;
            let totalprice = 0;
            tempList = tempList.map((obj) => {
                obj.tprice = Number(obj.ppricenew) * Number(obj.pno);
                totalprice += obj.tprice;
                return obj;
            })
            this.setState({
                productList: tempList,
```

```
                totalprice: totalprice
            });
        }
    }

    handleClick(obj) {
        console.log('Remove object', obj._id);
        this.props.removeFromCart(obj);
        const store = createStore(rootReducer);
        console.log('Central State', store.getState());
        if(store.getState().productList.length) {
            document.getElementById('topcart').hidden = false;
        } else {
            document.getElementById('topcart').hidden = true;
        }
        let tempList = store.getState().productList;
        let totalprice = 0;
        tempList = tempList.map((obj) => {
            obj.tprice = Number(obj.pprice) * Number(obj.pno);
            totalprice += obj.tprice;
            return obj;
        })
        this.setState({
            productList: tempList,
            totalprice: totalprice
        });
    }

    buy() {
        const edevice = [], food = [], clothes = [], furniture = [];
        this.state.productList.forEach(obj => {
            if(obj.ptype === 'Plant Nutritions') {
                edevice.push(obj);
            } else if(obj.ptype === 'Tools') {
                clothes.push(obj);
            } else if(obj.ptype === 'Horticulture Seeds') {
                furniture.push(obj);
            } else if(obj.ptype === 'Forage Seeds') {
                food.push(obj);
            }
        });
        console.log('Plant Nutritions', edevice);
        console.log('Forage Seeds', food);
        console.log('Horticulture Seeds', furniture);
        console.log('Tools', clothes);
```

```
        const custid = localStorage.getItem('id');
        Axios.post("http://localhost:5000/delivery/add", {data:
this.state.productList, custid: custid})
            .then((res) => {
                console.log(res.data.message);
                if(edevice) {
                    Axios.post("http://localhost:5000/customer/productList/plantN
utritions/update", edevice)
                        .then(res => {
                            console.log(res.data.message);
                        });
                }
                if(clothes) {
                    Axios.post("http://localhost:5000/customer/productList/tools/
update", clothes)
                        .then(res => {
                            console.log(res.data.message);
                        });
                }
                if(food) {
                    Axios.post("http://localhost:5000/customer/productList/forage
Seeds/update", food)
                        .then(res => {
                            console.log(res.data.message);
                        });
                }
                if(furniture) {
                    Axios.post("http://localhost:5000/customer/productList/Hortic
ultureSeeds/update", furniture)
                        .then(res => {
                            console.log(res.data.message);
                        });
                }
                this.remove();
                this.props.history.push("/payment");
            },
            (err) => {
                console.log(err);
            });
    }

    remove() {
        console.log('Cart reset!!');
        this.props.clearCart(this.state.productList);
        const store = createStore(rootReducer);
```

```
        console.log('Central State', store.getState());
        if(store.getState().productList.length) {
            document.getElementById('topcart').hidden = false;
        } else {
            document.getElementById('topcart').hidden = true;
        }
        this.setState({
            productList: store.getState().productList
        });
    }

    render() {
        let elem = (<div style={{textAlign: "center"}}><h1>You need to login
first...</h1></div>);
        if(localStorage.getItem('token')) {
            elem = (
                <span>
                    <div className="topcart m-2" id="topcart" hidden>
                        <p>Total price: {this.state.totalprice}</p>
                        <button type="button" id="buy" className="btn btn-dark m-
1" onClick={this.buy}>Buy</button>
                        <button type="button" id="reset" className="btn btn-dark
m-1" onClick={this.remove}>Clear Cart</button>
                    </div>
                    <div>
                    {
                        this.state.productList.map((obj) => {
                            return (
                                <div className="jumbotron" key={obj._id}>
                                    <div className="container">
                                        <div className="row">
                                            <div className="col-sm-4"><img
src={obj.pimg} alt={obj.pname}/></div>
                                            <div className="col-sm-8">
                                                <p className="lead">Product Name:
{obj.pname}</p>
                                                <p className="lead">Product
Price: {obj.ppricenew}</p>
                                                <p className="lead">Number of
items bought: {obj.pno}</p>
                                                <p className="lead">Price:</p>
                                                <input type="text" id={"price" +
obj._id} name="price" value={obj.tprice} readOnly/><br/><br/>
```

```
                                                <button type="button"
className="btn btn-danger" onClick={() =>
{this.handleClick(obj)}}>Remove</button>
                                            </div>
                                        </div>
                                    </div>
                                </div>
                            );
                        })
                    }
                     </div>
                </span>
            );
        }
        return(
            <Fragment>
                <Navbar type='customer'/>
                {elem}
            </Fragment>
        );
    }
}

const mapStateToProps = (state) => {
    return {
        productList: state.productList
    }
}

const mapDispatchToProps = (dispatch) => {
    return {
      removeFromCart: (obj) => dispatch({type: 'REMOVE_FROM_CART', object: obj}),
      clearCart: (data) => dispatch({type: 'CLEAR_CART', data: data})
    }
}

export default withRouter(connect(mapStateToProps, mapDispatchToProps)(Cart));
```

payment.js

```
import React, { Fragment } from "react";
import Navbar from '../../layout/navbar';
import './payment.css';
import { Link } from "react-router-dom";

function Payment() {
```

```
  return (
    <Fragment>
      <Navbar type='customer'/>
      <div class="container">
        <div class="row">
            <div class="col-lg-4 mb-lg-0 mb-3">
                <div class="card p-3">
                    <div class="img-box">
                        <img src="https://www.freepnglogos.com/uploads/visa-logo-
download-png-21.png" alt=""/>
                    </div>
                    <div class="number">
                        <label class="fw-bold" for="">**** **** **** 1060</label>
                    </div>
                    <div class="d-flex align-items-center justify-content-
between">
                        <small><span class="fw-bold">Expiry
date:</span><span>10/16</span></small>
                        <small><span class="fw-
bold">Name:</span><span>Kumar</span></small>
                    </div>
                </div>
            </div>
            <div class="col-lg-4 mb-lg-0 mb-3">
                <div class="card p-3">
                    <div class="img-box">
                        <img
src="https://www.freepnglogos.com/uploads/mastercard-png/file-mastercard-logo-
svg-wikimedia-commons-4.png"
                            alt=""/>
                    </div>
                    <div class="number">
                        <label class="fw-bold">**** **** **** 1060</label>
                    </div>
                    <div class="d-flex align-items-center justify-content-
between">
                        <small><span class="fw-bold">Expiry
date:</span><span>10/16</span></small>
                        <small><span class="fw-
bold">Name:</span><span>Kumar</span></small>
                    </div>
                </div>
            </div>
            <div class="col-lg-4 mb-lg-0 mb-3">
                <div class="card p-3">
```

```
                    <div class="img-box">
                        <img src="https://www.freepnglogos.com/uploads/discover-
png-logo/credit-cards-discover-png-logo-4.png"
                             alt=""/>
                    </div>
                    <div class="number">
                        <label class="fw-bold">**** **** **** 1060</label>
                    </div>
                    <div class="d-flex align-items-center justify-content-
between">
                        <small><span class="fw-bold">Expiry
date:</span><span>10/16</span></small>
                        <small><span class="fw-
bold">Name:</span><span>Kumar</span></small>
                    </div>
                </div>
            </div>
            <div class="col-12 mt-4">
                <div class="card p-3">
                    <p class="mb-0 fw-bold h4">Card Payment</p>
                </div>
            </div>
            <div class="col-12">
                <div class="card p-3">
                    <div class="card-body border p-0">
                        <p>
                            <a class="btn btn-primary p-2 w-100 h-100 d-flex
align-items-center justify-content-between"
                               data-bs-toggle="collapse" href="#collapseExample"
role="button" aria-expanded="true"
                               aria-controls="collapseExample">
                               <span class="fw-bold">Credit Card</span>
                               <span class="">
                                   <span class="fab fa-cc-amex"></span>
                                   <span class="fab fa-cc-mastercard"></span>
                                   <span class="fab fa-cc-discover"></span>
                               </span>
                            </a>
                        </p>
                        <div class="collapse show p-3 pt-0" id="collapseExample">
                            <div class="row">
                                <div class="col-lg-7">
                                    <form action="" class="form">
                                        <div class="row">
                                            <div class="col-12">
```

```
                                            <div class="form__div">
                                                <input type="text"
class="form-control" maxLength={16} placeholder=" "/>
                                                <label for=""
class="form__label">Card Number</label>
                                            </div>
                                        </div>

                                        <div class="col-6">
                                            <div class="form__div">
                                                <input type="text"
class="form-control" placeholder=" "/>
                                                <label for=""
class="form__label">MM / yy</label>
                                            </div>
                                        </div>


                                        <div class="col-6">
                                            <div class="form__div">
                                                <input type="password"
class="form-control" maxLength={3} placeholder=" "/>
                                                <label for=""
class="form__label">cvv code</label>
                                            </div>
                                        </div>
                                        <div class="col-12">
                                            <div class="form__div">
                                                <input type="text"
class="form-control" placeholder=" "/>
                                                <label for=""
class="form__label">name on the card</label>
                                            </div>
                                        </div>
                                        <Link to="/paymentSuccess">
                                        <div class="col-12">
                                            <div class="btn btn-primary w-
100">Pay</div>
                                        </div>
                                        </Link>
                                    </div>
                                </form>
                            </div>
                        </div>
                    </div>
```

```
                    </div>
                </div>
            </div>
        </div>
    </div>

    </Fragment>
  );
}

export default Payment;
```

customer.js

```
import React, { Fragment } from "react";
import { withRouter } from "react-router-dom";
import "./customer.css";
import Navbar from '../../layout/navbar';

const customer = (props) => {

  return (
    <Fragment>
        <Navbar type="customer" />

        <div className="container">
          <div className="row m-3">
            <div className="col-sm-6">
              <div
                className="card edevice"
                onClick={() => {
                  props.history.push("/customer/dashboard/plantNutritions");
                }}
              >
                <div className="card-body">
                  <h5 className="card-title">Plant Nutritions</h5>
                  <p className="card-text" style={{ color: "black" }}>
                  BUY AGRICULTURE PLANT NUTRITION PRODUCTS ONLINE
                  </p>
                </div>
              </div>
              {/* </Link> */}
            </div>
            <div className="col-sm-6">
              <div
```

```
            className="card food"
            onClick={() => {
              props.history.push("/customer/dashboard/forageSeeds");
            }}
          >
            <div className="card-body">
              <h5 className="card-title">Forage Seeds</h5>
              <p className="card-text" style={{ color: "black" }}>
              BUY HIGH NUTRITION FORAGE SEEDS ONLINE
              </p>
            </div>
          </div>
        </div>
        <div className="row m-3">
          <div className="col-sm-6">
            <div
              className="card furniture"
              onClick={() => {
                props.history.push("/customer/dashboard/horticultureSeeds");
              }}
            >
              <div className="card-body">
                <h5 className="card-title">Horticulture Seeds</h5>
                <p className="card-text" style={{ color: "black" }}>
                  BUY HYBRID VEGETABLE, FLOWER, AND FRUIT SEEDS ONLINE
                </p>
              </div>
            </div>
          </div>
          <div className="col-sm-6">
            <div
              className="card clothes"
              onClick={() => {
                props.history.push("/customer/dashboard/tools");
              }}
            >
              <div className="card-body">
                <h5 className="card-title">Tools & Instruments</h5>
                <p className="card-text" style={{ color: "black" }}>
                BUY AGRICULTURAL MACHINERY AND ACCESSORIES ONLINE
                </p>
              </div>
            </div>
          </div>
```

```
          </div>
        </div>
      </Fragment>
  );
}

export default withRouter(customer);
```

forageSeeds.js

```
import React, { Fragment } from 'react';
import ReactDOM from 'react-dom';
import Axios from 'axios';
import Product from './product';
import { createStore } from 'redux';
import rootReducer from '../../../reducers/rootReducer';
import Navbar from '../../layout/navbar';
import './searchbox.css';
import './sidebar.css';
import './product.css';
export default class forageSeeds extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            dataList: [],
            pnamelist: [],
            pdesclist: [],
            suggestions: [],
            modalobj: {}
        };
        this.filter = this.filter.bind(this);
        this.sortData = this.sortData.bind(this);
        this.handleChange = this.handleChange.bind(this);
        this.search = this.search.bind(this);
        this.handleKeyUp = this.handleKeyUp.bind(this);
        this.productmodal = this.productmodal.bind(this);
        this.handleNumberChange = this.handleNumberChange.bind(this);
        this.handleClick = this.handleClick.bind(this);
    }

    componentDidMount() {
        this.getData();
    }
```

```
    getData() {
        Axios.get("http://localhost:5000/customer/forageSeeds/data")
            .then((res) => {
                console.log('Data from backend', res.data);
                this.setState({
                    dataList: res.data.data
                });
                this.sortData();
                this.displaydata();
            })
    }

    sortData() {
        const temp = this.state.dataList;
        const tempname = [];
        const tempdesc = [];
        temp.forEach(obj => {
            tempname.push(obj.pname);
            tempdesc.push(obj.pdesc);
        });
        this.setState({
            pnamelist: tempname,
            pdesclist: tempdesc
        });
    }

    displaydata() {
        const n = this.state.dataList.length;
        let count = 0;
        const rowElem = document.createElement('div');
        rowElem.setAttribute('class', 'row my-1');
        for(let i=0;i<n ;i++){
            const colElem = document.createElement('div');
            colElem.setAttribute('class', 'col-3');
            colElem.setAttribute('id', 'prod2'+count);
            count++;
            rowElem.appendChild(colElem);
            document.getElementById('container2').appendChild(rowElem);
            if(count === n) break;
        }
        for(let i=0;i<n;i++) {
            ReactDOM.render(<Product dataObj={this.state.dataList[i]} id={i}
modalview={this.productmodal}/>, document.getElementById('prod2'+i));
        }
```

```
    }

    handleSideBar() {
        document.getElementById('foodsidebar').style.display = 'block';
    }

    closeSideBar() {
        document.getElementById('foodsidebar').style.display = 'none';
    }

    filter() {
        const pcat = document.getElementById('pcat2').value;
        const radioasc = document.getElementById('asc2');
        const radiodesc = document.getElementById('desc2');
        let order = '';
        if(radioasc.checked) {
            order = 'asc';
        } else if (radiodesc.checked) {
            order = 'desc';
        } else {
            order = '';
        }
        let lower = document.getElementById('lower2').value;
        let higher = document.getElementById('higher2').value;
        let temp = [];
        if(!lower) {
            lower = 0;
        }
        console.log(lower, higher);
        if(!higher) {
            this.state.dataList.forEach(obj => {
                console.log((Number(obj.ppricenew)));
                if(Number(obj.ppricenew) >= lower && pcat === obj.pcategory &&
pcat === obj.pcategory) {
                    temp.push(obj);
                    console.log(temp);
                }
            });
        } else {
            this.state.dataList.forEach(obj => {
                console.log((Number(obj.ppricenew)));
                if((Number(obj.ppricenew) >= lower)&&(Number(obj.ppricenew) <=
higher) && pcat === obj.pcategory) {
                    temp.push(obj);
                    console.log(temp);
```

```
                }
            });
        }
        console.log(order);
        if(order === 'asc') {
            temp.sort((a,b) => (a.ppricenew > b.ppricenew) ? 1 : ((a.ppricenew < 
b.ppricenew) ? -1 : 0));
        } else if(order === 'desc') {
            temp.sort((a,b) => (a.ppricenew < b.ppricenew) ? 1 : ((a.ppricenew > 
b.ppricenew) ? -1 : 0));
        } else {

        }
        console.log(temp);
        this.closeSideBar();
        this.displayfiltereddata(temp);
    }

    displayfiltereddata(data) {
        document.getElementById('container2').innerHTML = '';
        const n = data.length;
        let count = 0;
        const rowElem = document.createElement('div');
        rowElem.setAttribute('class', 'row my-1');
        for(let i=0;i<n ;i++){
            const colElem = document.createElement('div');
            colElem.setAttribute('class', 'col-3');
            colElem.setAttribute('id', 'filtprod2'+count);
            count++;
            rowElem.appendChild(colElem);
            document.getElementById('container2').appendChild(rowElem);
            if(count === n) break;
        }
        for(let i=0;i<n;i++) {
            ReactDOM.render(<Product dataObj={data[i]} id={i} 
modalview={this.productmodal}/>, document.getElementById('filtprod2'+i));
        }
    }

    productmodal(obj) {
        const modal = document.getElementById("foodproductModal");
        modal.style.display = "block";
        this.setState({
            modalobj: obj
        });
```

```
    }

    handleClose = () => {
        const modal = document.getElementById("foodproductModal");
        modal.style.display = "none";
    }

    handleKeyUp(e) {
        if (e.keyCode === 13) {
            const valtosearch = this.state.suggestions[0];
            this.search(valtosearch);
        }
    }

    handleChange(e) {
        const tempname = [...this.state.pnamelist];
        console.log(tempname);
        console.log(e.keyCode,e.target.value);
        const inputval = e.target.value;
        if(e.target.value.length === 0) {
            this.setState({
                suggestions: []
            });
            document.getElementById('container2').innerHTML = '';
            this.displaydata();
            return;
        }
        let suggestedvalues = [];
        const tempdesc = this.state.pdesclist;
        let key,value;
        for(key in tempname) {
            if(suggestedvalues.length <=5) {
                value = tempname[key];
                console.log(value, value.indexOf(inputval));
                if(value.toLowerCase().indexOf(inputval) > -1) {
                    suggestedvalues.push(value);
                }
            }
        }
        for(key in tempdesc) {
            if(suggestedvalues.length <=5) {
                value = tempdesc[key];
                if(value.toLowerCase().indexOf(inputval) > -1) {
                    suggestedvalues.push(value);
                }
```

```
            }
        }
        this.setState({
            suggestions: [...suggestedvalues]
        });
        console.log(suggestedvalues);
    }

    search(name) {
        console.log(name);
        const templist = [];
        this.state.dataList.forEach(obj => {
            if(obj.pname === name) {
                templist.push(obj);
            } else if(obj.pdesc === name){
                templist.push(obj);
            }
        });
        this.displayfiltereddata(templist);
        this.setState({
            suggestions: []
        });
    }

    handleNumberChange(e) {
        console.log(e.target.value);
        this.state.modalobj.pno = e.target.value;
    }

    handleClick() {
        if(document.getElementById('foodquantity' +
this.state.modalobj._id).value) {
            console.log(document.getElementById('foodquantity' +
this.state.modalobj._id).value);
            const store = createStore(rootReducer);
            store.dispatch({type: 'ADD_TO_CART', object: this.state.modalobj});
            document.getElementById('foodreset' +
this.state.modalobj._id).click();
        } else {
            console.log("Enter the number of products you want to buy!!");
        }
    }

    render() {
        return(
```

```
            <Fragment>
                <Navbar type='customer'/>
                <div className="search my-0">
                    <input type="text" className="searchTerm" placeholder="What
are you looking for?" onChange={this.handleChange} onKeyUp={this.handleKeyUp}
id={"foodsearch" + this.state.modalobj._id}/>
                    <button type="button" className="searchButton"
onClick={this.handleSideBar}>
                        Filter
                    </button>
                </div>
                <ul className="searchcontainer">
                {
                    this.state.suggestions.map(elem => {
                        return (
                            <li key={elem} >
                                <button type="button" className="searchsuggest"
                                onClick={() => {
                                    this.search(elem);
                                }}>{elem}</button>
                            </li>
                        )
                    })
                }
                </ul>
                <span id="foodsidebar"  className="sidebar">
                    <div>
                    <i className="fa fa-close" onClick={this.closeSideBar}></i>
                    </div><br/>
                    <div  className="sidebarcontent">
                        <h3>Forage Seeds</h3>
                        <label>Product Brand:</label><br/>
                        <select id="pcat2">
                            <option value="Hayleys">Hayleys</option>
                            <option value="Cargills">Cargills</option>
                            <option value="CIC Holdings">CIC Holdings</option>
                            <option value="Green Farms Pvt">Green Farms
Pvt</option>
                            <option value="Non-Branded">Non-Branded</option>
                        </select><br/>
                        <label>Price range:</label><br/>
                        <input type="number" id="lower2" min="0"/> -&gt;
Lower<br/><br/>
                        <input type="number" id="higher2" min="0"/> -&gt;
Higher<br/><br/>
```

```
                        <button type="button" className="btn btn-light"
onClick={this.filter}>Search</button>
                    </div>
                </span>
                <div id="foodproductModal" className="modal">
                    <div className="modal-content">
                        <div className="closecontainer">
                            <div
className="heading"><h5>{this.state.modalobj.pname} </h5></div>
                            <div id="close" className="close"
onClick={this.handleClose}>&times;</div>
                        </div>
                        <form>
                            <div className="rowcontainer">
                                <div className="side">
                                    <img className=""
src={this.state.modalobj.pimg} alt={this.state.modalobj.pname}/>
                                </div>
                                <div className="main">
                                    <div className="m-4 modalbody">
                                        <label className="discprice px-
1">Rs.{this.state.modalobj.ppricenew} </label>
                                        <label className="discount px-
1">{this.state.modalobj.pdiscount}% </label>
                                        <label
className="price">{this.state.modalobj.pprice} </label><br/>
                                        <label>Available:
{this.state.modalobj.pno}</label><br/>
                                        <label>Product description:</label><br/>
                                        <p>{this.state.modalobj.pdesc}</p>
                                        <label>Number:</label><br/>
                                        <input type="number" id={"foodquantity" +
this.state.modalobj._id} name="quantity" min="1" max={this.state.modalobj.pno}
onChange={this.handleNumberChange}/><br/>
                                        <button type="button" className="btn btn-
dark my-1" id={"foodcart" + this.state.modalobj._id}
onClick={this.handleClick}>Add to cart</button>
                                        <button type="reset" id={"foodreset" +
this.state.modalobj._id} hidden>Reset</button>
                                    </div>
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
```

```
                <span id="container2" className="container"></span>
            </Fragment>
        );  }}
```

horticulture.js

```
import React, { Fragment } from 'react';
import ReactDOM from 'react-dom';
import Axios from 'axios';
import Product from './product';
import { createStore } from 'redux';
import rootReducer from '../../../reducers/rootReducer';
import Navbar from '../../layout/navbar';
import './searchbox.css';
import './sidebar.css';
import './product.css';

export default class horticultureSeeds extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            dataList: [],
            pnamelist: [],
            pdesclist: [],
            suggestions: [],
            modalobj: {}
        };
        this.filter = this.filter.bind(this);
        this.sortData = this.sortData.bind(this);
        this.handleChange = this.handleChange.bind(this);
        this.search = this.search.bind(this);
        this.handleKeyUp = this.handleKeyUp.bind(this);
        this.productmodal = this.productmodal.bind(this);
        this.handleNumberChange = this.handleNumberChange.bind(this);
        this.handleClick = this.handleClick.bind(this);
    }

    componentDidMount() {
        this.getData();
    }

    getData() {
        Axios.get("http://localhost:5000/customer/horticultureSeeds/data")
            .then((res) => {
                console.log('Data from backend', res.data);
                this.setState({
```

```
                    dataList: res.data.data
                });
                this.sortData();
                this.displaydata();
            })
    }

    sortData() {
        const temp = this.state.dataList;
        const tempname = [];
        const tempdesc = [];
        temp.forEach(obj => {
            tempname.push(obj.pname);
            tempdesc.push(obj.pdesc);
        });
        this.setState({
            pnamelist: tempname,
            pdesclist: tempdesc
        });
    }

    displaydata() {
        const n = this.state.dataList.length;
        let count = 0;
        const rowElem = document.createElement('div');
        rowElem.setAttribute('class', 'row my-1');
        for(let i=0;i<n ;i++){
            const colElem = document.createElement('div');
            colElem.setAttribute('class', 'col-3');
            colElem.setAttribute('id', 'prod3'+count);
            count++;
            rowElem.appendChild(colElem);
            document.getElementById('container3').appendChild(rowElem);
            if(count === n) break;
        }
        for(let i=0;i<n;i++) {
            ReactDOM.render(<Product dataObj={this.state.dataList[i]} id={i}
modalview={this.productmodal}/>, document.getElementById('prod3'+i));
        }
    }

    handleSideBar() {
        document.getElementById('furnituresidebar').style.display = 'block';
    }
```

```
    closeSideBar() {
        document.getElementById('furnituresidebar').style.display = 'none';
    }

    filter() {
        const pcat = document.getElementById('pcat3').value;
        const radioasc = document.getElementById('asc2');
        const radiodesc = document.getElementById('desc2');
        let order = '';
        if(radioasc.checked) {
            order = 'asc';
        } else if (radiodesc.checked) {
            order = 'desc';
        } else {
            order = '';
        }
        let lower = document.getElementById('lower3').value;
        let higher = document.getElementById('higher3').value;
        let temp = [];
        if(!lower) {
            lower = 0;
        }
        console.log(lower, higher);
        if(!higher) {
            this.state.dataList.forEach(obj => {
                console.log((Number(obj.ppricenew)));
                if(Number(obj.ppricenew) >= lower && pcat === obj.pcategory) {
                    temp.push(obj);
                    console.log(temp);
                }
            });
        } else {
            this.state.dataList.forEach(obj => {
                console.log((Number(obj.ppricenew)));
                if((Number(obj.ppricenew) >= lower)&&(Number(obj.ppricenew) <=
higher) && pcat === obj.pcategory) {
                    temp.push(obj);
                    console.log(temp);
                }
            });
        }
        console.log(order);
        if(order === 'asc') {
            temp.sort((a,b) => (a.ppricenew > b.ppricenew) ? 1 : ((a.ppricenew <
b.ppricenew) ? -1 : 0));
```

```
        } else if(order === 'desc') {
            temp.sort((a,b) => (a.ppricenew < b.ppricenew) ? 1 : ((a.ppricenew >
b.ppricenew) ? -1 : 0));
        } else {

        }
        console.log(temp);
        this.closeSideBar();
        this.displayfiltereddata(temp);
    }

    displayfiltereddata(data) {
        document.getElementById('container3').innerHTML = '';
        const n = data.length;
        let count = 0;
        const rowElem = document.createElement('div');
        rowElem.setAttribute('class', 'row my-1');
        for(let i=0;i<n ;i++){
            const colElem = document.createElement('div');
            colElem.setAttribute('class', 'col-3');
            colElem.setAttribute('id', 'filtprod3'+count);
            count++;
            rowElem.appendChild(colElem);
            document.getElementById('container3').appendChild(rowElem);
            if(count === n) break;
        }
        for(let i=0;i<n;i++) {
            ReactDOM.render(<Product dataObj={data[i]} id={i}
modalview={this.productmodal}/>, document.getElementById('filtprod3'+i));
        }
    }

    productmodal(obj) {
        const modal = document.getElementById("furnitureproductModal");
        modal.style.display = "block";
        this.setState({
            modalobj: obj
        });
    }

    handleClose = () => {
        const modal = document.getElementById("furnitureproductModal");
        modal.style.display = "none";
    }
```

```
    handleKeyUp(e) {
        if (e.keyCode === 13) {
            const valtosearch = this.state.suggestions[0];
            this.search(valtosearch);
        }
    }

    handleChange(e) {
        const tempname = [...this.state.pnamelist];
        console.log(tempname);
        console.log(e.keyCode,e.target.value);
        const inputval = e.target.value;
        if(e.target.value.length === 0) {
            this.setState({
                suggestions: []
            });
            document.getElementById('container3').innerHTML = '';
            this.displaydata();
            return;
        }
        let suggestedvalues = [];
        const tempdesc = this.state.pdesclist;
        let key,value;
        for(key in tempname) {
            if(suggestedvalues.length <=5) {
                value = tempname[key];
                console.log(value, value.indexOf(inputval));
                if(value.toLowerCase().indexOf(inputval) > -1) {
                    suggestedvalues.push(value);
                }
            }
        }
        if(suggestedvalues.length <=5) {
            value = tempdesc[key];
            if(value.toLowerCase().indexOf(inputval) > -1) {
                suggestedvalues.push(value);
            }
        }
        this.setState({
            suggestions: [...suggestedvalues]
        });
        console.log(suggestedvalues);
    }

    search(name) {
```

```
        console.log(name);
        const templist = [];
        this.state.dataList.forEach(obj => {
            if(obj.pname === name) {
                templist.push(obj);
            } else if(obj.pdesc === name){
                templist.push(obj);
            }
        });
        this.displayfiltereddata(templist);
        this.setState({
            suggestions: []
        });
    }

    handleNumberChange(e) {
        console.log(e.target.value);

        this.state.modalobj.pno = e.target.value;
    }

    handleClick() {
        if(document.getElementById('furniturequantity' +
this.state.modalobj._id).value) {
            console.log(document.getElementById('furniturequantity' +
this.state.modalobj._id).value);
            const store = createStore(rootReducer);
            store.dispatch({type: 'ADD_TO_CART', object: this.state.modalobj});
            document.getElementById('furniturereset' +
this.state.modalobj._id).click();
        } else {
            console.log("Enter the number of products you want to buy!!");
        }
    }

    render() {
        return(
            <Fragment>
                <Navbar type='customer'/>
                <div className="search my-0">
                    <input type="text" className="searchTerm" placeholder="What
are you looking for?" onChange={this.handleChange} onKeyUp={this.handleKeyUp}
id={"furnituresearch" + this.state.modalobj._id}/>
                    <button type="button" className="searchButton"
onClick={this.handleSideBar}>
```

```
                        Filter
                    </button>
                </div>
                <ul className="searchcontainer">
                {
                    this.state.suggestions.map(elem => {
                        return (
                            <li key={elem} >
                                <button type="button" className="searchsuggest"
                                onClick={() => {
                                    this.search(elem);
                                }}>{elem}</button>
                            </li>
                        )
                    })
                }
                </ul>
                <span id="furnituresidebar"  className="sidebar">
                    <div>
                    <i className="fa fa-close" onClick={this.closeSideBar}></i>
                    </div><br/>
                    <div className="sidebarcontent">
                        <h3>Horticulture Seeds</h3>
                        <label>Product Brand:</label><br/>
                        <select id="pcat3">
                            <option value="Hayleys">Hayleys</option>
                            <option value="Cargills">Cargills</option>
                            <option value="CIC Holdings">CIC Holdings</option>
                            <option value="Green Farms Pvt">Green Farms
Pvt</option>
                            <option value="Non-Branded">Non-Branded</option>
                        </select><br/>
                        <label>Price range:</label><br/>
                        <input type="number" id="lower3" min="0"/> -&gt;
Lower<br/><br/>
                        <input type="number" id="higher3" min="0"/> -&gt;
Higher<br/><br/>
                        <button type="button" className="btn btn-light"
onClick={this.filter}>Search</button>
                    </div>
                </span>
                <div id="furnitureproductModal" className="modal">
                    <div className="modal-content">
                        <div className="closecontainer">
```

```
                            <div
className="heading"><h5>{this.state.modalobj.pname} </h5></div>
                            <div id="close" className="close"
onClick={this.handleClose}>&times;</div>
                        </div>
                        <form>
                            <div className="rowcontainer">
                                <div className="side">
                                    <img className=""
src={this.state.modalobj.pimg} alt={this.state.modalobj.pname}/>
                                </div>
                                <div className="main">
                                    <div className="m-4 modalbody">
                                        <label className="discprice px-
1">Rs.{this.state.modalobj.ppricenew} </label>
                                        <label className="discount px-
1">{this.state.modalobj.pdiscount}% </label>
                                        <label
className="price">{this.state.modalobj.pprice} </label><br/>
                                        <label>Available:
{this.state.modalobj.pno}</label><br/>
                                        <label>Product description:</label><br/>
                                        <p>{this.state.modalobj.pdesc}</p>
                                        <label>Number:</label><br/>
                                        <input type="number"
id={"furniturequantity" + this.state.modalobj._id} name="quantity" min="1"
max={this.state.modalobj.pno} onChange={this.handleNumberChange}/><br/>
                                        <button type="button" className="btn btn-
dark my-1" id={"furniturecart" + this.state.modalobj._id}
onClick={this.handleClick}>Add to cart</button>
                                        <button type="reset" id={"furniturereset"
+ this.state.modalobj._id} hidden>Reset</button>
                                    </div>
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
                <span id="container3" className="container"></span>
            </Fragment>
        );
    }
}
```

plantNutritions.js

```
import React, { Fragment } from 'react';
import ReactDOM from 'react-dom';
import { withRouter } from 'react-router-dom';
import Axios from 'axios';
import Product from './product';
import Navbar from '../../layout/navbar';
import './searchbox.css';
import './sidebar.css';
import './product.css';
import { createStore } from 'redux';
import rootReducer from '../../../reducers/rootReducer';
class plantNutritions extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            dataList: [],
            pnamelist: [],
            pdesclist: [],
            suggestions: [],
            modalobj: {}
        };
        this.filter = this.filter.bind(this);
        this.sortData = this.sortData.bind(this);
        this.handleChange = this.handleChange.bind(this);
        this.search = this.search.bind(this);
        this.handleKeyUp = this.handleKeyUp.bind(this);
        this.productmodal = this.productmodal.bind(this);
        this.handleNumberChange = this.handleNumberChange.bind(this);
        this.handleClick = this.handleClick.bind(this);
    }

    componentDidMount() {
        this.getData();
    }

    getData() {
        Axios.get("http://localhost:5000/customer/plantNutritions/data")
            .then((res) => {
                console.log('Data from backend', res.data);
                this.setState({
                    dataList: res.data.data
                });
                this.sortData();
                this.displaydata();
            })
```

```
    }

    sortData() {
        const temp = this.state.dataList;
        const tempname = [];
        const tempdesc = [];
        temp.forEach(obj => {
            tempname.push(obj.pname);
            tempdesc.push(obj.pdesc);
        });
        this.setState({
            pnamelist: tempname,
            pdesclist: tempdesc
        });
    }

    displaydata() {
        const n = this.state.dataList.length;
        let count = 0;
        const rowElem = document.createElement('div');
        rowElem.setAttribute('class', 'row my-1');
        for(let i=0;i<n ;i++){
            const colElem = document.createElement('div');
            colElem.setAttribute('class', 'col-3');
            colElem.setAttribute('id', 'prod'+count);
            count++;
            rowElem.appendChild(colElem);
            document.getElementById('container').appendChild(rowElem);
            if(count === n) break;
        }
        for(let i=0;i<n;i++) {
            ReactDOM.render(<Product dataObj={this.state.dataList[i]} id={i}
modalview={this.productmodal}/>, document.getElementById('prod'+i));
        }
    }

    handleSideBar() {
        document.getElementById('edevicesidebar').style.display = 'block';
    }

    closeSideBar() {
        document.getElementById('edevicesidebar').style.display = 'none';
    }

    filter() {
```

```
        const pcat = document.getElementById('pcat').value;
        const radioasc = document.getElementById('asc');
        const radiodesc = document.getElementById('desc');
        let order = '';
        if(radioasc.checked) {
            order = 'asc';
        } else if (radiodesc.checked) {
            order = 'desc';
        } else {
            order = '';
        }
        let lower = document.getElementById('lower').value;
        let higher = document.getElementById('higher').value;
        let temp = [];
        if(!lower) {
            lower = 0;
        }
        console.log(lower, higher);
        if(!higher) {
            this.state.dataList.forEach(obj => {
                console.log((Number(obj.ppricenew)));
                if(Number(obj.ppricenew) >= lower && pcat === obj.pcategory) {
                    temp.push(obj);
                    console.log(temp);
                }
            });
        } else {
            this.state.dataList.forEach(obj => {
                console.log((Number(obj.ppricenew)));
                if((Number(obj.ppricenew) >= lower)&&(Number(obj.ppricenew) <=
higher) && pcat === obj.pcategory) {
                    temp.push(obj);
                    console.log(temp);
                }
            });
        }
        console.log(order);
        if(order === 'asc') {
            temp.sort((a,b) => (a.ppricenew > b.ppricenew) ? 1 : ((a.ppricenew <
b.ppricenew) ? -1 : 0));
        } else if(order === 'desc') {
            temp.sort((a,b) => (a.ppricenew < b.ppricenew) ? 1 : ((a.ppricenew >
b.ppricenew) ? -1 : 0));
        } else {
```

```
        }
        console.log(temp);
        this.closeSideBar();
        this.displayfiltereddata(temp);
    }
    displayfiltereddata(data) {
        document.getElementById('container').innerHTML = '';
        const n = data.length;
        let count = 0;
        const rowElem = document.createElement('div');
        rowElem.setAttribute('class', 'row my-1');
        for(let i=0;i<n ;i++){
            const colElem = document.createElement('div');
            colElem.setAttribute('class', 'col-3');
            colElem.setAttribute('id', 'filtprod'+count);
            count++;
            rowElem.appendChild(colElem);
            document.getElementById('container').appendChild(rowElem);
            if(count === n) break;
        }
        for(let i=0;i<n;i++) {
            ReactDOM.render(<Product dataObj={data[i]} id={i}
modalview={this.productmodal}/>, document.getElementById('filtprod'+i));
        }
    }
    productmodal(obj) {
        const modal = document.getElementById("edeviceproductModal");
        modal.style.display = "block";
        this.setState({
            modalobj: obj
        });
    }
    handleClose = () => {
        const modal = document.getElementById("edeviceproductModal");
        modal.style.display = "none";
    }
    handleKeyUp(e) {
        if (e.keyCode === 13) {
            const valtosearch = this.state.suggestions[0];
            this.search(valtosearch);
        }
    }

    handleChange(e) {
        const tempname = [...this.state.pnamelist];
```

```
        console.log(tempname);
        console.log(e.keyCode,e.target.value);
        const inputval = e.target.value;
        if(e.target.value.length === 0) {
            this.setState({
                suggestions: []
            });
            document.getElementById('container').innerHTML = '';
            this.displaydata();
            return;
        }
        let suggestedvalues = [];
        const tempdesc = this.state.pdesclist;
        let key,value;
        for(key in tempname) {
            if(suggestedvalues.length <=5) {
                value = tempname[key];
                console.log(value, value.indexOf(inputval));
                if(value.toLowerCase().indexOf(inputval) > -1) {
                    suggestedvalues.push(value);
                }
            }
        }
        for(key in tempdesc) {
            if(suggestedvalues.length <=5) {
                value = tempdesc[key];
                if(value.toLowerCase().indexOf(inputval) > -1) {
                    suggestedvalues.push(value);
                }
            }
        }
        this.setState({
            suggestions: [...suggestedvalues]
        });
        console.log(suggestedvalues);
    }
    search(name) {
        console.log(name);
        const templist = [];
        this.state.dataList.forEach(obj => {
            if(obj.pname === name) {
                templist.push(obj);
            } else if(obj.pdesc === name){
                templist.push(obj);
            }
```

```
        });
        this.displayfiltereddata(templist);
        this.setState({
            suggestions: []
        });
    }
    handleNumberChange(e) {
        console.log(e.target.value);
        this.state.modalobj.pno = e.target.value;
    }
    handleClick() {
        if(document.getElementById('edevicequantity' +
this.state.modalobj._id).value) {
            console.log(document.getElementById('edevicequantity' +
this.state.modalobj._id).value);
            const store = createStore(rootReducer);
            store.dispatch({type: 'ADD_TO_CART', object: this.state.modalobj});
            document.getElementById('edevicereset' +
this.state.modalobj._id).click();
        } else {
            console.log("Enter the number of products you want to buy!!");
        }
    }
    render() {

        return(
            <Fragment>
                <Navbar type='customer'/>
                <div className="search my-0">
                    <input type="text" className="searchTerm" placeholder="What
are you looking for?" onChange={this.handleChange} onKeyUp={this.handleKeyUp}
id={"edevicesearch" + this.state.modalobj._id}/>
                    <button type="button" className="searchButton"
onClick={this.handleSideBar}>
                        Filter
                    </button>
                </div>
                <ul className="searchcontainer">
                {
                    this.state.suggestions.map(elem => {
                        return (
                            <li key={elem} >
                                <button type="button" className="searchsuggest"
                                onClick={() => {
                                    this.search(elem);
```

```
                            }}>{elem}</button>
                        </li>
                    )
                })
            }
            </ul>
            <span id="edevicesidebar"  className="sidebar">
                <div>
                <i className="fa fa-close" onClick={this.closeSideBar}></i>
                </div><br/>
                <div  className="sidebarcontent">
                    <h3>Plant Nutritions</h3>
                    <label>Product Brand:</label><br/>
                    <select id="pcat">
                        <option value="Hayleys">Hayleys</option>
                        <option value="Cargills">Cargills</option>
                        <option value="CIC Holdings">CIC Holdings</option>
                        <option value="Green Farms Pvt">Green Farms
Pvt</option>
                        <option value="Non-Branded">Non-Branded</option>
                    </select><br/>
                    <label>Price range:</label><br/>
                    <input type="number" id="lower" min="0"/> -&gt;
Lower<br/><br/>
                    <input type="number" id="higher" min="0"/> -&gt;
Higher<br/><br/>
                    <p>Sort by price:</p>
                    <input type="radio" id="asc" name="sort" value="asc"/>
                    <label htmlFor="asc">Ascending</label><br/>
                    <input type="radio" id="desc" name="sort" value="desc"/>
                    <label htmlFor="desc">Descending</label><br/>
                    <button type="button" className="btn btn-light"
onClick={this.filter}>Search</button>
                </div>
            </span>
            <div id="edeviceproductModal" className="modal">
                <div className="modal-content">
                    <div className="closecontainer">
                        <div
className="heading"><h5>{this.state.modalobj.pname} </h5></div>
                        <div id="close" className="close"
onClick={this.handleClose}>&times;</div>
                    </div>
                    <form>
                        <div className="rowcontainer">
```

```
                                <div className="side">
                                    <img className=""
src={this.state.modalobj.pimg} alt={this.state.modalobj.pname}/>
                                </div>
                                <div className="main">
                                    <div className="m-4 modalbody">
                                        <label className="discprice px-
1">Rs.{this.state.modalobj.ppricenew} </label>
                                        <label className="discount px-
1">{this.state.modalobj.pdiscount}% </label>
                                        <label
className="price">{this.state.modalobj.pprice} </label><br/>
                                        <label>Available:
{this.state.modalobj.pno}</label><br/>
                                        <label>Product description:</label><br/>
                                        <p>{this.state.modalobj.pdesc}</p>
                                        <label>Number:</label><br/>
                                        <input type="number"
className="numberofprod" id={"edevicequantity" + this.state.modalobj._id}
name="edevicequantity" min="1" max={this.state.modalobj.pno}
onChange={this.handleNumberChange}/><br/>
                                        <button type="button" id={"edevicecart" +
this.state.modalobj._id} className="btn btn-dark my-1"
onClick={this.handleClick}>Add to cart</button>
                                        <button type="reset" id={"edevicereset" +
this.state.modalobj._id} hidden>Reset</button>
                                    </div>
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
                <span id="container" className="container"></span>
            </Fragment>
        );
    }
}

export default withRouter(plantNutritions);
```

Farmer View

Farmer.js

```
import React, { Fragment } from 'react';
import axios from 'axios';
```

```
import { MDBContainer, MDBRow, MDBCol, MDBInput, MDBCard, MDBCardBody,
MDBCardHeader } from 'mdbreact';
import Navbar from '../layout/navbar';
import './farmer.css';
import {storage} from '../firebase/index';

export default class farmer extends React.Component {
    constructor(props){
        super(props);
        this.state = {
            productList: [],
            rowDetails: {},
            options: ["Hayleys", "Cargills", "CIC Holdings", "Green Farms Pvt",
"Non-Branded"]
        };
        this.handleChange = this.handleChange.bind(this);
    }

    handleChange() {
        const nptype = document.getElementById('ptype').value;
        if(nptype === 'Plant Nutritions') {
            const options = ["Hayleys", "Cargills", "CIC Holdings", "Green Farms
Pvt", "Non-Branded"];
            this.setState({
                options: options
            });
        } else if(nptype === 'Tools') {
            const options = ["Hayleys", "Cargills", "CIC Holdings", "Green Farms
Pvt", "Non-Branded"];
            this.setState({
                options: options
            });
        } else if(nptype === 'Horticulture Seeds') {
            const options = ["Hayleys", "Cargills", "CIC Holdings", "Green Farms
Pvt", "Non-Branded"];
            this.setState({
                options: options
            });
        } else if(nptype === 'Forage Seeds') {
            const options = ["Hayleys", "Cargills", "CIC Holdings", "Green Farms
Pvt", "Non-Branded"];
            this.setState({
                options: options
            });
        }
```

```
    }

    addNewProduct = e => {
        e.preventDefault();
        console.log("Add button clicked!!");
        const ptype = document.getElementById('ptype').value;
        const pcategory = document.getElementById('pcategory').value;
        const pname = document.getElementById('pname').value;
        let pimg = document.getElementById('pimg').files[0];
        const pno = document.getElementById('pno').value;
        const pprice = document.getElementById('pprice').value;
        const pdiscount = document.getElementById('pdiscount').value;
        const pdesc = document.getElementById('pdesc').value;
        if(pname&&pimg&&pno&&pprice&&pdesc) {
            console.log("New product added!!");
            //send to database
            const uploadTask = storage.ref(`images/${pimg.name}`).put(pimg);
            uploadTask.on('state_changed',
            (snapshot) => {},
            (error) => {
                console.log('Firebase image upload error', error);
                this.notifyB('Error');
            },
            () => {
                storage.ref('images').child(pimg.name).getDownloadURL()
                    .then(url => {
                        console.log('URL', url);
                        const productData = {pimg:
url,ptype,pcategory,pname,pno,pprice,pdiscount,pdesc};
                        axios.post("http://localhost:5000/farmer/productList/add"
, productData)
                            .then(res => {
                                this.props.history.push("/farmer/record");
                                console.log(res.data.message);
                                document.getElementById('reset').click();
                            });
                    })
            });
        } else {
            console.log("Form empty!!");
        }
    }

    render() {
```

```
        let elem = (<div style={{textAlign: "center"}}><h1>You need to login
first...</h1></div>);
        if(localStorage.getItem('token') === 'warehouse') {
            elem = (
                <MDBContainer>
                    <div  className="my-5">
                    <MDBRow>
                        <MDBCol md="12">
                        <MDBCard>
                            <MDBCardHeader style={{backgroundColor: "black",
color: "white"}}>
                                <p className="h4 text-center py-4">Add New
Product:</p>
                            </MDBCardHeader>
                            <MDBCardBody>
                            <div className="newProduct">
                                <form>
                                    <div className="black-text">
                                        <label>Product Category:</label>
                                        <select id="ptype" className="browser-
default custom-select" onChange={this.handleChange}>
                                            <option value="Plant Nutritions"
defaultValue>Plant Nutritions</option>
                                            <option value="Tools">Tools</option>
                                            <option value="Forage Seeds">Forage
Seeds</option>
                                            <option value="Horticulture
Seeds">Horticulture Seeds</option>
                                        </select><br/><br/>
                                        <label>Product Brand:</label>
                                        <select id="pcategory"
className="browser-default custom-select">
                                        {
                                            this.state.options.map(elem => {
                                                return (
                                                    <option value={elem}
key={elem}>{elem}</option>
                                                )
                                            })
                                        }
                                        </select>
                                        <MDBInput
                                            label="Product name:"
                                            group
                                            type="text"
```

```
                    id="pname"
                    name="pname"
                />
                <label>
                Product Image:
                </label>
                <div>
                    <input
                        type="file"
                        id="pimg"
                        name="pimg"
                        accept="image/*"
                        />
                </div>
                <MDBInput
                    label="Product quantity:"
                    group
                    type="number"
                    id="pno"
                    name="pno"
                    min="1"
                />
                <MDBInput
                    label="Product price:"
                    group
                    type="number"
                    id="pprice"
                    name="pprice"
                    min="1"
                />
                <MDBInput
                    label="Product discount:"
                    group
                    type="number"
                    id="pdiscount"
                    name="pdiscount"
                    min="1"
                /><br/>
                <label>
                Product description:
                </label>
                <textarea
                className="form-control"
                id="pdesc"
                rows="5"
```

```
                                    />
                                </div>
                                <div className="text-center py-4 mt-3">
                                    <button type="button" className="btn btn-
dark mx-1" onClick={this.addNewProduct}>Add</button>
                                    <button type="reset" id="reset"
className="btn btn-dark mx-1" onClick={() => {
                                        window.location.reload();
                                    }}>Reset</button>
                                </div>
                            </form>
                        </div>
                        </MDBCardBody>
                    </MDBCard>
                    </MDBCol>
                </MDBRow>
                </div>
            </MDBContainer>
        );
    }
    return(
        <Fragment>
            <Navbar type='warehouse'/>
            {elem}
        </Fragment>
    );
  }

}
```

ModalComponents.js

```
import React, { Fragment } from 'react';
import axios from 'axios';
import './ModalComponent.css';
import {storage} from '../firebase/index';
export default class Table extends React.Component {
    constructor(props){
        super(props);
        this.state = {
            show: false
        };
    }
```

```
    handleClose = () => {
        const modal = document.getElementById("myModal");
        modal.style.display = "none";
    }
    handleShow = () => {
        const modal = document.getElementById("myModal");
        modal.style.display = "block";
    }

    updateProduct = e => {
        e.preventDefault();
        console.log("Update button clicked!!");
        const ptype = document.getElementById('mptype').value;
        const pcategory = document.getElementById('mpcategory').value;
        const pname = document.getElementById('mpname').value;
        let pimg = document.getElementById('mpimg').files[0];
        const pno = document.getElementById('mpno').value;
        const pprice = document.getElementById('mpprice').value;
        const pdiscount = document.getElementById('mpdiscount').value;
        const pdesc = document.getElementById('mpdesc').value;
        if(pname&&pno&&pprice&&pdiscount) {
            console.log("New product added!!");
            //send to database
            if(pimg) {
                const uploadTask = storage.ref(`images/${pimg.name}`).put(pimg);
            uploadTask.on('state_changed',
            (snapshot) => {},
            (error) => {
                console.log('Firebase image upload error', error);
                this.notifyB('Error');
            },
            () => {
                storage.ref('images').child(pimg.name).getDownloadURL()
                    .then(url => {
                        console.log('URL', url);
                        const productData = {pid:this.props.row._id,pimg:
url,ptype,pcategory,pname,pno,pprice,pdiscount,pdesc};
                        axios.put("http://localhost:5000/farmer/productList/updat
e", {data: {data: productData}})
                        .then(res => {
                            console.log(res.data.message);
                            this.props.get();
                            document.getElementById('close').click();
                        });
                    })
```

```
                });
            } else {
                pimg = this.props.row.pimg;
                const productData =
{pid:this.props.row._id,pimg,ptype,pcategory,pname,pno,pprice,pdiscount,pdesc};
                    axios.put("http://localhost:5000/farmer/productList/update",
{data: {data: productData}})
                        .then(res => {
                            console.log(res.data.message);
                            this.props.get();
                            document.getElementById('close').click();
                        });
            }
        } else {
            console.log("Form not full!!");
        }
    }

    componentDidUpdate() {
        const row = this.props.row;
        const ptype = document.getElementById('mptype');
        const pcategory = document.getElementById('mpcategory');
        const pname = document.getElementById('mpname');
        //const pimg = document.getElementById('mpimg');
        const pno = document.getElementById('mpno');
        const pprice = document.getElementById('mpprice');
        const pdiscount = document.getElementById('mpdiscount');
        const pdesc = document.getElementById('mpdesc');
        if(row) {
            ptype.value = row.ptype;
            pcategory.value = row.pcategory;
            pname.value = row.pname;
            pno.value = row.pno;
            pprice.value = row.pprice;
            pdiscount.value = row.pdiscount;
            pdesc.value = row.pdesc;
        }
    }

    render(){
        return(
            <Fragment>
                    <button id="myBtn" onClick={this.handleShow} hidden>Open
Modal</button>
```

```
                    <div id="myModal" className="modal">
                        <div className="modal-content">
                            <div className="closecontainer">
                                <div className="heading"><h3>Edit
Product</h3></div>
                                <div id="close" className="close"
onClick={this.handleClose}>&times;</div>
                            </div>
                            <form>
                                <label>Product type:</label>
                                <input type="text" id="mptype" disabled/><br/>
                                <label>Product category:</label>
                                <input type="text" id="mpcategory"
disabled/><br/>
                                <label>Product name:</label>
                                <input type="text" id="mpname" name="mpname"
disabled/><br/>
                                <label>Product image:</label>
                                <img src={this.props.row.pimg}
alt={this.props.row.pname} className="m-1" width="100" height="100"/><br/>
                                <label>New product image:</label>
                                <input type="file" id="mpimg" name="mpimg"
accept="image/*"/><br/>
                                <label>Product quantity:</label>
                                <input type="number" id="mpno" name="mpno"
defaultValue="0"/><br/>
                                <label>Product price:</label>
                                <input type="number" id="mpprice" name="mpprice"
defaultValue="0"/><br/>
                                <label>Product discount:</label>
                                <input type="number" id="mpdiscount"
name="mpdiscount" defaultValue="0"/><br/>
                                <label>Product description:</label>
                                <input type="text" id="mpdesc" name="mpdesc"
defaultValue="0"/><br/>
                                <button type="button" className="btn btn-dark mx-
1" onClick={this.updateProduct}>Update</button>
                            </form>
                        </div>
                    </div>
                </Fragment>
            );
        }
}
```

record.js

```
import React, { Fragment } from 'react';
import Table from './Table';
import axios from 'axios';
import ModalComponent from './ModalComponent';
import Navbar from '../layout/navbar';
import './farmer.css';

export default class warehouse extends React.Component {
    constructor(props){
        super(props);
        this.state = {
            columns:
["PID","PIMAGE","CATEGORY","BRAND","NAME","QTY","PRICE","DISCOUNT","PDESC","DELET
E","EDIT"],
            productList: [],
            rowDetails: {}
        };
        this.handleDelete = this.handleDelete.bind(this);
        this.handleEdit = this.handleEdit.bind(this);
        this.getData = this.getData.bind(this);
        this.get = this.get.bind(this);
    }

    componentDidMount() {
        if(localStorage.getItem('token') === 'warehouse') {
            this.getData();
        }
    }

    componentDidUpdate() {
        const addProductButton = document.getElementById('addProduct');
        if(this.state.productList.length > 0) {
            addProductButton.hidden = false;
        } else {
            addProductButton.hidden = true;
        }
    }

    get() {
        this.getData();
    }
```

```
    getData() {
        axios.get("http://localhost:5000/farmer")
            .then(res => {
                console.log('Data from backend', res.data);
                const tempList = res.data.data.map(obj => {
                    delete obj.__v;
                    return obj;
                })
                tempList.sort((a,b) => (a.ptype > b.ptype) ? 1 : ((b.ptype > 
a.ptype) ? -1 : 0));
                this.setState({
                    productList: tempList
                });
            });
    }
    addProduct = e => {
        console.log("Add list of products to customer view table!!");
        const plantNutritionsList = [], ToolsList = [], horticultureSeedsList = 
[], forageSeedsList = [];
        for(let i=0;i<this.state.productList.length;i++) {
            if(this.state.productList[i].ptype === 'Plant Nutritions') {
                plantNutritionsList.push(this.state.productList[i]);
            }
            if(this.state.productList[i].ptype === 'Tools') {
                ToolsList.push(this.state.productList[i]);
            }
            if(this.state.productList[i].ptype === 'Horticulture Seeds') {
                horticultureSeedsList.push(this.state.productList[i]);
            }
            if(this.state.productList[i].ptype === 'Forage Seeds') {
                forageSeedsList.push(this.state.productList[i]);
            }
        }
        console.log('Plant Nutritions', plantNutritionsList);
        console.log('Tools', ToolsList);
        console.log('Forage Seeds', forageSeedsList);
        console.log('Horticulture Seeds', horticultureSeedsList);
        axios.post("http://localhost:5000/customer/productList/plantNutritions/ad
d", plantNutritionsList)
                    .then(res => {
                        console.log(res.data.message);
                    });
        axios.post("http://localhost:5000/customer/productList/tools/add", 
ToolsList)
```

```
                    .then(res => {
                        console.log(res.data.message);
                    });
        axios.post("http://localhost:5000/customer/productList/forageSeeds/add",
forageSeedsList)
                    .then(res => {
                        console.log(res.data.message);
                    });
        axios.post("http://localhost:5000/customer/productList/horticultureSeeds/
add", horticultureSeedsList)
                    .then(res => {
                        console.log(res.data.message);
                    });
    }
    handleDelete(row) {
        console.log("Delete!!", row);
        axios.delete("http://localhost:5000/farmer/productList/delete", {data:
{id: row._id}})
            .then(res => {
                console.log(res.data.message);
                this.getData();
            })
    }
    handleEdit(row) {
        console.log("Edit!!", row);
        this.setState({
            rowDetails: row
        });
        document.getElementById('myBtn').click();
    }
    render() {
        let elem = (<div style={{textAlign: "center"}}><h1>You need to login
first...</h1></div>);
        if(localStorage.getItem('token') === 'warehouse') {
            elem = (
                <span>
                <div className="productList">
                    <Table columns={this.state.columns}
data={this.state.productList} delete={this.handleDelete} edit={this.handleEdit}
hidden/><br/>
                    <div className="addbutton">
                        <button type="button" id="addProduct" className="btn btn-
dark mx-1" onClick={this.addProduct}><a href="/farmer/Success">Publish Your
Product</a></button>
                    </div>
```

```
                </div>
                <ModalComponent row={this.state.rowDetails} get={this.get} />
            </span>
            );
        }
        return(
            <Fragment>
                <Navbar type='warehouserecord'/>
                {elem}
            </Fragment>
        );
    }

}
```

Table.js

```
import React, { Fragment } from 'react';
import './farmer.css';

export default class Table extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            columns: [],
            data: [],
            pnamelist: [],
            pidlist: [],
            suggestions: []
        };
        this.handleChange = this.handleChange.bind(this);
        this.search = this.search.bind(this);
        this.handleKeyUp = this.handleKeyUp.bind(this);
        this.handleClick = this.handleClick.bind(this);
    }

    deleteProduct = (row) => {
        this.props.delete(row);
    }
    editProduct = (row) => {
        this.props.edit(row);
    }

    static getDerivedStateFromProps(props, state) {
        const temp = props.data;
        const tempname = [];
```

```
        const tempid = [];
        temp.forEach(obj => {
            tempname.push(obj.pname);
            tempid.push(obj._id);
        });
        return {
            columns: props.columns,
            data: props.data,
            pnamelist: tempname,
            pidlist: tempid
        };
    }

    handleChange(e) {
        const tempname = [...this.state.pnamelist];
        console.log(tempname);
        console.log(e.keyCode,e.target.value);
        const inputval = e.target.value;
        if(e.target.value.length === 0) {
            this.setState({
                suggestions: []
            });
            return;
        }
        let suggestedvalues = [];
        const tempid = this.state.pidlist;
        let key,value;
        for(key in tempname) {
            if(suggestedvalues.length <=5) {
                value = tempname[key];
                console.log(value, value.indexOf(inputval));
                if(value.toLowerCase().indexOf(inputval) > -1) {
                    suggestedvalues.push(value);
                }
            }
        }
        for(key in tempid) {
            if(suggestedvalues.length <=5) {
                value = tempid[key];
                if(value.toLowerCase().indexOf(inputval) > -1) {
                    suggestedvalues.push(value);
                }
            }        }
        this.setState({
            suggestions: [...suggestedvalues]
```

```
        });
        console.log(suggestedvalues);
    }
    handleKeyUp(e) {
        if (e.keyCode === 13) {
            const valtosearch = this.state.suggestions[0];
            this.search(valtosearch);
        }
    }
    handleClick() {
        this.search(this.state.suggestions[0]);
    }
    search(name) {
        console.log(name);
        let id = '';
        this.state.data.forEach(elem => {
            if(elem.pname === name) {
                id = elem._id;
            } else if(elem._id === name) {
                id = elem._id;
            }
        })
        document.getElementById("tabrow" + id).scrollIntoView();
    }

    render(){
        return(
            <Fragment>
                <div className="search my-0">
                    <input className="searchTerm" id="warehousesearch"
type="text" placeholder="Search by name or id..." onChange={this.handleChange}
onKeyUp={this.handleKeyUp}/>
                    <button type="button" className="searchbutton"
onClick={this.handleClick}>
                        Search
                    </button>
                </div>
                <ul className="searchContainer" id="searchContainer">
                {
                    this.state.suggestions.map(elem => {
                        return (
                            <li key={elem}>
                                <button type="button" className="searchSuggest"
                                onClick={() => {
                                    this.search(elem);
```

```
                                }}>{elem}</button>
                            </li>   )
                        })
                    }
                    </ul>
                    <table id="t01">
                        <tbody>
                        <tr>
                            {this.state.columns.map(data => <th
key={data}>{data}</th>)}
                        </tr>
                        {this.state.data.map(row => {
                                    return(
                                    <tr key={row._id} id={"tabrow" + row._id}>
                                        {Object.keys(row).map(rowdatakey => {
                                                if(rowdatakey !== 'pimg')
                                                return <td
key={rowdatakey}>{row[rowdatakey]}</td>
                                                else {
                                                    const img = row[rowdatakey];
                                                    return <td key={rowdatakey}><img
src={img} width="100" height ="100" alt={row['pname']}/></td>
                                                }
                                            })}
                                    <td><button type="button" className="editwh
mx-1" onClick= {() => {this.deleteProduct(row)}}><i className="fa fa-
remove"></i></button></td>
                                    <td><button type="button" className="delwh
mx-1" onClick= {() => {this.editProduct(row)}}><i className="fa fa-
edit"></i></button></td>
                                    </tr>
                                    );
                                })
                        }
                        </tbody>
                    </table>
                </Fragment>
            );
        }
}
```