

Project Name: Clearview Grocery

Team Leader: Dan Imbimbo

Team Members: Dan Imbimbo

Contributions percentage: 100% of the project was coded and documented by me.

Project Summary: The goal of this website was to act as an online e-store that would allow for customers to complete purchases and also provide relevant information stored in the store's database to employees; all the while, allowing the website to function easily on mobile devices through the use of bootstrap. (To do this, I utilized the final project I created by myself for Internet Computing and switched everything to utilize JavaScript & NodeJS).

Table of contents:

- ***NodeJS Packages Utilized:***

express; body-parser; mysql; bcryptjs; express-session; ejs; passport; passport-local; connect-flash

- ***Main Directory:***

clearview.sql – sql database initialization query; **server.js** – script file containing all of the logic and routes for the NodeJS web application

- ***Views/Images:***

Contains all of the images used across the website

- ***Views/CSS:***

main.css – holds CSS style rules

- ***Views/Partials:***

head.ejs – contains html metadata used across each page; **initHeader.ejs** – header for the initial website view; **custHeader.ejs** – header for the customer view of the website; **empHeader.ejs** – header for the employee view of the website; **footer.ejs** – footer for all views

- ***Views/Pages:***

welcome.ejs – initial website view

- ***Views/Pages/Customer:***

about.ejs – page displaying information about the storefront; **products.ejs** – displays the store’s product list page; **cart.ejs** - displays the customer’s current cart; **checkout.ejs** – displays the customer checkout form.

- **Views/Pages/Employee:**

login.ejs – displays the employee login page; **empTable.ejs** – displays the employee table page; **newEmp.ejs** – displays the new employee form; **prodTable.ejs** – displays the product table page, **newProd.ejs** – displays the new product form; **orderTable.ejs** – displays the order table page.

User Manual:

The website first allows users to choose whether they are a customer or an employee. Assuming they choose they are a customer, the first thing they will see is the about page of the website, which displays a brief description on the site itself. The customer view also has a navigation bar allowing for the customer to choose whether they want to navigate to the about page, the products page, their current cart, or return to the initial view. If they move to the products page, they will see that the products are listed by categories off to the left-hand side of the page, and they can choose between the different categories to view the products within each. Users can also choose to add one of the items to their cart, which will take them to their view of the cart. In the cart they can choose to remove 1 of any items that they have or decide to checkout. If they elect to checkout they are taken to the checkout form where they must enter their name, address, zip code, and card number before choosing to place the order. After placing the order they will be presented with a message thanking them for their purchase and sent back to the initial view.

In the case of choosing the employee option on the initial view, the user will be presented with a login screen where they must enter an email and password which will be checked against those in the database, and if it is valid then they will be presented with the table view of employees at the company. If they are logged in as the CEO, then they will see the option to fire employees and to add a new one, which will bring up a form for the new employee’s information. The navigation bar in the employee view provides the options of viewing the employee table, the products table, the orders table, or log out and return to the initial view. In the case of choosing the products table, it will display all of the products for purchase on the site and allow the CEO to add new products in a similar fashion to adding employees. And lastly, if the order table is chosen it will display all placed orders, allowing the CEO to delete any.

Source Code:

-clearview.sql

```
-- create and select the database
drop database if exists clearviewDatabase;
create database clearviewDatabase;
use clearviewDatabase;

create table Categories(
```

```
    categoryID int NOT NULL AUTO_INCREMENT,  
    categoryName varchar(255) NOT NULL,  
    PRIMARY KEY(categoryID)  
);  
  
create table Products(  
    productID int NOT NULL AUTO_INCREMENT,  
    categoryID int NOT NULL,  
    productName varchar(255) NOT NULL,  
    price decimal(10,2) NOT NULL,  
    imageLocation varchar(255) NOT NULL,  
    PRIMARY KEY(productID),  
    FOREIGN KEY(categoryID) REFERENCES Categories(categoryID)  
);  
  
create table Departments(  
    deptID int NOT NULL AUTO_INCREMENT,  
    deptName varchar(255) NOT NULL,  
    PRIMARY KEY(deptID)  
);  
  
create table Employees(  
    empID int NOT NULL AUTO_INCREMENT,  
    deptID int NOT NULL,  
    fName varchar(255) NOT NULL,  
    lName varchar(255) NOT NULL,  
    salary decimal(10,2) NOT NULL,  
    email varchar(255) NOT NULL,  
    password varchar(255) NOT NULL,  
    PRIMARY KEY(empID),  
    FOREIGN KEY(deptID) REFERENCES Departments(deptID)  
);  
  
create table Orders(  
    orderID int NOT NULL AUTO_INCREMENT,  
    orderTotal decimal(10,2),  
    orderDate datetime NOT NULL,  
    custFName varchar(255) NOT NULL,  
    custLName varchar(255) NOT NULL,  
    custStreet varchar(255) NOT NULL,  
    custCity varchar(255) NOT NULL,  
    custState varchar(2) NOT NULL,  
    custZip varchar(5) NOT NULL,  
    custCard varchar(255) NOT NULL,  
    PRIMARY KEY(orderID)
```

```

);
create table OrderItems(
    orderID int NOT NULL,
    productID int NOT NULL,
    quantity int NOT NULL,
    cost decimal(10,2) NOT NULL,
    CONSTRAINT oiPK PRIMARY KEY(orderID, productID),
    FOREIGN KEY(orderID) REFERENCES Orders(orderID),
    FOREIGN KEY(productID) REFERENCES Products(productID)
);

-- populate the database
insert into Categories(categoryName) VALUES
('Produce'),
('Beverages'),
('Tools');

insert into Products(categoryID,productName,price,imageLocation) VALUES
(1,'Honeycrisp Apple',1.34,'../../images/apple.jpg'),
(1,'Navel Orange',0.88,'../../images/orange.jpg'),
(2,'Alpine Spring Water, 1 Gal',0.98,'../../images/water.jpg'),
(2,'Dairy Pure Whole Milk, 1 Gal',4.42,'../../images/milk.jpg'),
(3,'Claw Hammer, 16oz',6.95,'../../images/hammer.jpg'),
(3,'Reversible Screwdriver, 6-in-1',7.95,'../../images/screwdriver.jpg');

insert into Departments(deptName) VALUES
('CEO'),
('Customer Service'),
('Accounting');

insert into Employees(deptID,fName,lName,salary,email,password) VALUES
(1,'Francis','Decker',55000,'fdecker@gmail.com','$2a$10$ztlvCln2VJrubSK3Kw0Jr7e.LueTn7xGPBdyDxTwxVq2MICsq0o6gq'),
(2,'Jimbo','Wickens',25000,'jwickens@gmail.com','$2a$10$cA3VIPst5omDzzTXnx/W50/dVHNR57J.u0botmrQCK4GXPw3bprf.'),
(3,'Melissa','Mackey',42000,'mmackey@gmail.com','$2a$10$SbjdSLD4pQHGbS3A7D9fKORBuZnaXa8S9lSFUiwM9YBRP10ix7Y5C');

insert into
Orders(orderDate,custFName,custLName,custStreet,custCity,custState,custZip,custCard)
VALUES

```

```
('2022-04-28 23:59:59','Nigel','Poedep','123 Easy St','Nowhere','NJ','04567','01234567 8901 2345');
```

```
insert into OrderItems(orderID,productID,quantity,cost) VALUES
(1,1,2,2*(SELECT price FROM Products WHERE productID = 1)),
(1,3,1,1*(SELECT price FROM Products WHERE productID = 3)),
(1,5,1,1*(SELECT price FROM Products WHERE productID = 5));
```

```
UPDATE Orders
SET orderTotal = (SELECT SUM(cost) FROM OrderItems WHERE orderID = 1) * (1 + 0.07)
WHERE orderID = 1;
```

```
-- Create a user named clearview with password as clearviewpass
GRANT SELECT, INSERT, DELETE, UPDATE
ON clearviewDatabase.*
TO clearview@localhost
IDENTIFIED BY 'clearviewpass';
```

-server.js

```
////////////////////////////////////
//Initialization - START//
////////////////////////////////////
//Configuring express server
const express = require('express');
const app = express();

//Body parser config
const bodyparser = require('body-parser');
app.use(bodyparser.json());
app.use(bodyparser.urlencoded({ extended: true }));

//Static Middleware
const path = require('path');
app.use(express.static(path.join(__dirname, 'views')))

//Set EJS as templating engine
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

//Session config
const session = require('express-session');
```

```

app.use(
  session({
    secret: "secret_string",
    cookie: { maxAge: null }, //auto-delete after closing browser
    saveUninitialized: true,
    resave: false
  })
);

//Port config
const port = 7000;
app.listen(port, () => console.log('Listening on port %s', port));

//Defining MySQL database object
const mysql = require('mysql');
const mysqlConnection = mysql.createConnection({
  host: 'localhost',
  user: 'clearview',
  password: 'clearviewpass',
  database: 'clearviewDatabase',
  multipleStatement: true
});

//Establish connection with database using MySQL object's credentials
mysqlConnection.connect((err) =>{
  if(!err)
    console.log('Connection Established Successfully');
  else
    console.log('Connection Failed!');
});

//////////
//PassportJS - Start//
//////////
//Login authentication config
const passport = require('passport');
const LocalStrategy = require('passport-local').Strategy;
app.use(passport.initialize());
app.use(passport.session());
const flash = require('connect-flash');
app.use(flash());
const bcrypt = require('bcryptjs');

```

```

const verifyCallback = (email, password, done) =>{
  //checks if the entered email address is in the database
  mysqlConnection.query('SELECT * FROM employees where email =?', [email],
function(err, row){
  if(err){
    return done(err);
  }

  //checks if matching email was even found
  if(row.length == 0){
    return done(null, false, ('error', 'Invalid login.));
  }

  //checks if the password matches the one associated with email in database
  const isValid = validPassword(password, row[0].password);

  user = {
    empID : row[0].empID,
    deptID : row[0].deptID,
    email : row[0].email
  };

  //if the password matches then serialize the user
  if(isValid){
    return done(null, user);
  }
  else
    return done(null, false, ('error', 'Invalid login.));
  });
};

passport.use(new LocalStrategy({ usernameField:'email', passwordField:'pass' },
verifyCallback));

passport.serializeUser((user,done) =>{
  done(null, user.empID);
});

passport.deserializeUser((empID,done) =>{
  mysqlConnection.query('SELECT * FROM employees where empID =?', [empID], (err,
row) =>{

```



```

        if(err){
            return done(err);
        }
        done(null, row[0]);
    });
});

//Checks if the entered password matches the hash
function validPassword(password, hash){
    var hashVerify = bcrypt.compareSync(password,hash);
    return hashVerify;
}

//Determines if an employee has logged in
function isAuth(req,res,next){
    if(req.isAuthenticated())
        next();
    else{
        req.flash('error', 'Please login to access desired page.')
        res.redirect('/login');
    }
}

//Determines the CEO is logged in
function isAdmin(req,res,next){
    if(req.isAuthenticated() && req.user.empID == 1)
        next();
    else{
        req.flash('error', 'You do not have access to that page.');
        res.redirect('/empTable');
    }
}

//Determines if the entered employee email already exists in the database
function emailExists(req,res,next){
    mysqlConnection.query('SELECT * FROM employees where email =?', [req.body.email],
    (err, row) =>{
        if(err)
            console.log(err);
        else if(row.length > 0){

```

```

        req.flash('error', 'Email address already exists in the database.');
```

```

        res.redirect('/newEmployee');
```

```

    }
    else
        next();
});
}

//Determines if the entered product already exists in the database
function productExists(req,res,next){
    mysqlConnection.query('SELECT * FROM products where productName =?',
[req.body.prodName], (err, row) =>{
    if(err)
        console.log(err);
    else if(row.length > 0){
        req.flash('error', 'Product already exists in the database.');
```

```

        res.redirect('/newProduct');
```

```

    }
    else
        next();
});
}

//////////
//PassportJS - End//
//////////
//////////
//Initialization - END //
//////////
//Default end point
app.get('/', (req, res) =>{
    res.redirect('/welcome');
});

//Router to welcome page
app.get('/welcome', (req, res) =>{
    var userSession = req.session; //variable for accessing session
    userSession.cart = {}; //sets session cart to empty
    userSession.runningTotal = 0; //sets running total to 0

    res.render('pages/welcome');
```

```

});

```

```

//////////
//Customer - START//
//////////
//Router to customer about page
app.get('/about', (req, res) =>{
    res.render('pages/customer/about');
});

//Router to customer product list page (sorted by category ID)
app.get('/products/:categoryID', (req, res) =>{
    //Retrieves all categories
    mysqlConnection.query('SELECT * FROM categories', (err,cats) =>{
        if(!err){
            //Retrieves all products associated with category ID
            mysqlConnection.query('SELECT * FROM products WHERE categoryID =?',
[req.params.categoryID], (err,prods) =>{
                if(!err){
                    //Retrieves category associated with category ID
                    mysqlConnection.query('SELECT * FROM categories WHERE categoryID
=?', [req.params.categoryID], (err,row) =>{
                        if(!err){
                            res.render('pages/customer/productList', { categories :
cats, products : prods, selectedCat : row[0] });
                        }
                        else
                            console.log(err);
                    });
                }
                else
                    console.log(err);
            });
        }
        else
            console.log(err);
    });
});

//Router for processing the add to cart form action
app.post('/addToCart', (req, res) =>{
    var userSession = req.session; //variable for accessing session

```

```

//retrieves the item from its submitted form
var item = {
  productID: req.body.productID,
  productName: req.body.productName,
  productPrice: req.body.productPrice,
  quantity: 1,
  imageUrl: req.body.imageUrl
};

//if the item is already in the cart then increase its quantity
//otherwise adds the item to the cart
if(typeof userSession.cart[item.productID] !== 'undefined')
  userSession.cart[item.productID].quantity += 1;
else
  userSession.cart[item.productID] = item;

//adds the item's price + tax to the running total
userSession.runningTotal += (item.productPrice * (1+0.07));

res.redirect('/cart');
});

//Router for processing the remove from cart form action
app.post('/removeFromCart', (req, res) =>{
  var userSession = req.session; //variable for accessing session

  var ID = req.body.productID; //retrieves the item ID from its submitted form

  //updates the running total to reflect the change in quantity
  userSession.runningTotal -= (userSession.cart[ID].productPrice * (1+0.07));

  //if the item's quantity is greater than 1 then decrease its quantity
  //otherwise remove the item from the cart entirely
  if(userSession.cart[ID].quantity > 1){
    userSession.cart[ID].quantity -= 1;
  }
  else
    delete userSession.cart[ID];

  res.redirect('/cart');
});

```

```
//Router to the cart page
app.get('/cart', (req, res) =>{
    var userSession = req.session; //variable for accessing session

    //Debug display
    console.log("Cart:");
    console.log(userSession.cart);
    console.log("Total:");
    console.log(userSession.runningTotal);

    res.render('pages/customer/cart', { cart : userSession.cart, total:
userSession.runningTotal });
});

//Router for processing the checkout form action
app.post('/checkout', (req,res) =>{
    res.render('pages/customer/checkout');
});

//Router for processing the place order form action
app.post('/placeOrder', (req, res) =>{
    var userSession = req.session; //variable for accessing session

    //retrieves the checkout details from its submitted form
    var first = req.body.first;
    var last = req.body.last;
    var street = req.body.street;
    var city = req.body.city;
    var state = req.body.state;
    var zip = req.body.zip;
    var card = req.body.card;

    //determine the current data and time
    var today = new Date();
    var date = today.getFullYear()+'-'+(today.getMonth()+1)+'-'+today.getDate();
    var time = today.getHours() + ":" + today.getMinutes() + ":" + today.getSeconds();
    var dateTime = date+' '+time;

    //stored running total (rounded to 2 decimal places)
    var orderTotal = userSession.runningTotal.toFixed(2);
```

```

//calls function to add the order to the database
addOrder(dateTime,orderTotal,first,last,street,city,state,zip,card);

//query to obtain created order's ID
var query = "SELECT * FROM Orders "+
            "WHERE orderDate = '"+dateTime+
            "' AND orderTotal = '"+orderTotal+"' AND custLName = '"+last+"'";

//retrieves the order ID from the database
mysqlConnection.query(query,(err, row) =>{
    if(!err){
        var ID = row[0].orderId;
        var items = {cart: userSession.cart}
        //calls function to add each order item to the database
        addOrderItems(ID, items);
    }
    else
        console.log(err);
});

res.redirect('/welcome');
});

//Adds the given information into the database as an order
function addOrder(dateTime,orderTotal,first,last,street,city,state,zip,card){
    var query = "INSERT INTO orders"+

    "(orderDate,orderTotal,custFName,custLName,custStreet,custCity,custState,custZip,custC
ard) "+

        "VALUES"+
        "('"+dateTime+"','"+orderTotal+"','"+first+"','"+last+"','"+street+
        "','"+city+"','"+state+"','"+zip+"','"+card+"')";

    //adds the order to the database
    mysqlConnection.query(query,(err, rows) =>{
        if(!err)
            console.log("Order successfully inserted.");
        else
            console.log(err);
    });
}

```

```

    });
}

//Adds the cart's items to the database under the given order ID
function addOrderItems(orderID, items){
    //loops through each item in the cart and adds it to the database
    Object.keys(items.cart).forEach(function(prodID){
        var cost = items.cart[prodID].quantity * items.cart[prodID].productPrice;
//calculates the total cost of each item
        var query = "INSERT INTO orderitems"+
                    "(orderID,productID,quantity,cost) "+
                    "VALUES"+

"('"+orderID+"','"+prodID+"','"+items.cart[prodID].quantity+"','"+cost.toFixed(2)+"')";
;

        //adds the order item to the database
        mysqlConnection.query(query,(err, rows) =>{
            if(!err)
                console.log("Order item successfully inserted.");
            else
                console.log(err);
        });
    });
};

//////////
//Customer - END//
//////////
//////////
//Employee - START//
//////////
//Router to login page
app.get('/login', (req,res) =>{
    const errors = req.flash().error || [];
    res.render('pages/employee/login', { errors });
});

//Router for processing login form action
app.post('/login', passport.authenticate('local', {
    successRedirect: '/empTable',
    failureRedirect: '/login',

```

```

        failureFlash: true
    }));

//Router for processing logout
app.get('/logout', (req,res) =>{
    //deletes the user from session and returns to welcome
    req.logout(function(err) {
        if (err) { return next(err); }
        res.redirect('/welcome');
    });
});

//Router to employee table
app.get('/empTable', isAuthenticated, (req,res) =>{
    const errors = req.flash().error || [];
    mysqlConnection.query('SELECT * FROM employees', (err, rows) =>{
        if(!err){
            res.render('pages/employee/empTable', { employees : rows, user, errors });
        }
        else
            console.log(err);
    });
});

//Router for processing fire employee form action
app.post('/fireEmployee', isAdmin, (req, res) =>{
    mysqlConnection.query('DELETE FROM employees WHERE empID =?', [req.body.empID],
    (err, rows) =>{
        if(!err){
            res.redirect('/empTable');
        }
        else
            console.log(err);
    });
});

//Router to new employee page
app.get('/newEmployee', isAdmin, (req, res) =>{
    const errors = req.flash().error || [];
    res.render('pages/employee/newEmp', { errors });
})

```



```

//Router for processing hire employee form action
app.post('/hireEmployee', isAdmin, emailExists, (req, res) =>{
    var first = req.body.first;
    var last = req.body.last;
    var salary = req.body.salary;
    var deptID = req.body.dept;
    var email = req.body.email;
    var password = hashPassword(req.body.pass);

    //inserts new employee into the database and returns to employee table
    query = "INSERT INTO employees(deptID, fName, lName, salary, email, password) "+
    "VALUES('"+deptID+"', '"+first+"', '"+last+"', '"+salary+"', '"+email+"', '"+password+"')";
    mysqlConnection.query(query, (err, rows) =>{
        if(err)
            console.log(err);
        else
            res.redirect('/empTable');
    });
});

//Hashes the newly entered password
function hashPassword(password){
    var hashedPass = bcrypt.hashSync(password);
    return hashedPass;
}

//Router to product table
app.get('/prodTable', isAuth, (req, res) =>{
    mysqlConnection.query('SELECT * FROM products', (err, rows) =>{
        if(!err){
            res.render('pages/employee/prodTable', { products : rows, user });
        }
        else
            console.log(err);
    });
});

//Router to new product page
app.get('/newProduct', isAdmin, (req, res) =>{

```

```

    const errors = req.flash().error || [];
    res.render('pages/employee/newProd', { errors });
  })

  //Router for processing add product form action
  app.post('/addProduct', isAdmin, productExists, (req, res) =>{
    var categoryID = req.body.catID;
    var productName = req.body.prodName;
    var price = req.body.price;
    var imageLocation = "../..../images/"+req.body.imageName;

    //inserts new product into the database and returns to product table
    query = "INSERT INTO products(categoryID, productName, price, imageLocation) "+
    "VALUES('"+categoryID+"', '"+productName+"', '"+price+"', '"+imageLocation+"')";
    mysqlConnection.query(query, (err, rows) =>{
      if(err)
        console.log(err);
      else
        res.redirect('/prodTable');
    });
  });

  //Router to order table
  app.get('/orderTable', isAuthenticated, (req, res) =>{
    mysqlConnection.query('SELECT * FROM orders', (err, rows) =>{
      if(!err){
        res.render('pages/employee/orderTable', { orders : rows, user });
      }
      else
        console.log(err);
    });
  });

  //Router for processing remove order form action
  app.post('/removeOrder', isAdmin, (req, res) =>{
    mysqlConnection.query('DELETE FROM orderItems WHERE orderID =?',
    [req.body.orderID], (err, rows) =>{
      if(!err){
        mysqlConnection.query('DELETE FROM orders WHERE orderID =?',
        [req.body.orderID], (err, rows) =>{

```

```

                if(!err){
                    res.redirect('/orderTable');
                }
                else
                    console.log(err);
            });
        }
        else
            console.log(err);
    })
});

//////////
//Employee - END//
//////////

```

-main.css

```

header {
    border-bottom: 2px solid;
    padding-bottom: 1em;
}

footer {
    border-top: 2px solid;
    padding: 1em 1em;
    text-align: right;
}

body {
    background-color: wheat;
}

th {
    padding: 10px 10px;
}

td {
    padding: 10px 10px;
}

.picture{
    padding-right: 0px;
}

.navbar{
    background-color: brown;
}

```

```
.btn.custom{
  border:1px solid #767676;
  color: black;
  background-color: #EFEFEF;
}
.btn.custom:hover{
  border:1px solid #4F4F4F;
  background-color: #E5E5E5;
}
```

-head.ejs

```
<!-- the head section -->
<head>
  <title>Clearview Grocery</title>
  <meta charset = "utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-U02eT0CpHqdsSJQ6hJty5KVphtPhzWj9W01c1HTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

  <link rel="stylesheet" type="text/css" href="/css/main.css">
</head>
```

-initHeader.ejs

```
<header>
  <!--Navbar START-->
```

```

    <nav class="navbar navbar-expand-sm navbar-dark sticky-top">
      <a class="navbar-brand"></a>
    </nav>
    <!--Navbar END-->
</header>

```

-footer.ejs

```

<footer>
  <p class="copyright">
    <b>&copy; 2022 Clearview</b>
  </p>
</footer>

```

-custHeader.ejs

```

<header>
  <!--Navbar START-->
  <nav class="navbar navbar-expand-sm navbar-dark sticky-top">
    <a class="navbar-brand"></a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#collapsingNav">
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="collapsingNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item">
          <a class="nav-link" href="/about">about</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/products/1">products</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/cart">cart</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/welcome" style="font-weight:bold"
onclick="message()">EXIT</a>
        </li>
      </ul>
    </div>
  </nav>

```

```

        </div>
    </nav>
    <!--Navbar END-->

    <script>
        function message(){
            alert('We hope to see you again soon!');
        }
    </script>
</header>

```

-empHeader.ejs

```

<header>
    <!--Navbar START-->
    <nav class="navbar navbar-expand-sm navbar-dark sticky-top">
        <a class="navbar-brand"></a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#collapsingNav">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="collapsingNav">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item">
                    <a class="nav-link" href="/empTable">Employees</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/prodTable">Products</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/orderTable">Orders</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/logout" style="font-weight:bold"
onclick="message()">LOG OUT</a>
                </li>
            </ul>
        </div>
    </nav>
    <!--Navbar END-->

```

```

    <script>
      function message(){
        alert('Logging out...');
      }
    </script>
  </header>

```

-welcome.ejs

```

<!DOCTYPE html>
<html>
  <!--Includes the defined head-->
  <%- include('../partials/head') %>

  <body>
    <!--Includes the defined header-->
    <%- include('../partials/initHeader') %>

    <main>
      <div class="container-fluid" style="text-align:center">
        <h1>Welcome!</h1>
        <p></p>
        <!--Links to either user view or employee view-->
        <div style="padding-bottom: 1em">
          <h6>Please select your role:</h6>
          <a href="/about" class="btn btn-success"
onclick=test()>Customer</a>
          <a href="/login" class="btn btn-success">Employee</a>
        </div>
      </div>
    </main>

    <!--Includes the defined footer-->
    <%- include('../partials/footer') %>
  </body>
</html>

```

-about.ejs

```

<!DOCTYPE html>
<html>

```

```

<!--Includes the defined head-->
<%- include('../..partials/head') %>

<body>
  <!--Includes the defined header-->
  <%- include('../..partials/custHeader') %>

  <main>
    <div class="container-fluid" style="padding:.5em">
      <p></p>
      <h6>
        Hello Customer,<br><br>
        WELCOME!!! You have stumbled upon the greatest store known to mankind,
Clearview Gorcery!<br><br>
        We are the #1 rated totally-not-fake destination for all your e-
commerce needs.<br><br>
        On this site you will find a curated selection of products ranging
from Beverages, Produce, and even Tools.<br><br>
        <b>-Website by Dan Imbimbo (Using NodeJS + JavaScript)</b>
      </h6>
    </div>
  </main>

  <!--Includes the defined footer-->
  <%- include('../..partials/footer') %>
</body>
</html>

```

-productList.ejs

```

<!DOCTYPE html>
<html>
  <!--Includes the defined head-->
  <%- include('../..partials/head') %>

  <body>
    <!--Includes the defined header-->
    <%- include('../..partials/custHeader') %>

    <main>
      <h1 style="padding-left:5px">Product Inventory</h1>

```



```

<div class="row container-fluid">
  <div class="col-sm-3">
    <!-- lists all categories -->
    <h2>Categories</h2>
    <nav>
      <ul>
        <% categories.forEach(function(category){ %>
          <li>
            <a href="/products/<%= category.categoryID %>"><%=
category.categoryName %></a>
          </li>
        <% }) %>
      </ul>
    </nav>
  </div>
  <div class="col-md">
    <!-- displays all products -->
    <h2><%= selectedCat.categoryName %></h2>
    <table>
      <tr>
        <th>Picture</th>
        <th>Name</th>
        <th>Price</th>
      </tr>
      <% products.forEach(function(product){ %>
        <tr>
          <td class="picture"><img src=<%= product.imageLocation %>
alt="Product Image" max-width="500px" width="35%" height="auto"></td>
          <td><%= product.productName %></td>
          <td>$<%= product.price %></td>
          <td>
            <form action="/addToCart" method="post">
              <input type="hidden" name="productID"
                value="<%= product.productID %>">
              <input type="hidden" name="productName"
                value="<%= product.productName %>">
              <input type="hidden" name="productPrice"
                value="<%= product.price %>">
              <input type="hidden" name="imageLocation"
                value="<%= product.imageLocation %>">

```

```

        <input type="submit" value="Add">
      </form>
    </td>
  </tr>
<% }) %>
</table>
</div>
</div>
</main>

<!--Includes the defined footer-->
<%- include('../..partials/footer') %>
</body>
</html>

```

-cart.ejs

```

<!DOCTYPE html>
<html>
<!--Includes the defined head-->
<%- include('../..partials/head'); %>

<body>
  <!--Includes the defined header-->
  <%- include('../..partials/custHeader'); %>

  <main>
    <h1 style="padding-left:5px">Cart</h1>

    <div class="row container-fluid">
      <div class="col-xl">
        <table>
          <!--Checks if the cart is empty-->
          <% if(Object.keys(cart) != 0) { %>
            <tr>
              <th>Picture</th>
              <th>Name</th>
              <th>Quantity</th>
              <th>Price</th>
            </tr>
            <!--Loops through and displays each item in the cart-->
            <% Object.keys(cart).forEach(function(item){ %>

```

```

        <tr>
            <td class="picture"><img src=<%=
cart[item].imageLocation %> alt="Product Image" max-width="500px" width="40%";
height="auto"></td>

            <td><%= cart[item].productName %></td>
            <td>x<%= cart[item].quantity %></td>
            <td>$<%= cart[item].productPrice %></td>
            <td>
                <form action="/removeFromCart" method="post">
                    <input type="hidden" name="productID"
                        value="<%= cart[item].productID %>">
                    <input type="submit" value="Remove">
                </form>
            </td>
        </tr>
    <% }) %>
    <!--Displays checkout button and cart total with tax-->
    <tr>
        <td>
            <h6>Total including tax (7%): $<%= total.toFixed('2')
%></h6>

            <form action="/checkout" method="post">
                <input type="submit" value="Checkout">
            </form>
        </td>
    </tr>
    <% } else{ %>
        <tr>
            <h6>Cart is empty.</h6>
        </tr>
    <% } %>
</table>
</div>
</div>
</main>

    <!--Includes the defined footer-->
    <%- include('../..partials/footer'); %>
</body>
</html>

```

-checkout.ejs

```
<!DOCTYPE html>
<html>
<!--Includes the defined head-->
<%- include('../..../partials/head'); %>

<body>
  <!--Includes the defined header-->
  <%- include('../..../partials/custHeader'); %>

  <main>
    <h1>Checkout</h1>

    <div class="container-fluid">
      <form id="form" action ="/placeOrder" method ="post">
        <div class="form-group">
          <strong>Name</strong>
        </div>
        <div class="form-group row">
          <label for="first" class="col-sm-2 col-form-label">First
Name:</label>
          <div class="col-sm-10">
            <input type = "text" class = "form-control" name ="first"
placeholder = "Bob" pattern="[A-Za-z]+" required>
          </div>
        </div>
        <div class="form-group row">
          <label for="last" class="col-sm-2 col-form-label">Last
Name:</label>
          <div class="col-sm-10">
            <input type = "text" class = "form-control" name ="last"
placeholder = "McGee" pattern="[A-Za-z]+" required>
          </div>
        </div>
        <div class="form-group">
          <strong>Address</strong>
        </div>
        <div class="form-group row">
          <label for="street" class="col-sm-2 col-form-
label">Street:</label>
          <div class="col-sm-10">
```

```
        <input type = "text" class = "form-control" name = "street"
placeholder = "123 Anywhere Ave" pattern="[A-Za-z0-9\s]+" required>
    </div>
</div>
<div class="form-group row">
    <label for="city" class="col-sm-2 col-form-label">City:</label>
    <div class="col-sm-10">
        <input type = "text" class = "form-control" name = "city"
placeholder = "Some City" pattern="[A-Za-z0-9\s]+" required>
    </div>
</div>
<div class="form-group row">
    <label for="state" class="col-sm-2 col-form-label">State:</label>
    <div class="col-sm-10">
        <select class = "form-control" name = "state" required>
            <option value="AL" selected>(AL) Alabama</option>
            <option value="AK">(AK) Alaska</option>
            <option value="AZ">(AZ) Arizona</option>
            <option value="AR">(AR) Arkansas</option>
            <option value="CA">(CA) California</option>
            <option value="CO">(CO) Colorado</option>
            <option value="CT">(CT) Connecticut</option>
            <option value="DE">(DE) Delaware</option>
            <option value="DC">(DC) District Of Columbia</option>
            <option value="FL">(FL) Florida</option>
            <option value="GA">(GA) Georgia</option>
            <option value="HI">(HI) Hawaii</option>
            <option value="ID">(ID) Idaho</option>
            <option value="IL">(IL) Illinois</option>
            <option value="IN">(IN) Indiana</option>
            <option value="IA">(IA) Iowa</option>
            <option value="KS">(KS) Kansas</option>
            <option value="KY">(KY) Kentucky</option>
            <option value="LA">(LA) Louisiana</option>
            <option value="ME">(ME) Maine</option>
            <option value="MD">(MD) Maryland</option>
            <option value="MA">(MA) Massachusetts</option>
            <option value="MI">(MI) Michigan</option>
            <option value="MN">(MN) Minnesota</option>
            <option value="MS">(MS) Mississippi</option>
            <option value="MO">(MO) Missouri</option>
```

```

        <option value="MT">(MT) Montana</option>
        <option value="NE">(NE) Nebraska</option>
        <option value="NV">(NV) Nevada</option>
        <option value="NH">(NH) New Hampshire</option>
        <option value="NJ">(NJ) New Jersey</option>
        <option value="NM">(NM) New Mexico</option>
        <option value="NY">(NY) New York</option>
        <option value="NC">(NC) North Carolina</option>
        <option value="ND">(ND) North Dakota</option>
        <option value="OH">(OH) Ohio</option>
        <option value="OK">(OK) Oklahoma</option>
        <option value="OR">(OR) Oregon</option>
        <option value="PA">(PA) Pennsylvania</option>
        <option value="RI">(RI) Rhode Island</option>
        <option value="SC">(SC) South Carolina</option>
        <option value="SD">(SD) South Dakota</option>
        <option value="TN">(TN) Tennessee</option>
        <option value="TX">(TX) Texas</option>
        <option value="UT">(UT) Utah</option>
        <option value="VT">(VT) Vermont</option>
        <option value="VA">(VA) Virginia</option>
        <option value="WA">(WA) Washington</option>
        <option value="WV">(WV) West Virginia</option>
        <option value="WI">(WI) Wisconsin</option>
        <option value="WY">(WY) Wyoming</option>
    </select>
</div>
</div>
<div class="form-group row">
    <label for="zip" class="col-sm-2 col-form-label">Zip Code:</label>
    <div class="col-sm-10">
        <input type = "text" class = "form-control" name ="zip"
placeholder = "01234" maxlength=5 pattern ="[0-9]+" required>
    </div>
</div>
<div class="form-group row">
    <label for="card" class="col-sm-2 col-form-label">Card
Number:</label>
    <div class="col-sm-10">

```

```

        <input type = "text" class = "form-control" name = "card"
placeholder = "#### #### #### ####" maxlength="19" pattern="\d{4} +\d{4} +\d{4} +\d{4}" required>
    </div>
</div>
<div class="form-group">
    <input type="submit" value="Place Order">
</div>
</form>
</div>
</main>

<!--Includes the defined footer-->
<%- include('../..partials/footer'); %>
</body>

<script>
    var form = document.getElementById("form");
    form.addEventListener("submit", (e) => {
        e.preventDefault();
        alert("Your order has been placed. Thank you!");
        form.submit();
    });
</script>
</html>

```

-login.ejs

```

<!DOCTYPE html>
<html>
<!--Includes the defined head-->
<%- include('../..partials/head'); %>

<body>
    <!--Includes the defined header-->
    <%- include('../..partials/initHeader'); %>

    <main>
        <div class="container-fluid">
            <h1 style="text-align:center; padding-top: .5em;">Employee Login</h1>
            <!--Displays any errors-->
            <% for(let i=0; i < errors.length; i++){ %>

```

```

        <h6 style="color:red"><%= errors[i] %></h6>
    <% } %>
    <form action = "/login" method = "post">
        <div class="form-group row">
            <label for="email" class="col-sm-2 col-form-label">Email:</label>
            <div class="col-sm-10">
                <input type="email" class="form-control" name="email"
placeholder="Enter email" required>
            </div>
        </div>
        <div class="form-group row">
            <label for="pass" class="col-sm-2 col-form-
label">Password:</label>
            <div class="col-sm-10">
                <input type="password" class = "form-control" name="pass"
placeholder="Enter password" required>
            </div>
        </div>
        <div class="form-group row">
            <label class="col-sm-2 col-form-label"></label>
            <div class="col-sm-10">
                <button type="submit" class="btn btn-primary">Login</button>
            </div>
        </div>
        <div class="form-group row">
            <label class="col-sm-2 col-form-label"></label>
            <div class="col-sm-10">
                <p>Not an employee?</p>
                <a href="/welcome" class="btn btn-success">Go back</a>
            </div>
        </div>
    </div>
</main>

<!--Includes the defined footer-->
<%= include('../partials/footer'); %>
</body>
</html>

```

-empTable.ejs

<!DOCTYPE html>


```

<html>
<!--Includes the defined head-->
<%- include('../..//partials/head'); %>

<body>
  <!--Includes the defined header-->
  <%- include('../..//partials/empHeader'); %>

  <main>
    <div class="container-fluid">
      <h1>Employee List</h1>
      <h6>(Logged in: <%= user.email %>)</h6>
      <!--Displays any errors-->
      <% for(let i=0; i < errors.length; i++){ %>
        <h6 style="color:red"><%= errors[i] %></h6>
      <% } %>
      <div class="table-responsive">
        <!-- displays all employees -->
        <table>
          <tr>
            <th>Employee ID</th>
            <th>Department ID</th>
            <th>Email</th>
            <th>Employee Name</th>
            <!-- also displays salary, email, and password if CEO is
logged in -->
            <% if(user.deptID == 1){ %>
              <th>Salary</th>
            <% } %>
          </tr>
          <% employees.forEach(function(employee){ %>
            <tr>
              <td><%= employee.empID %></td>
              <td><%= employee.deptID %></td>
              <td><%= employee.email %></td>
              <td><%= employee.fName + " " + employee.lName %></td>
              <!-- also displays salary, email, and password if CEO is
logged in -->
              <% if(user.deptID == 1){ %>
                <td><%= employee.salary %></td>
                <!-- don't show fire next to CEO -->

```

```

        <% if(employee.deptID != 1){ %>
            <td>
                <form action="/fireEmployee" method="post">
                    <input type="hidden" name="empID"
                        value="<%= employee.empID %>">
                    <input type="submit" value="Fire">
                </form>
            </td>
        <% } %>
    <% } %>
</tr>
<% }) %>
<!-- displays option to add a new employee if CEO is logged in -->
<% if(user.deptID == 1){ %>
    <tr>
        <td><a class="btn custom" href="/newEmployee">Add new
employee</a></td>
    </tr>
<% } %>
</table>
</div>
</div>
</main>

<!--Includes the defined footer-->
<%- include('../..partials/footer'); %>
</body>
</html>

```

-newEmp.ejs

```

<!DOCTYPE html>
<html>
<!--Includes the defined head-->
<%- include('../..partials/head'); %>

<body>
    <!--Includes the defined header-->
    <%- include('../..partials/empHeader'); %>

    <main>
        <div class="container-fluid">

```

```

<h1>New Employee</h1>
<!--Displays any errors-->
<% for(let i=0; i < errors.length; i++){ %>
    <h6 style="color:red"><%= errors[i] %></h6>
<% } %>
<form action = "/hireEmployee" method = "post">
    <div class="form-group row">
        <label for="first" class="col-sm-2 col-form-label">First
Name:</label>
        <div class="col-sm-10">
            <input type = "text" class = "form-control" name ="first"
placeholder = "John" pattern="[A-Za-z]+" required>
        </div>
    </div>
    <div class="form-group row">
        <label for="last" class="col-sm-2 col-form-label">Last
Name:</label>
        <div class="col-sm-10">
            <input type = "text" class = "form-control" name ="last"
placeholder = "Smith" pattern="[A-Za-z]+" required>
        </div>
    </div>
    <div class="form-group row">
        <label for="street" class="col-sm-2 col-form-
label">Salary:</label>
        <div class="col-sm-10">
            <input type = "number" class = "form-control" name ="salary"
placeholder = "20000" min="0" pattern ="[0-9]+" required>
        </div>
    </div>
    <div class="form-group row">
        <label for="state" class="col-sm-2 col-form-
label">Department:</label>
        <div class="col-sm-10">
            <select class = "form-control" name ="dept" required>
                <option value="2" selected>Customer Service</option>
                <option value="3">Accounting</option>
            </select>
        </div>
    </div>
    <div class="form-group row">

```

```

        <label for="city" class="col-sm-2 col-form-label">Email:</label>
        <div class="col-sm-10">
            <input type="email" class="form-control" name="email"
placeholder="example@gmail.com" required>
        </div>
    </div>
    <div class="form-group row">
        <label for="pass" class="col-sm-2 col-form-
label">Password:</label>
        <div class="col-sm-10">
            <input type="password" class = "form-control" name="pass"
placeholder="password" required>
        </div>
    </div>
    <div class="form-group">
        <input type="submit" value="Hire Employee">
    </div>
</form>
</div>
</main>

<!--Includes the defined footer-->
<%- include('../..partials/footer'); %>
</body>
</html>

```

-prodTable.ejs

```

<!DOCTYPE html>
<html>
<!--Includes the defined head-->
<%- include('../..partials/head'); %>

<body>
    <!--Includes the defined header-->
    <%- include('../..partials/empHeader'); %>

    <main>
        <div class="container-fluid">
            <h1>Product List</h1>
            <h6>(Logged in: <%= user.email %>)</h6>
            <div class="table-responsive">

```

```

        <!-- displays all products -->
        <table>
            <tr>
                <th>Product ID</th>
                <th>Category ID</th>
                <th>Product Name</th>
                <th>Price</th>
                <th>Image (Location)</th>
            </tr>
            <% products.forEach(function(product){ %>
                <tr>
                    <td><%= product.productID %></td>
                    <td><%= product.categoryID %></td>
                    <td><%= product.productName %></td>
                    <td><%= product.price %></td>
                    <td><%= product.imageLocation %></td>
                </tr>
                <% }) %>
            <!-- displays option to add a new product if CEO is logged in -->
            <% if(user.deptID == 1){ %>
                <tr>
                    <td><a class="btn custom" href="/newProduct">Add new
product</a></td>
                </tr>
            <% } %>
        </table>
    </div>
</div>
</main>

    <!--Includes the defined footer-->
    <%- include('../partials/footer'); %>
</body>
</html>

```

-newProd.ejs

```

<!DOCTYPE html>
<html>
    <!--Includes the defined head-->
    <%- include('../partials/head'); %>

```

```

<body>
  <!--Includes the defined header-->
  <%- include('../..partials/empHeader'); %>

  <main>
    <div class="container-fluid">
      <h1>New Product</h1>
      <!--Displays any errors-->
      <% for(let i=0; i < errors.length; i++){ %>
        <h6 style="color:red"><%= errors[i] %></h6>
      <% } %>
      <form action = "/addProduct" method = "post">
        <div class="form-group row">
          <label for="state" class="col-sm-2 col-form-
label">Category:</label>
          <div class="col-sm-10">
            <select class = "form-control" name ="catID" required>
              <option value="1" selected>Produce</option>
              <option value="2">Beverages</option>
              <option value="3">Tools</option>
            </select>
          </div>
        </div>
        <div class="form-group row">
          <label for="first" class="col-sm-2 col-form-label">Product
Name:</label>
          <div class="col-sm-10">
            <input type = "text" class = "form-control" name ="prodName"
placeholder="Placeholder" pattern="[A-Za-z\s]+" required>
          </div>
        </div>
        <div class="form-group row">
          <label for="last" class="col-sm-2 col-form-label">Price:</label>
          <div class="col-sm-10">
            <input type = "number" class = "form-control" name ="price"
placeholder="9.99" min="0" step="0.01" pattern="[0-9]+" required>
          </div>
        </div>
        <div class="form-group row">
          <label for="street" class="col-sm-2 col-form-label">Image
Name:</label>

```

```

        <div class="col-sm-10">
            <input type = "text" class = "form-control" name ="imageName"
placeholder="example.jpg" pattern="[A-Za-z.]+ " required>
        </div>
    </div>
    <div class="form-group">
        <input type="submit" value="Add Product">
    </div>
</form>
</div>
</main>

<!--Includes the defined footer-->
<%- include('../..partials/footer'); %>
</body>
</html>

```

-orderTable.ejs

```

<!DOCTYPE html>
<html>
<!--Includes the defined head-->
<%- include('../..partials/head'); %>

<body>
    <!--Includes the defined header-->
    <%- include('../..partials/empHeader'); %>

    <main>
        <div class="container-fluid">
            <h1>Order List</h1>
            <h6>(Logged in: <%= user.email %>)</h6>
            <div class="table-responsive">
                <!-- displays all orders -->
                <table>
                    <tr>
                        <th>Order ID</th>
                        <th>Order Date</th>
                        <th>Order Total</th>
                        <th>Customer Name</th>
                        <th>Customer Address</th>
                        <th>Customer ZIP Code</th>

```

```

        </tr>
        <% orders.forEach(function(order){ %>
            <tr>
                <td><%= order.orderID %></td>
                <td><%= order.orderDate %></td>
                <td><%= order.orderTotal %></td>
                <td><%= order.custFName+" "+order.custLName %></td>
                <td><%= order.custStreet+", "+order.custCity+",
"+order.custState %></td>
                <td><%= order.custZip %></td>
                <!-- displays option to remove order if CEO is logged in
-->
                <!-- but dont display any buttons if there are no orders--
>

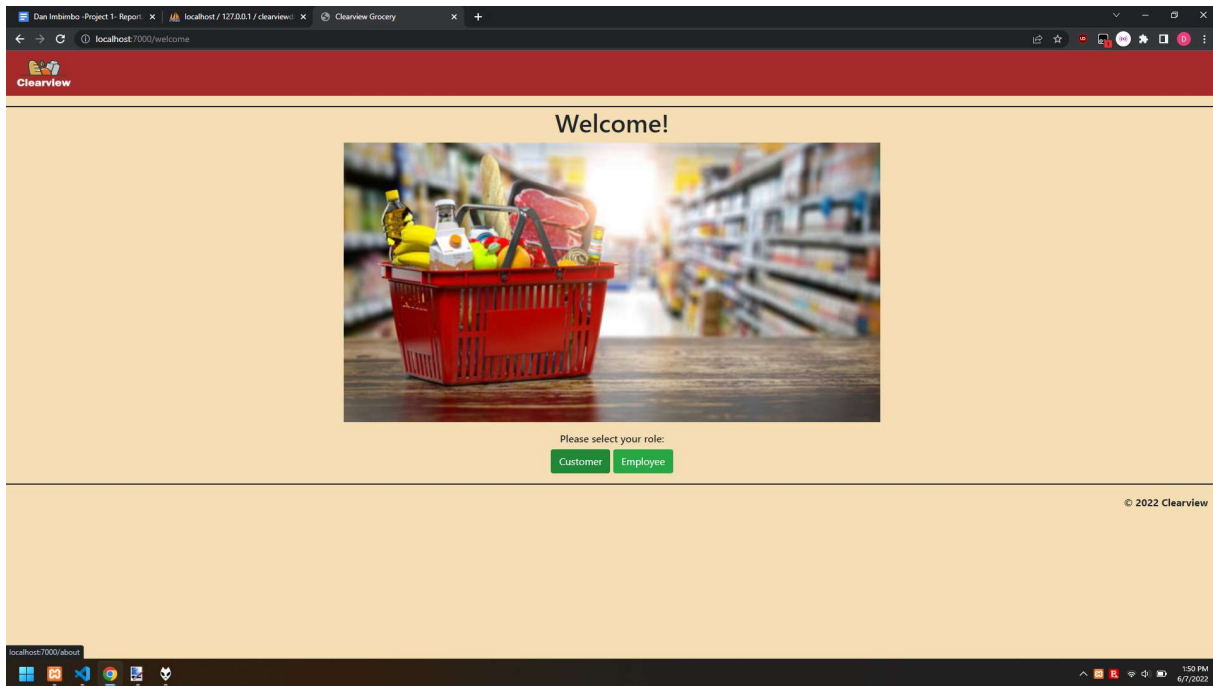
                <% if(user.deptID == 1){ %>
                    <td>
                        <form action="/removeOrder" method="post">
                            <input type="hidden" name="orderID"
                                value="<%= order.orderID %>">
                            <input type="submit" value="Remove">
                        </form>
                    </td>
                <% } %>
            </tr>
        <% }) %>
    </table>
</div>
</div>
</main>

<!--Includes the defined footer-->
<%- include('../..partials/footer'); %>
</body>
</html>

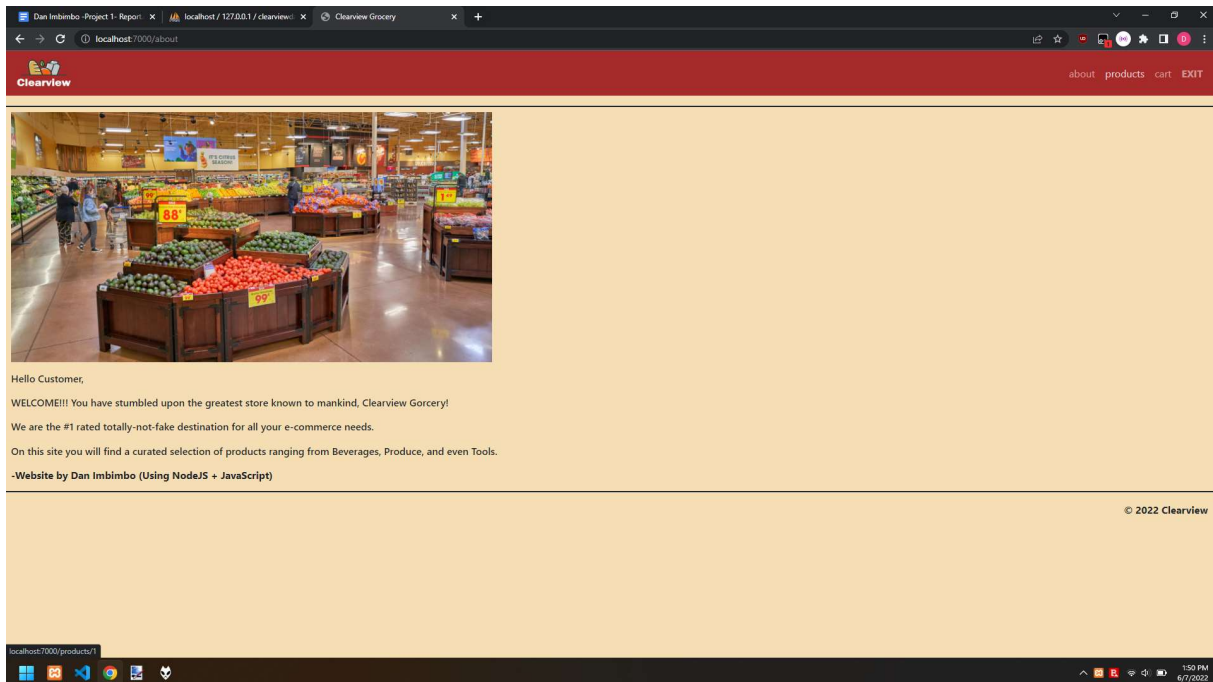
```


Test run:

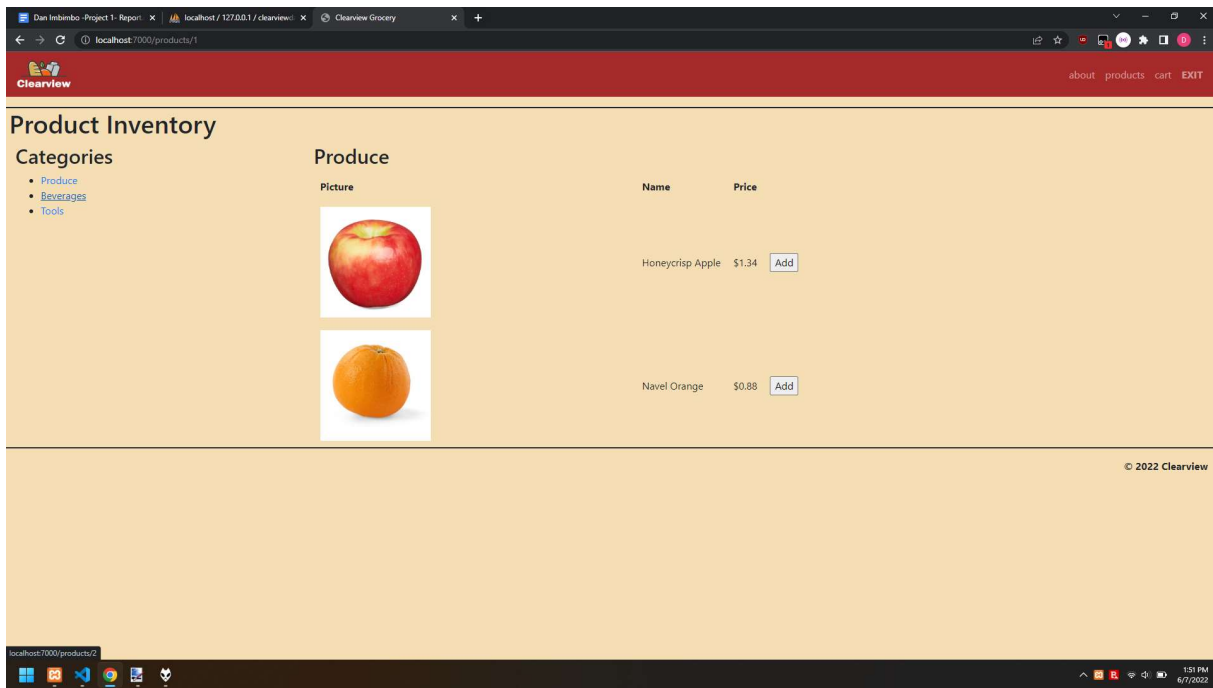
Initial welcome page which lets user choose between user or employee views:



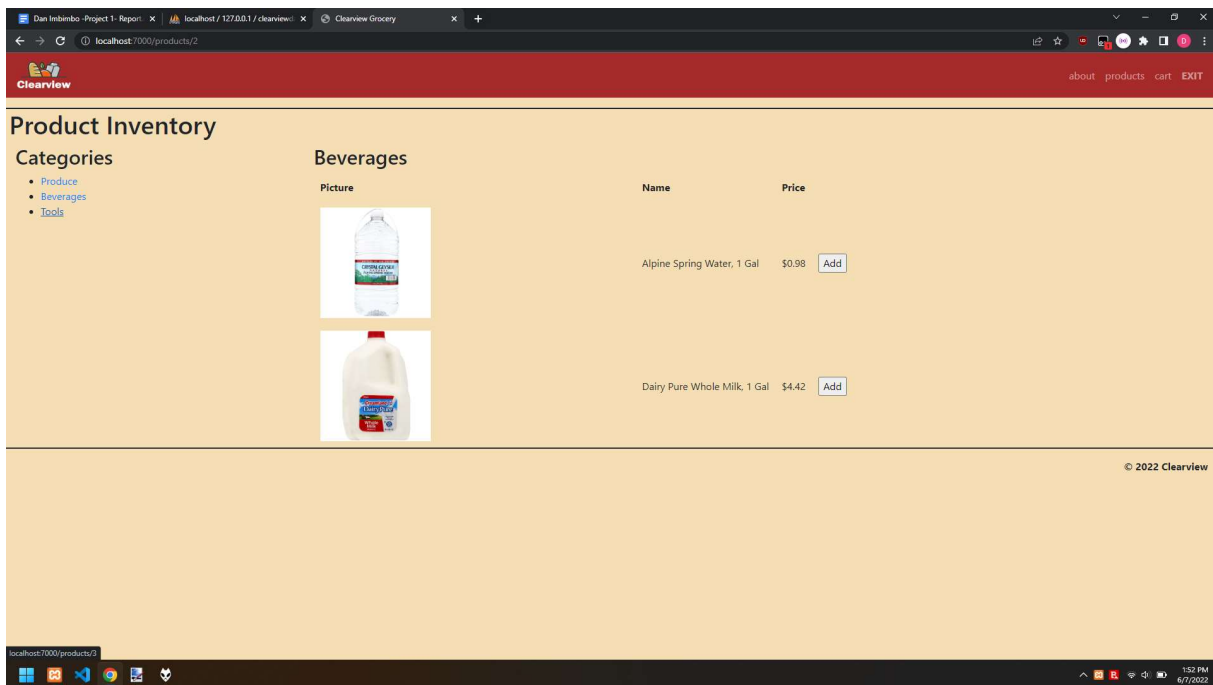
About page and using the navigation bar to go to products:



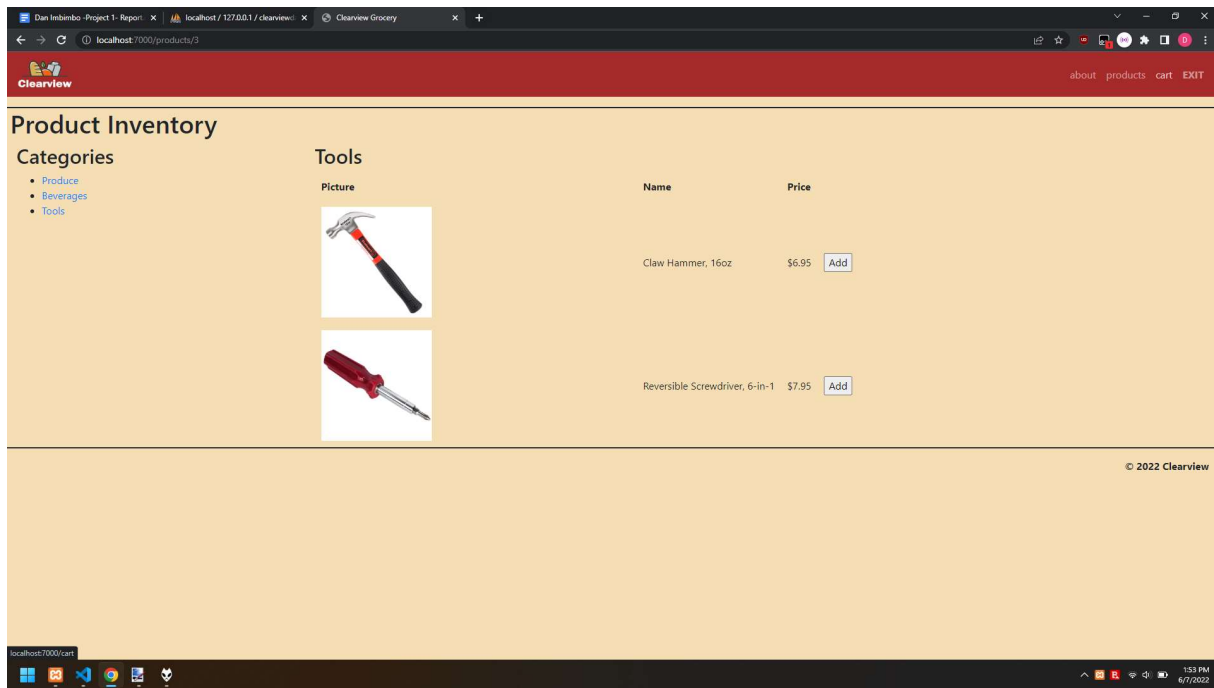
Products page showing the Produce category and navigating to Beverages:



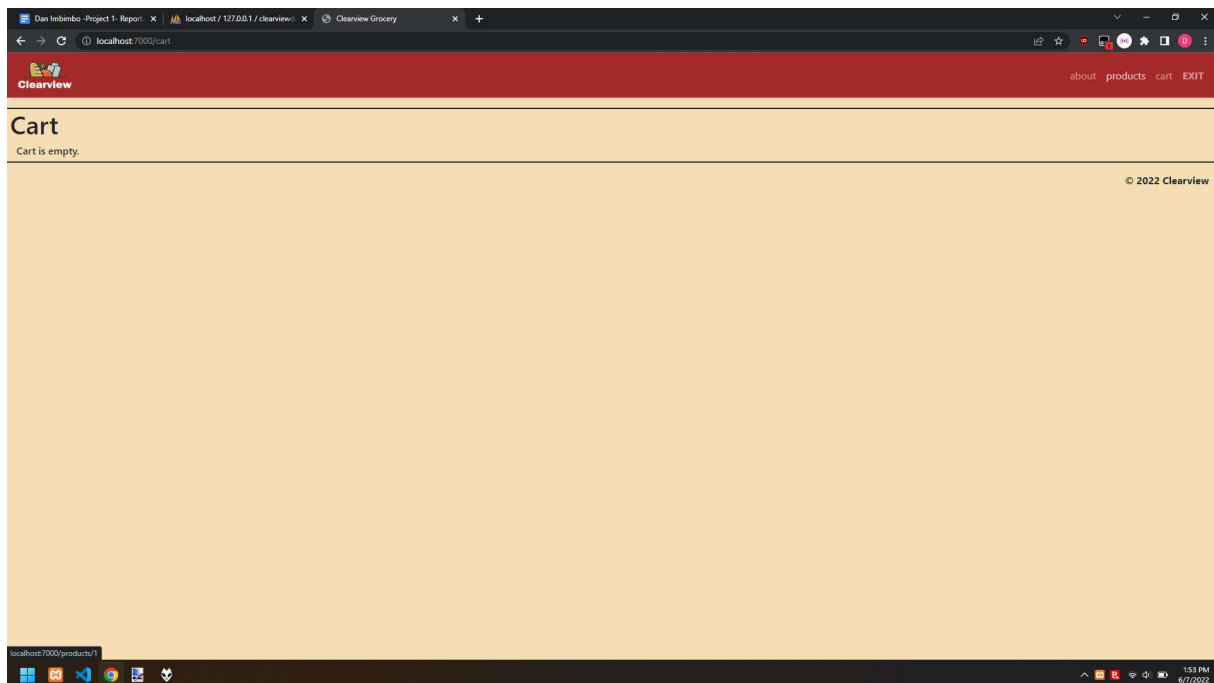
Products page showing the Beverages category and navigating to Tools:



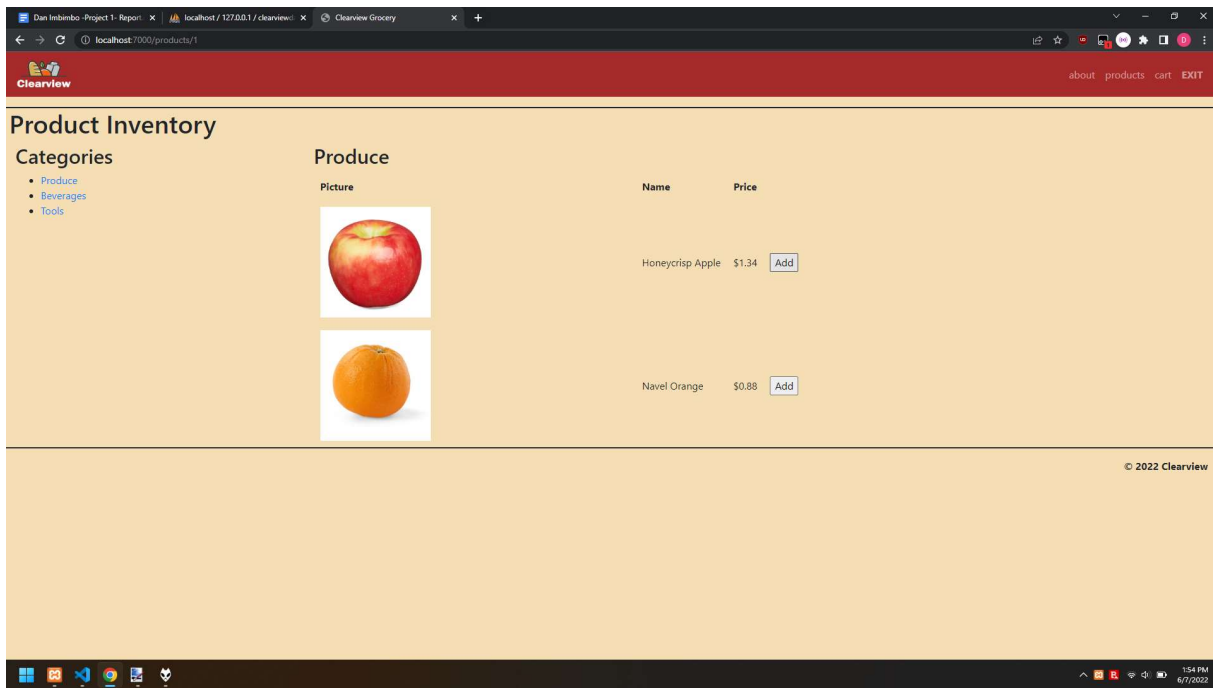
Products page showing the Tools category and navigating to the cart:



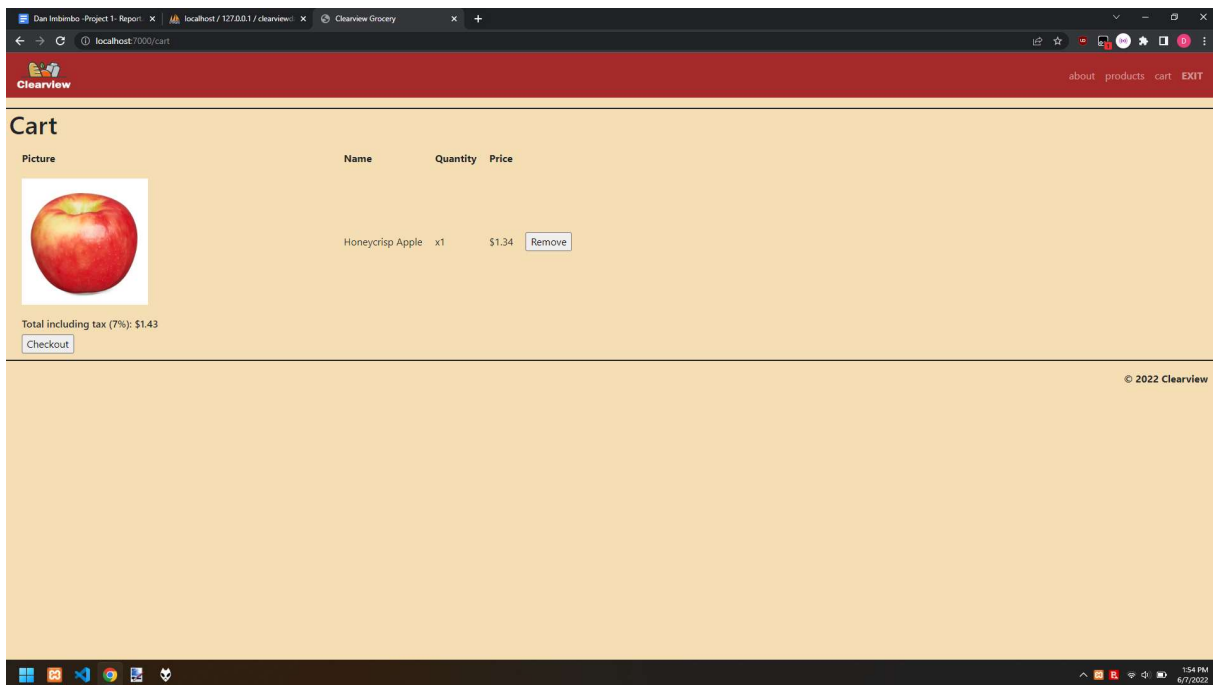
Cart page showing what an empty cart looks like and navigating back to products:



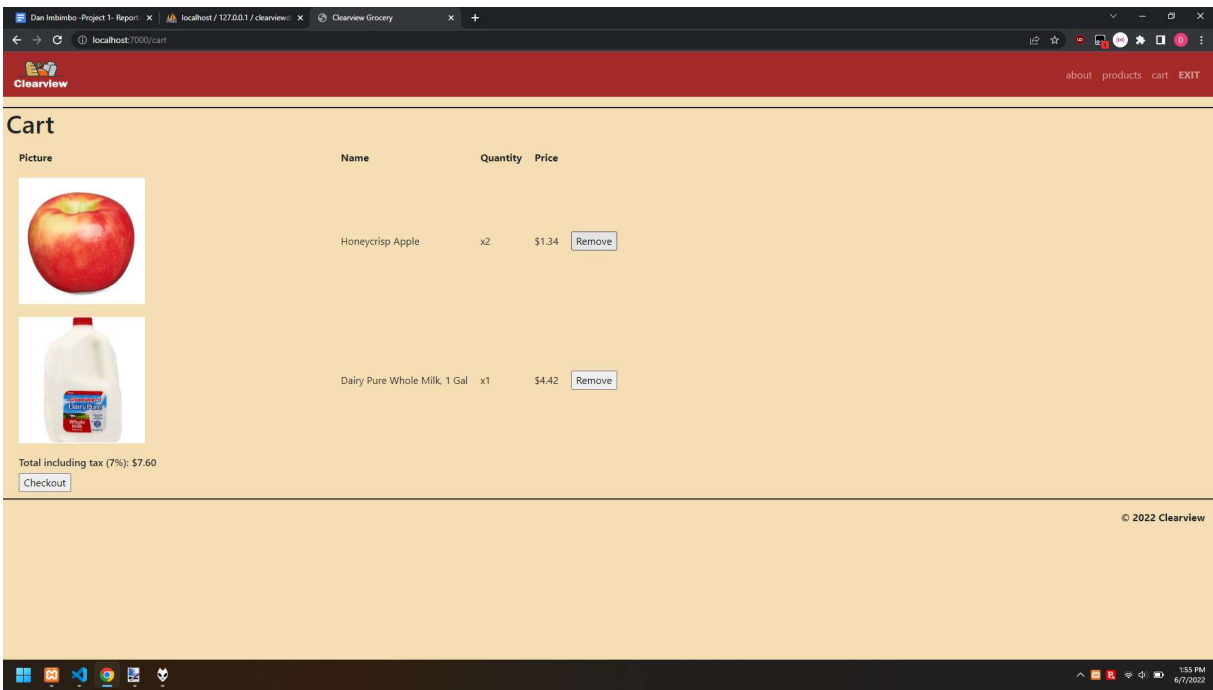
Products page while adding a Honeycrisp Apple to the cart:



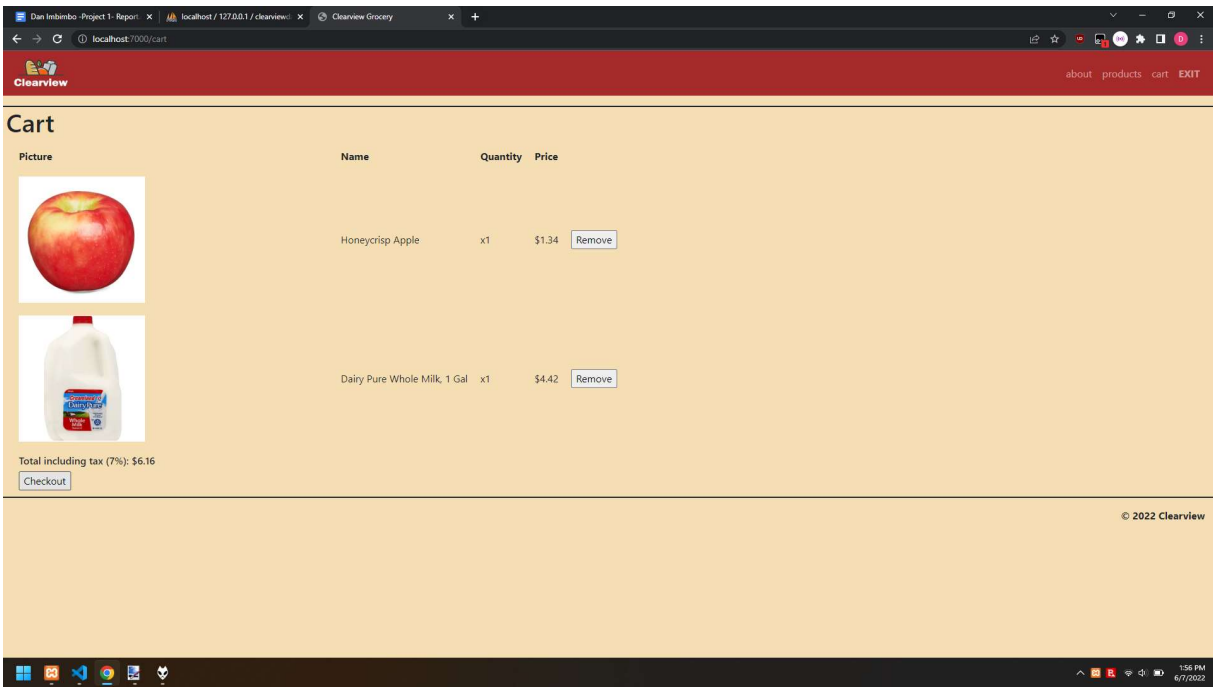
Cart page showing the added item:



Cart page after adding another apple and milk, and about to remove 1 apple:



Cart page after having removed 1 apple, and about to checkout:



Checkout form (with appropriate input patterns for each field) and about to place order:

The screenshot shows a web browser window with the URL `localhost:7000/checkout`. The page has a red header with the 'Clearview' logo and navigation links for 'about', 'products', 'cart', and 'EXIT'. The main content area is titled 'Checkout' and contains a form with the following fields and values:

- Name**
 - First Name: Bob
 - Last Name: McGee
- Address**
 - Street: 123 Anywhere Ave
 - City: Some City
 - State: (NJ) New Jersey (dropdown menu)
 - Zip Code: 01234
 - Card Number: 1111 2222 3333 4444

A 'Place Order' button is located at the bottom of the form. The footer of the page displays '© 2022 Clearview'.

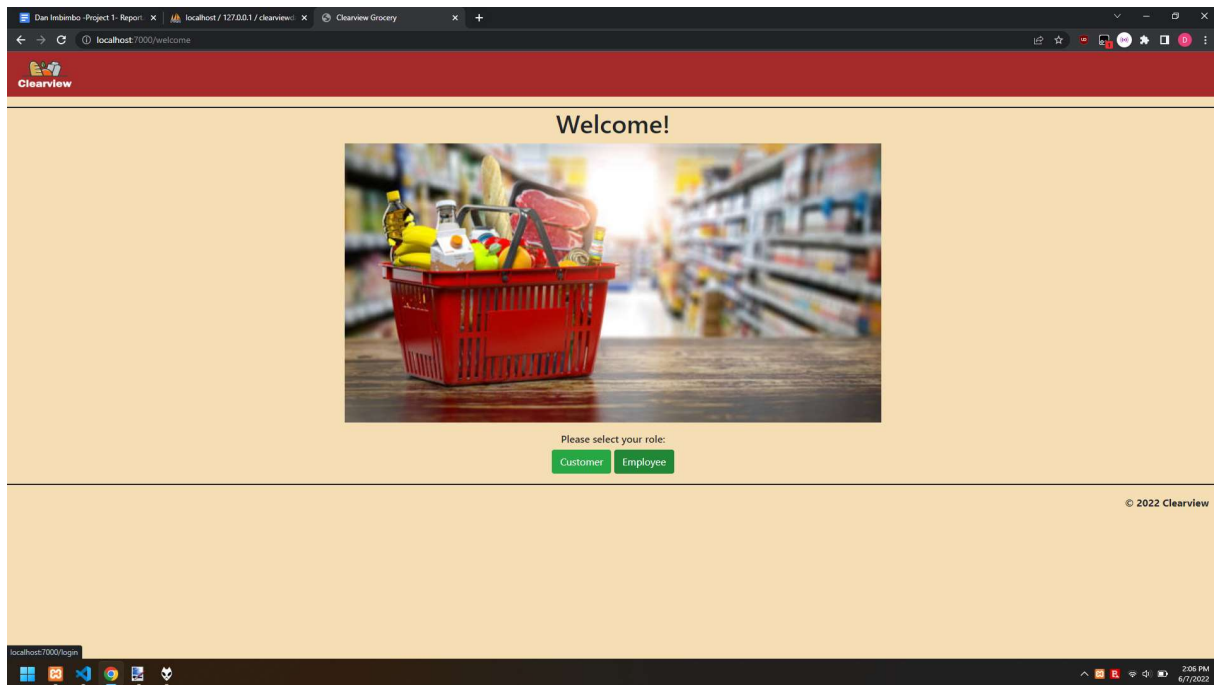
Checkout form after having placed the order with correct input:

This screenshot shows the same checkout form as the previous one, but with a dark grey modal overlay in the center. The modal contains the following text:

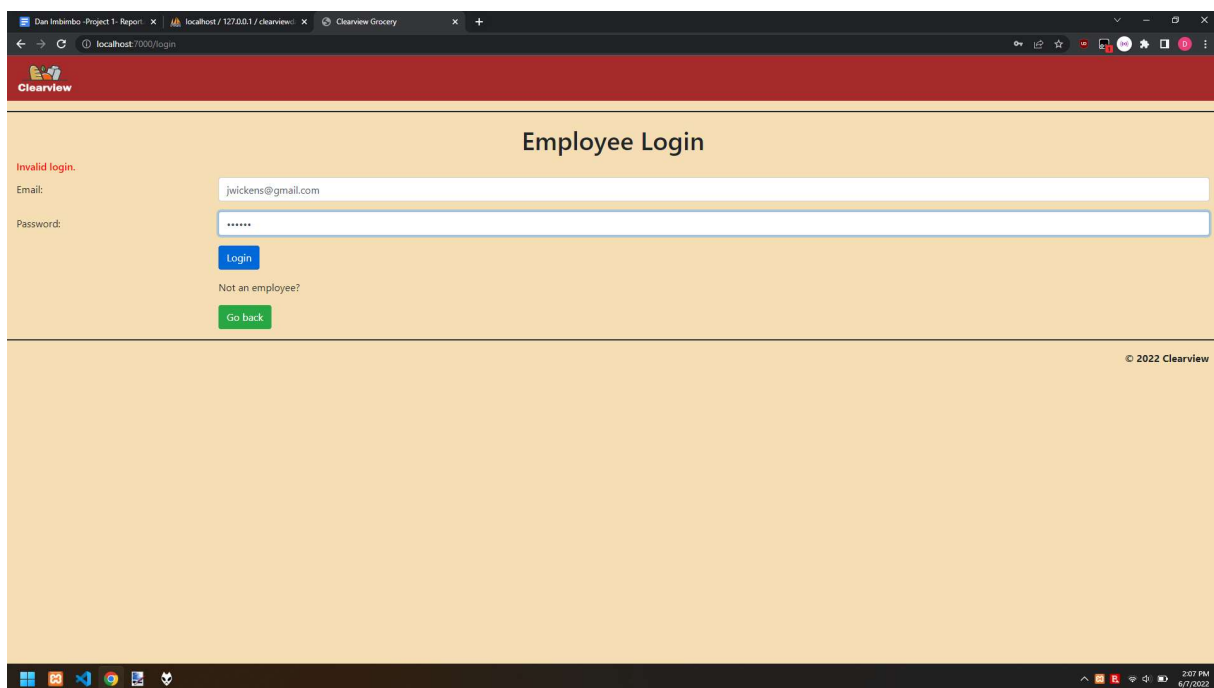
localhost:7000 says
Your order has been placed. Thank you!

An 'OK' button is located at the bottom right of the modal. The form fields and the 'Place Order' button remain visible behind the modal. The rest of the page, including the header and footer, is identical to the previous screenshot.

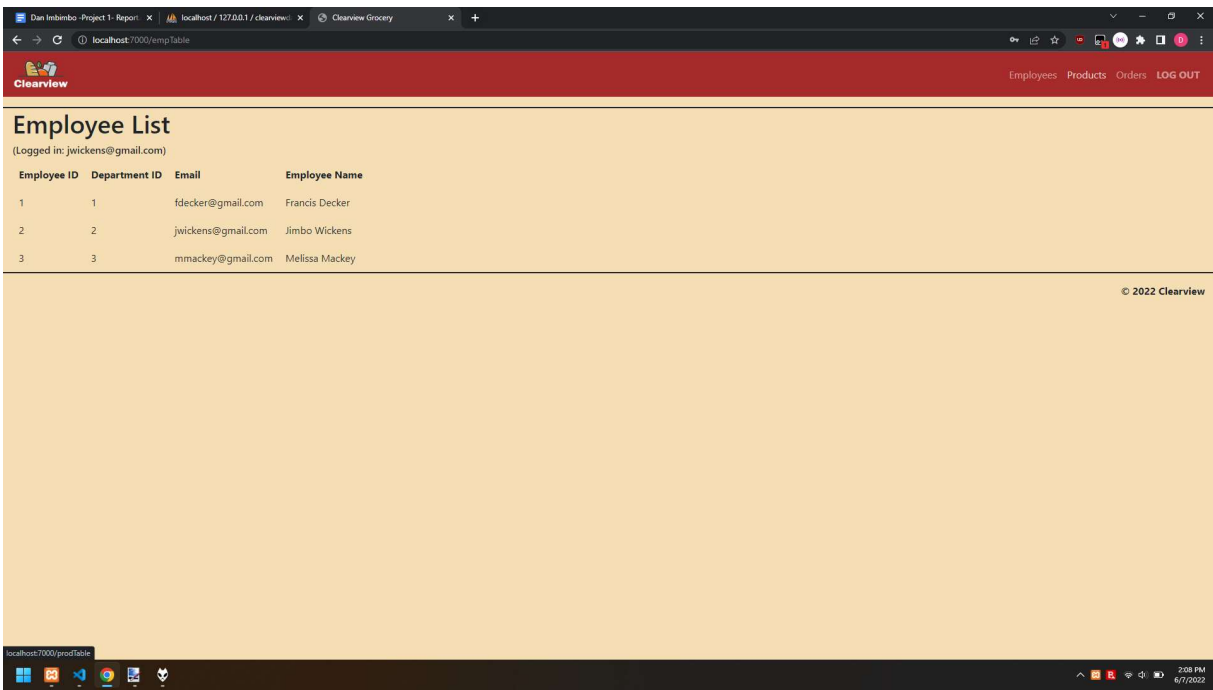
Returned to welcome page with the session cart and running total being reset, and now about to click 'Employee' button:



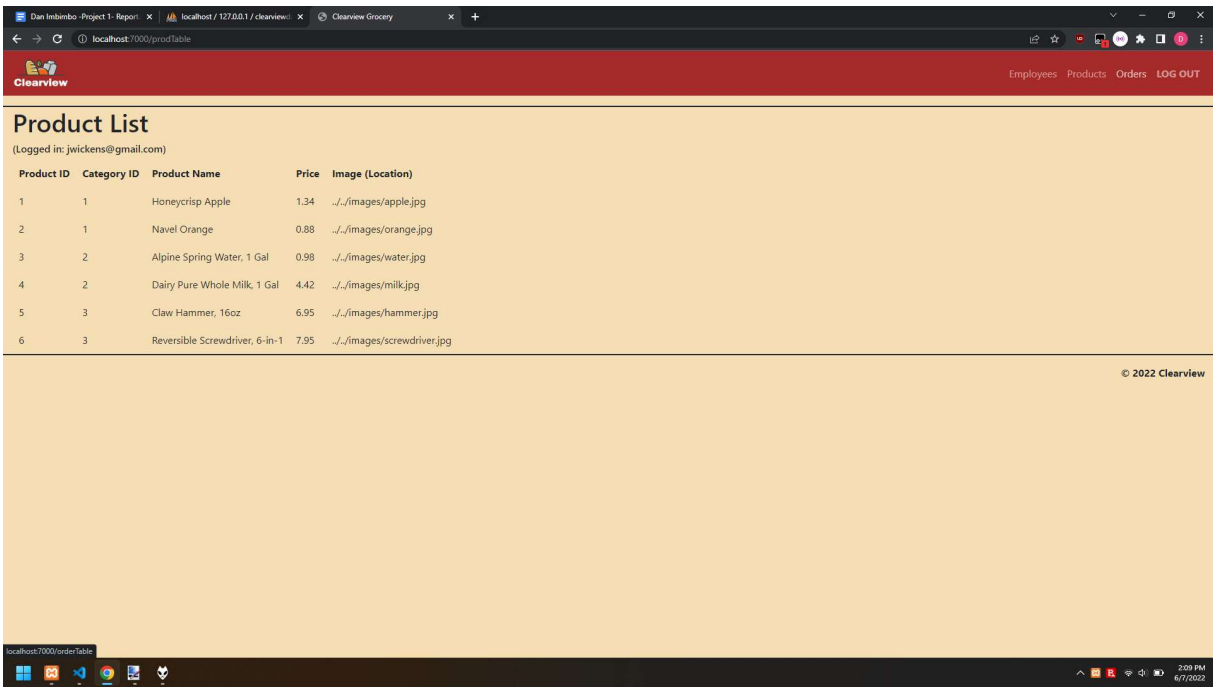
Invalid login message from incorrect info, and now about to login with correct details:



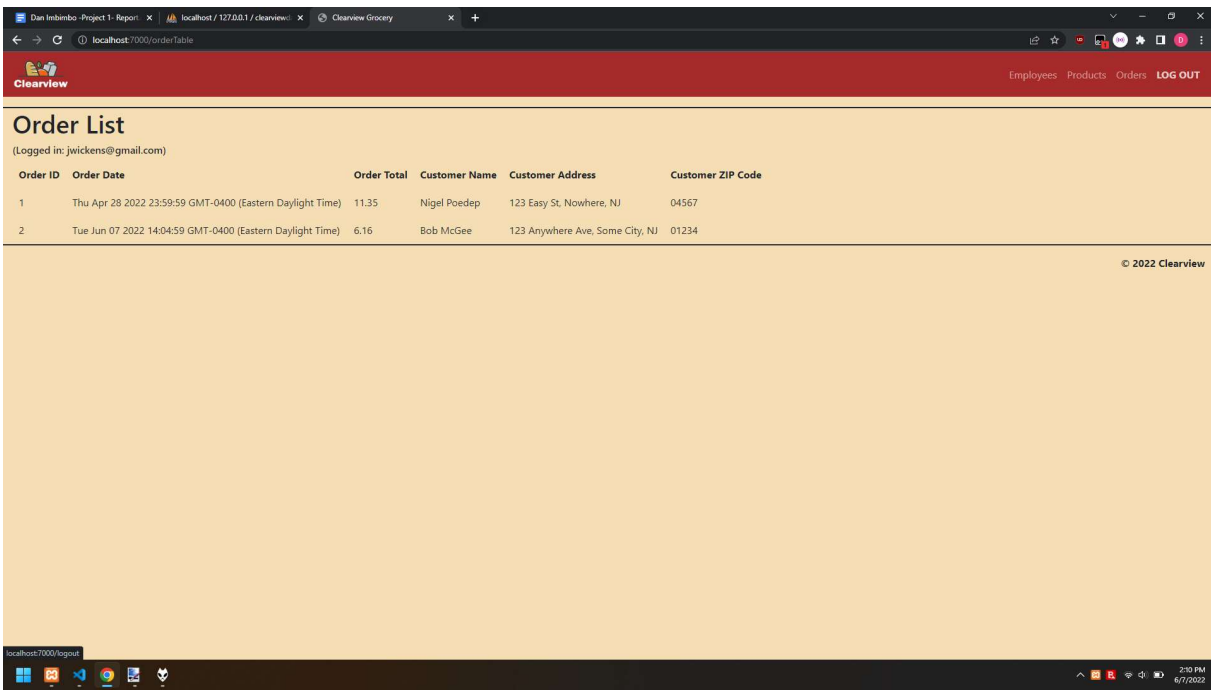
Successful login displaying the employee list for normal employees and about to navigate to Products:



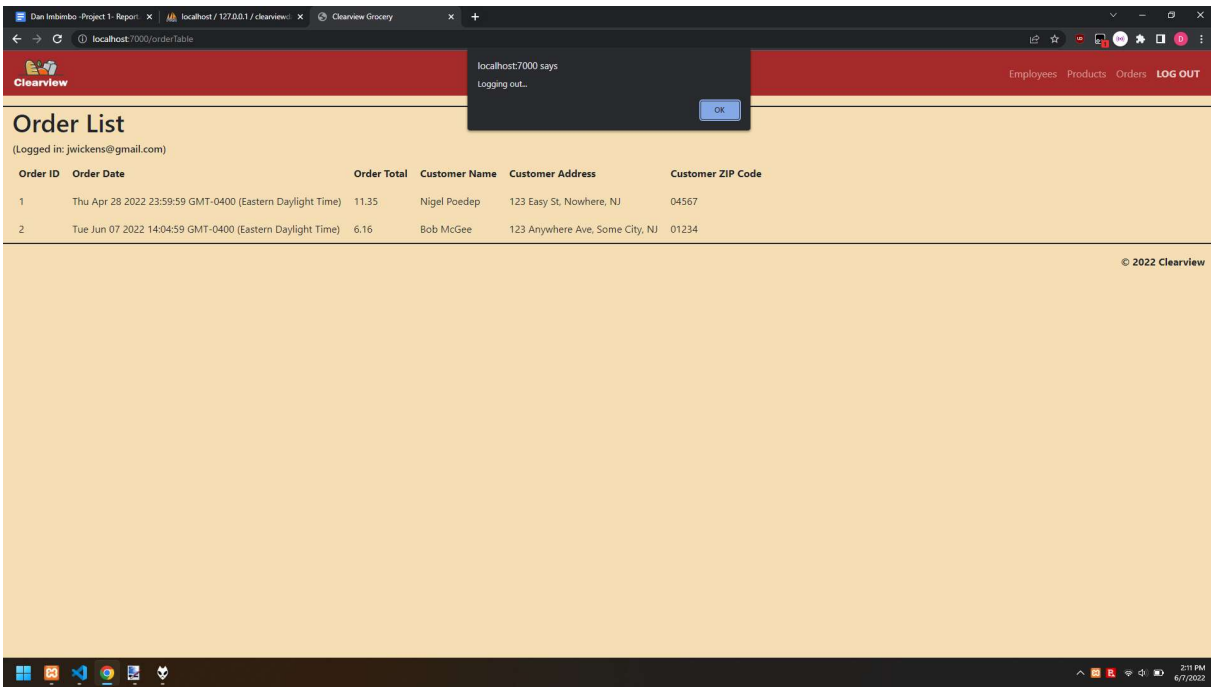
Product list displayed for normal employees and about to navigate to Orders:



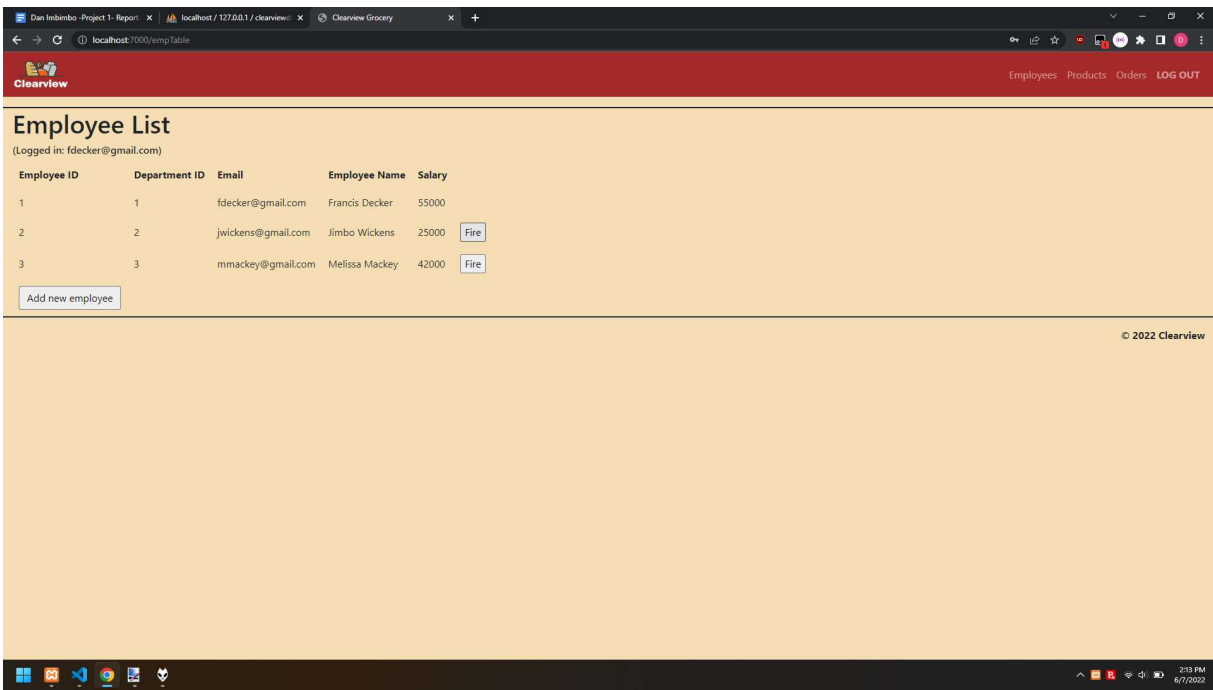
Order list displayed for normal employees and about to log out:



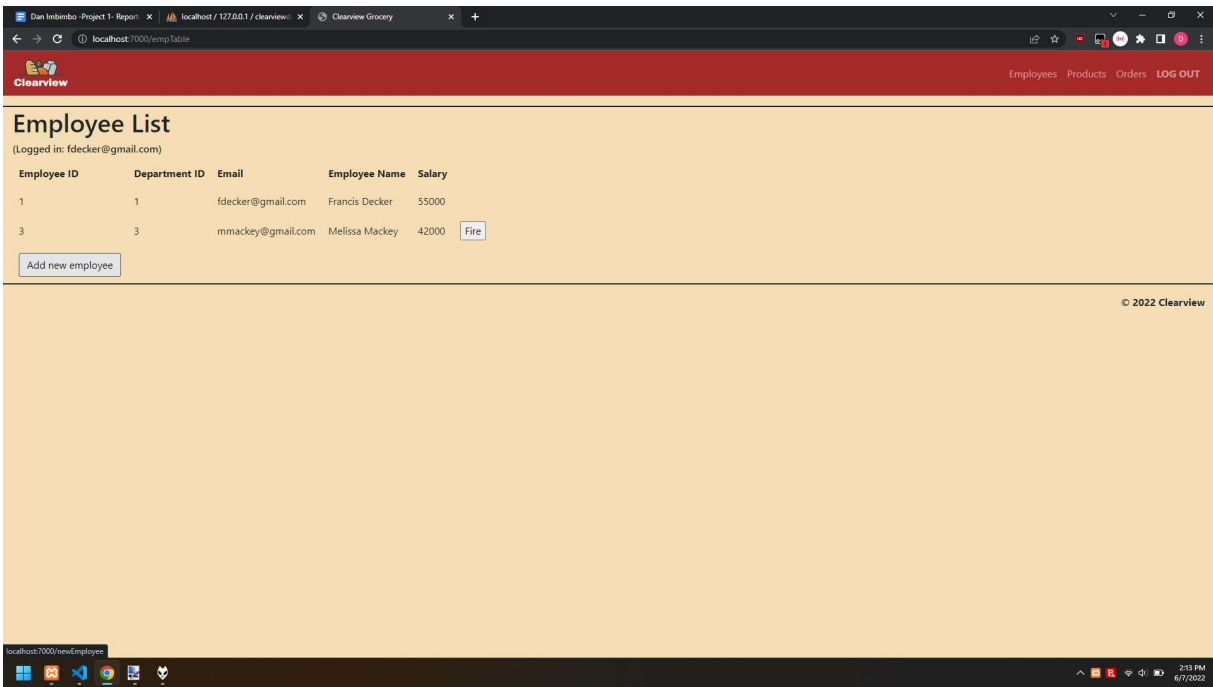
Logging out and terminating the previous session (returns to welcome page):



Employee list displayed after now logging in as the CEO and about to fire Jimbo Wickens:



Employee list after Jimbo Wickens has been fired and a new employee is about to be hired:



New employee form (with appropriate input patterns for each field) after attempting to hire the employee with an existing email address (fdecker@gmail.com), now about to hire John Smith with a new one:

New Employee
Email address already exists in the database.

First Name: John

Last Name: Smith

Salary: 20000

Department: Customer Service

Email: jsmith@gmail.com

Password: *****

Hire Employee

© 2022 Clearview

New employee John Smith shown in Employee List and about to navigate to Products:

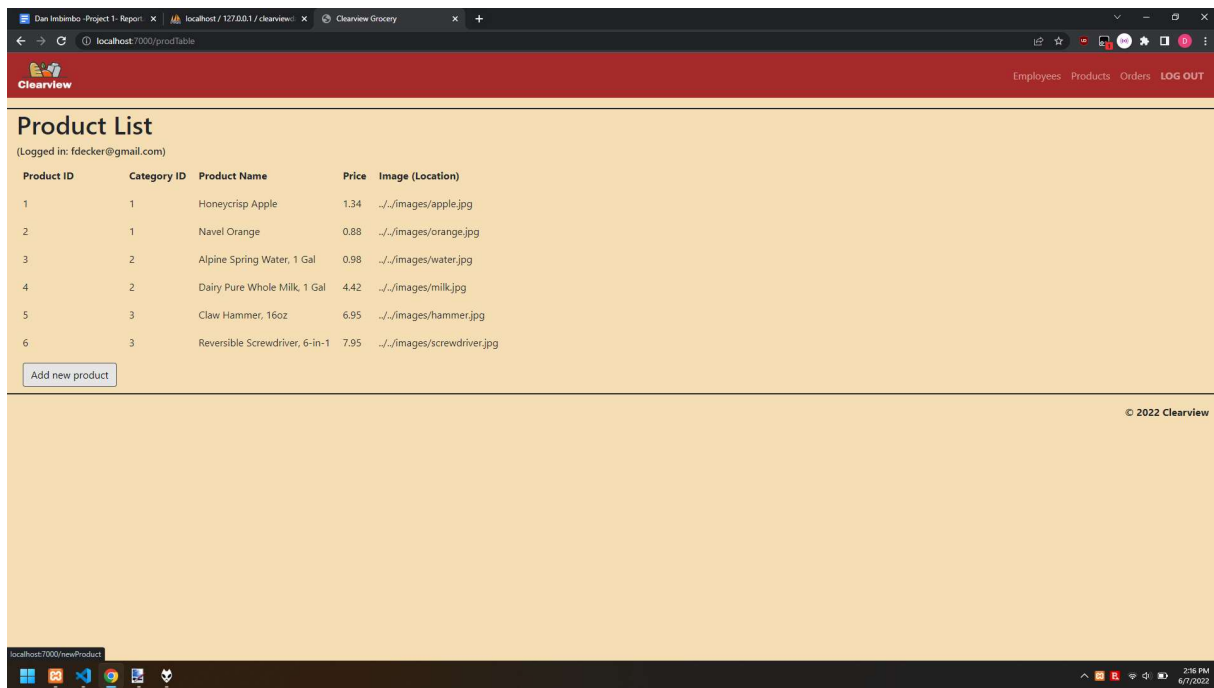
Employee List
(Logged in: fdecker@gmail.com)

Employee ID	Department ID	Email	Employee Name	Salary
1	1	fdecker@gmail.com	Francis Decker	55000
3	3	mmackey@gmail.com	Melissa Mackey	42000 <button>Fire</button>
4	2	jsmith@gmail.com	John Smith	20000 <button>Fire</button>

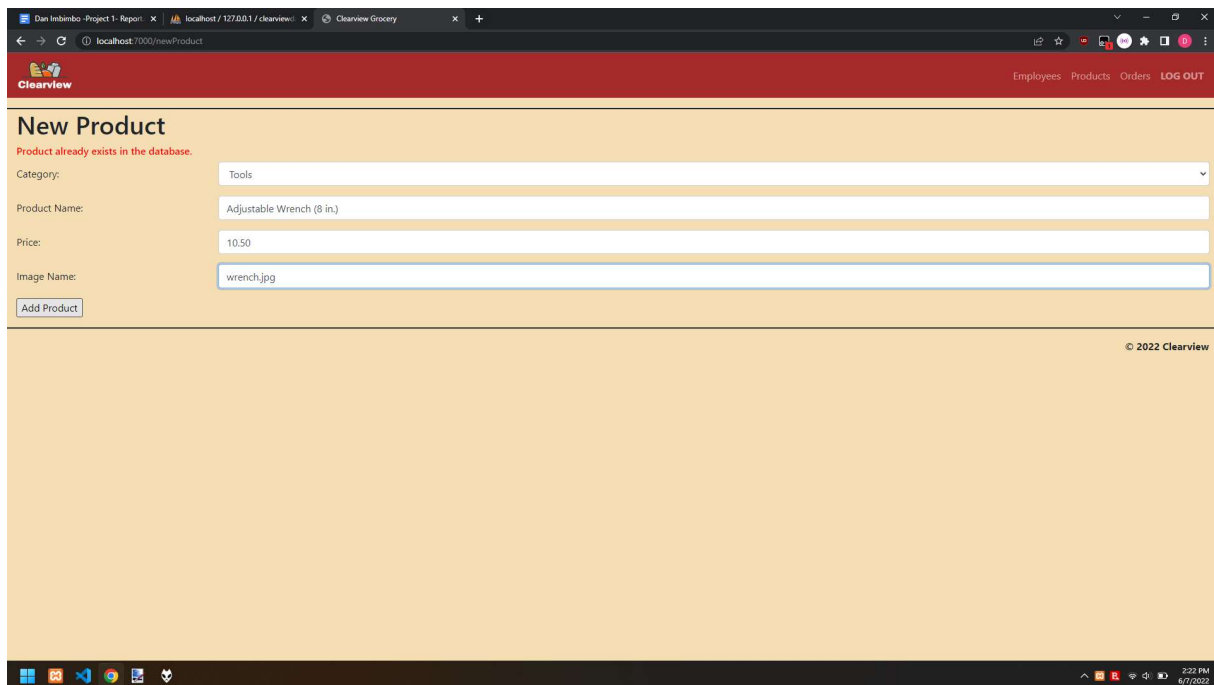
Add new employee

© 2022 Clearview

Product list displayed for CEO with ability to add new products:



New product form (with appropriate input patterns for each field) after attempting to add an existing product (Honeycrisp Apple), now about to add Adjustable Wrench to product list:



New product form with a Pear about to be added to product list:

New Product

Category:

Product Name:

Price:

Image Name:

© 2022 Clearview

Product List displayed with both new products and about to navigate to Orders:

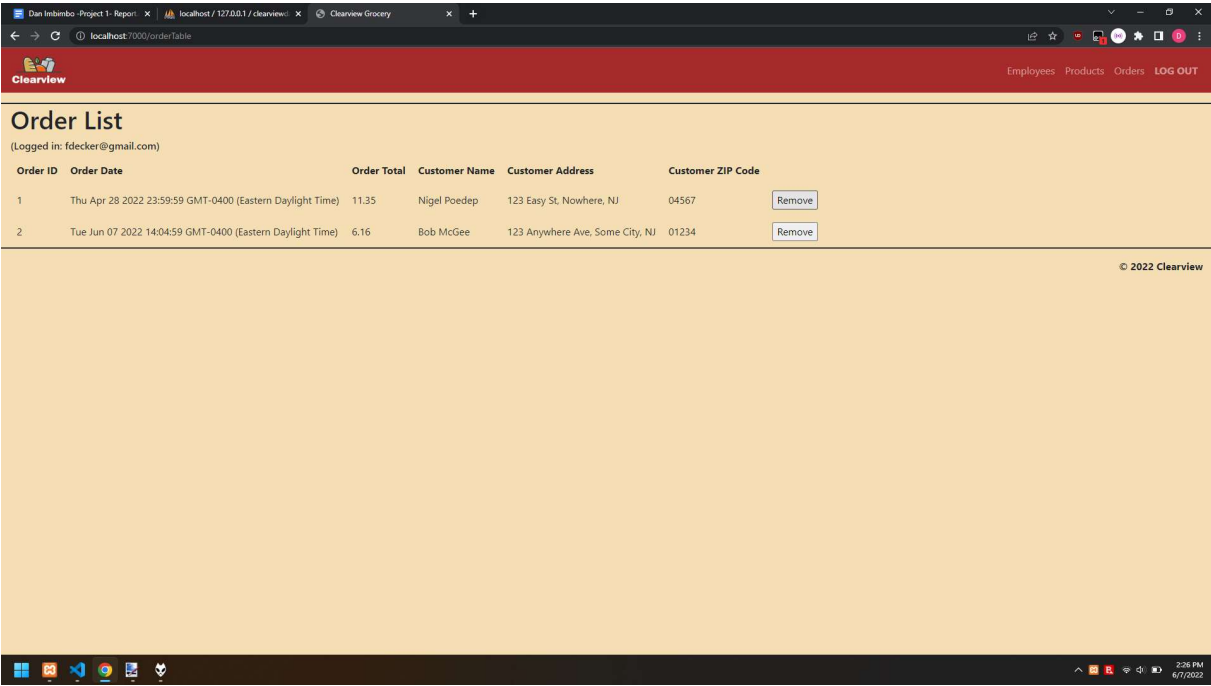
Product List

(Logged in: fdecker@gmail.com)

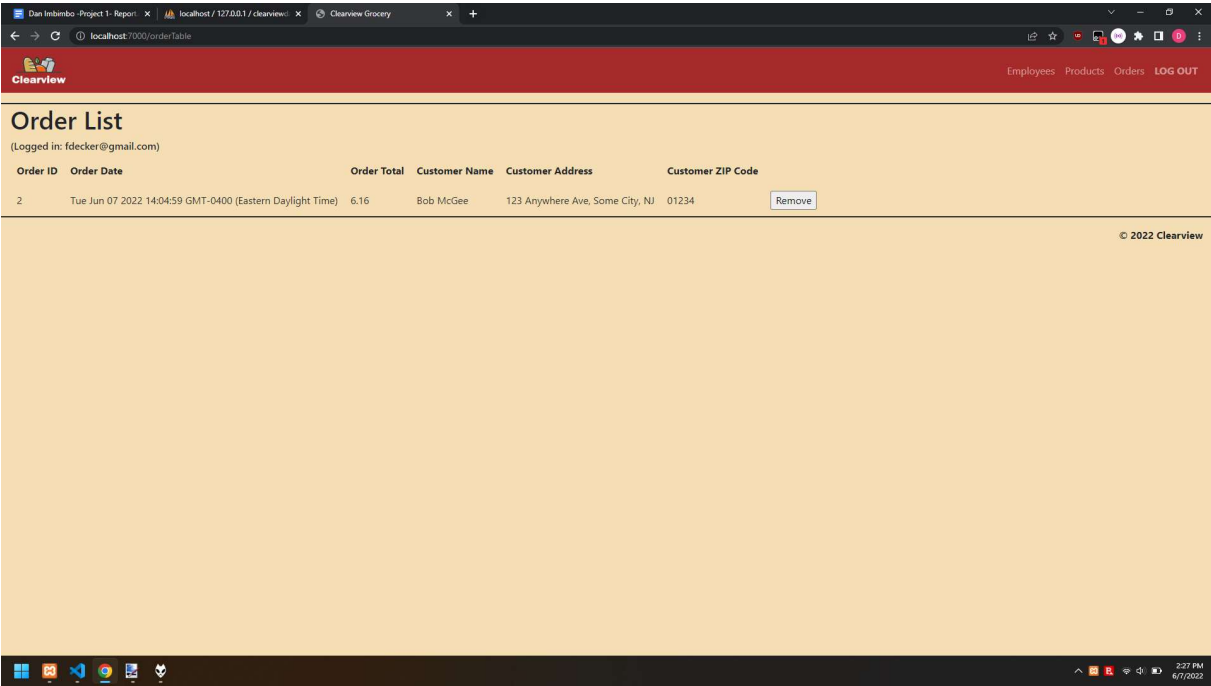
Product ID	Category ID	Product Name	Price	Image (Location)
1	1	Honeycrisp Apple	1.34	../images/apple.jpg
2	1	Navel Orange	0.88	../images/orange.jpg
3	2	Alpine Spring Water, 1 Gal	0.98	../images/water.jpg
4	2	Dairy Pure Whole Milk, 1 Gal	4.42	../images/milk.jpg
5	3	Claw Hammer, 16oz	6.95	../images/hammer.jpg
6	3	Reversible Screwdriver, 6-in-1	7.95	../images/screwdriver.jpg
7	3	Adjustable Wrench (8 in.)	10.5	../images/wrench.jpg
8	1	Pear	1.12	../images/pear.jpg

© 2022 Clearview

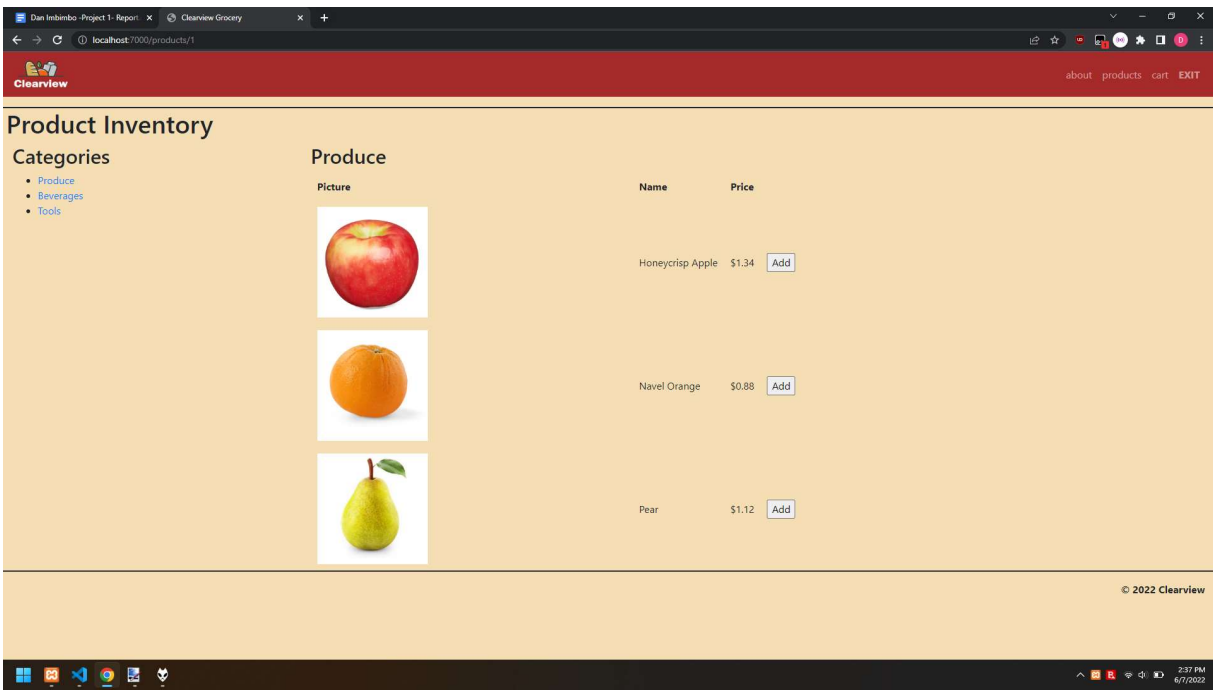
Order List displayed for CEO with ability to remove order and about to remove a previously placed order:



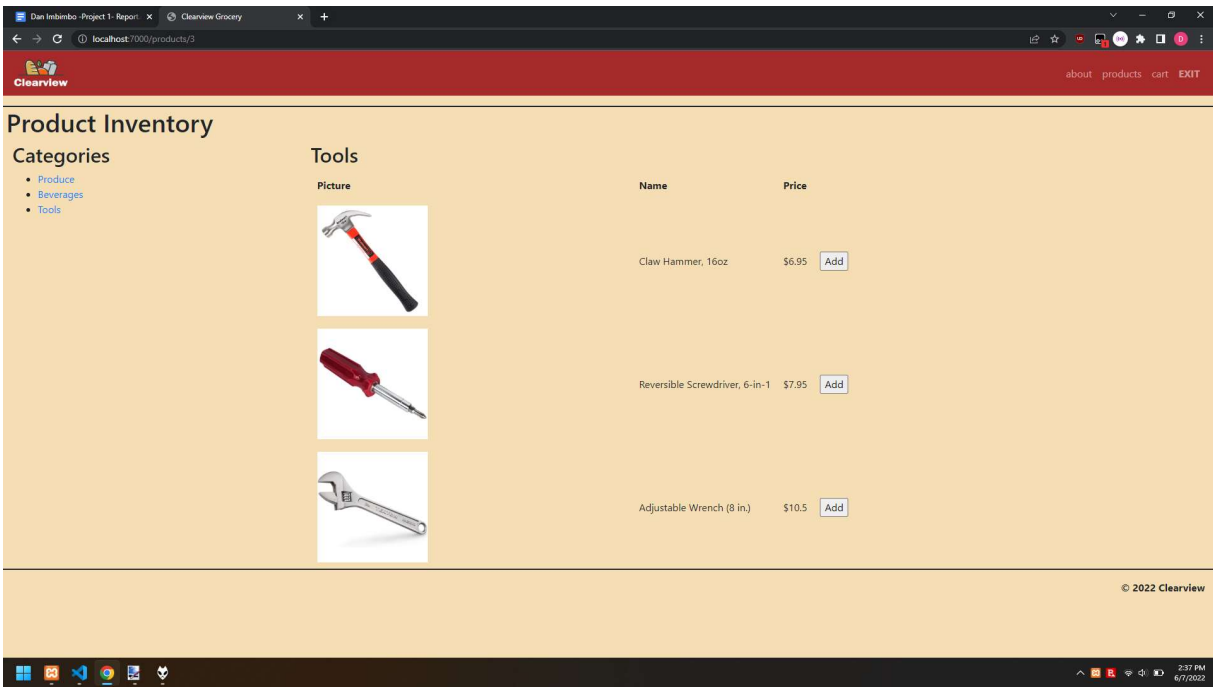
Order List after having deleted an order:



Pear now displayed in Customer view of product list:



Wrench now displayed in Customer view of product list:



References

- Eernisse, Matthew. "Embedded JavaScript Templates." *Npm*, <https://www.npmjs.com/package/ejs>.
- Hanson, Jared, and Peter Wooley. "Connect-Flash: Flash Message Middleware for Connect and Express." *GitHub*, <https://github.com/jaredhanson/connect-flash>.
- Macharia, Catherine. "Session Management in Node.js Using ExpressJS and Express Session." *Section*, 2 June 2021, <https://www.section.io/engineering-education/session-management-in-nodejs-using-expressjs-and-express-session/>.
- Tribhu, Jodiss. "Build a Login System in Node.js Using Passport.js and Mysql." *Medium*, Medium, 24 Jan. 2021, <https://jodiss-tri.medium.com/build-a-login-system-in-node-js-using-passport-js-and-mysql-52667cf3cc40>.
- "Use EJS as Template Engine in Node.js." *GeeksforGeeks*, 9 July 2021, <https://www.geeksforgeeks.org/use-ejs-as-template-engine-in-node-js/>.
- Wilson, Douglas. "Express-Session." *Npm*, <https://www.npmjs.com/package/express-session>.