Read Me, Approach and Schema

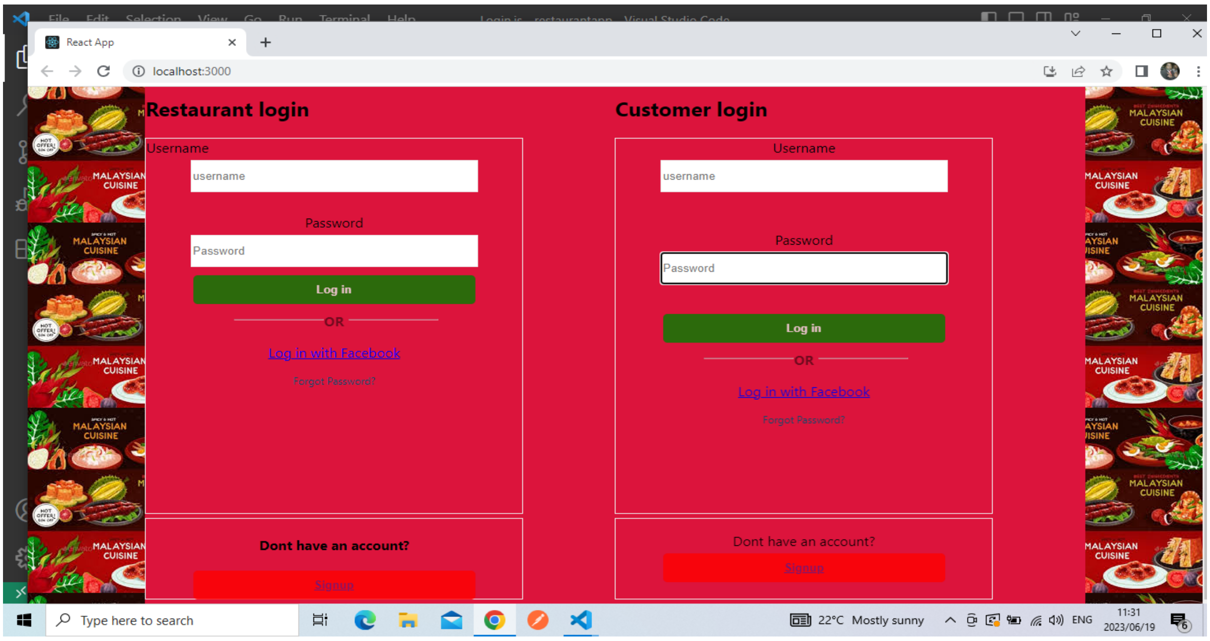
How run the program

1. Run cd orderfood
2. Run npm start
3. If not logged you are taken to loggin. If not registered you signup.
4. If signed in you can access the restaurant list. If you click on the link, you are taken to a list of pictures of different restaurant.
5. If you click on any picture of a restaurant, you are presented with a list of food items that you can add to cart with an add button.
6. As you add to cart the total is shown at the bottom.
7. The administrator can also add items through the add product link and also add picture of the product using the upload button.

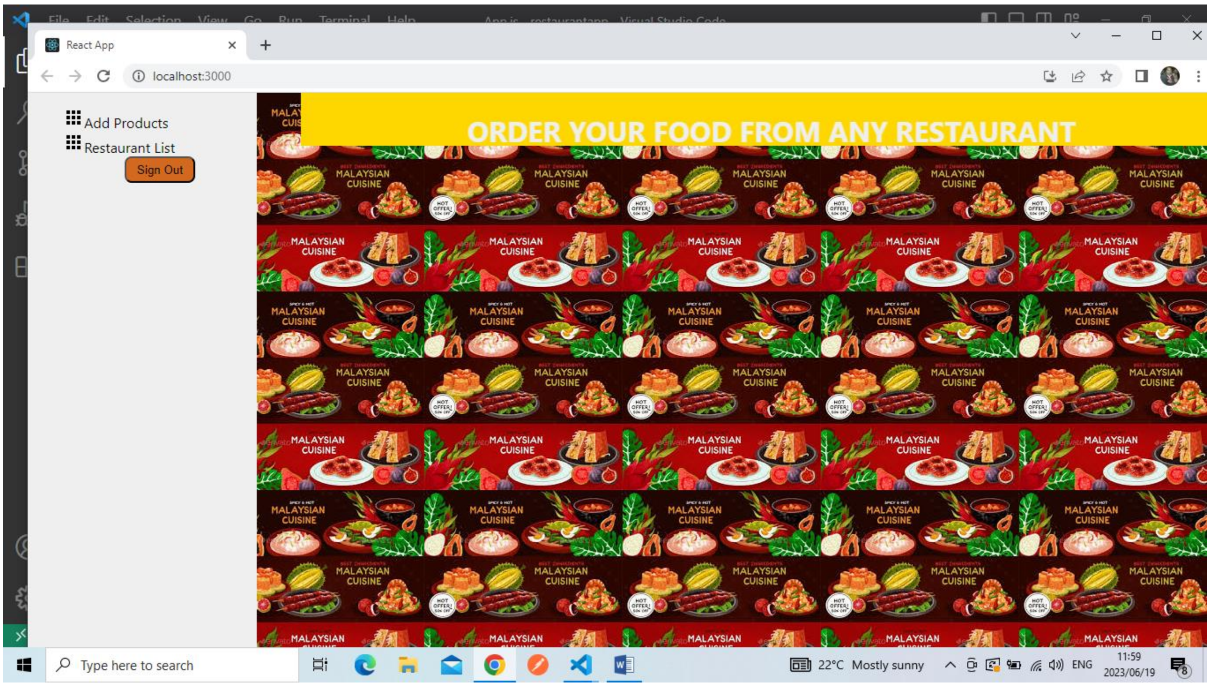
Approach

1. I started by desining the login and signup screens for both the customers and administrators
2. I then went on to design the landing apps which include the side navigation bar with links that takes me to the add product list and restaurant list.
3. I later on designed the foodlist that has add or remove functions. It can also calculate the total cost of the food bought. I also got pictures of food items and restaunts from the internet.
4. I later designed add product list using a table that I got from material(mui). I also added an upload box that can upload pictures of the items.
5. Afterwards I linked the login, signup and upload with the backend and as well as the mongoDb.
6. I also designed the two schemas for the products to be uploaded and the other one for users.

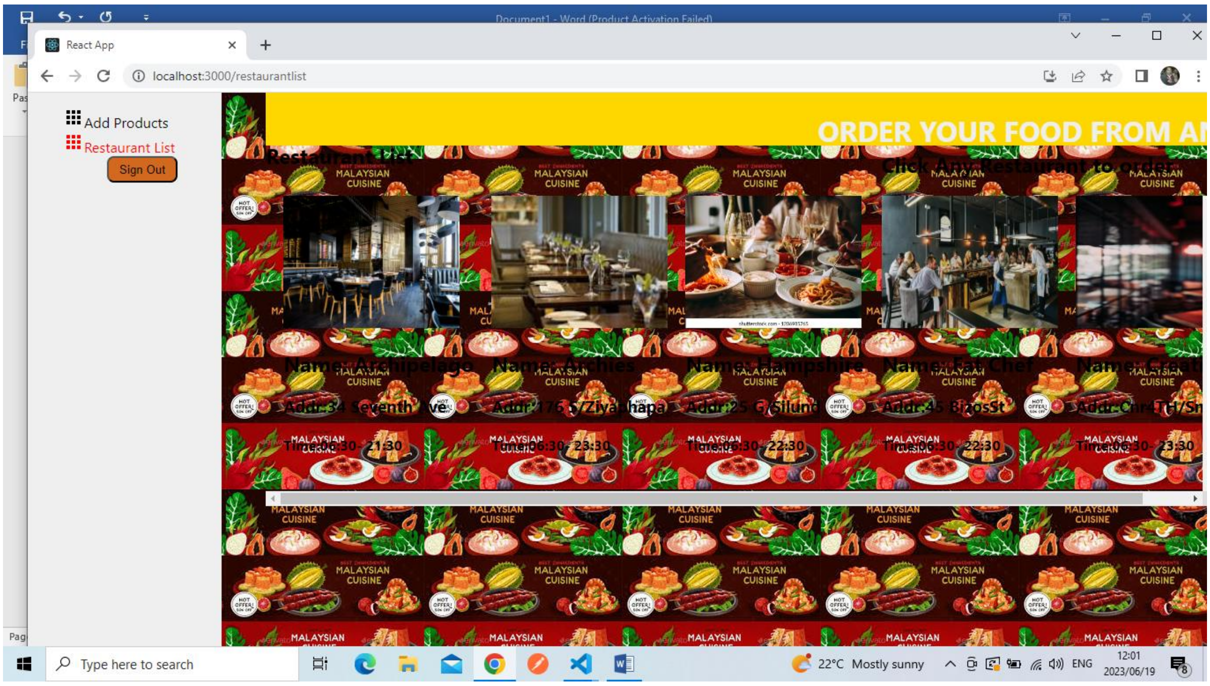
Login Screen



Home Screen



Restaurant list



Restaurant List

Click Any Restaurant to order

Name: Archipelago
Addr:34 Seventh Ave
Time:06:30- 21:30

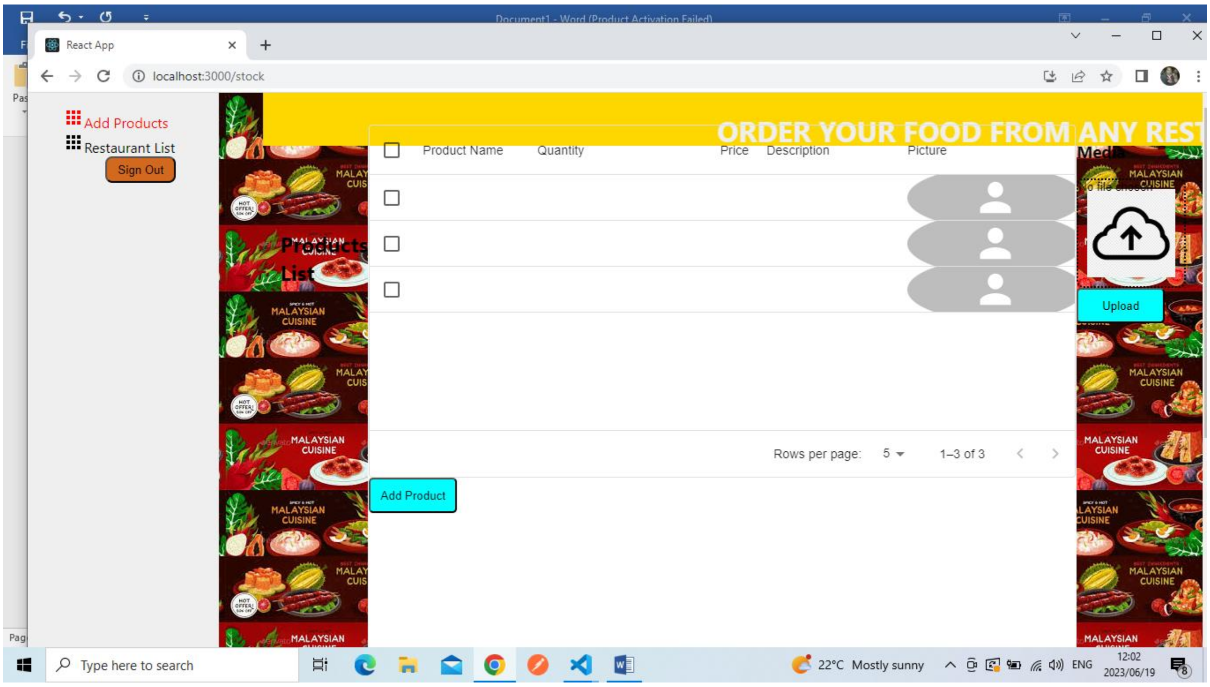
Name: Archies
Addr:176 S/Ziyaphapa
Time:06:30- 23:30

Name: Hampshire
Addr:25 G/Silund
Time:06:30- 22:30

Name: Fat Chef
Addr:45 BizosSt
Time:06:30- 22:30

Name: Creat
Addr:Chr4TH/Sm
Time:06:30- 23:30

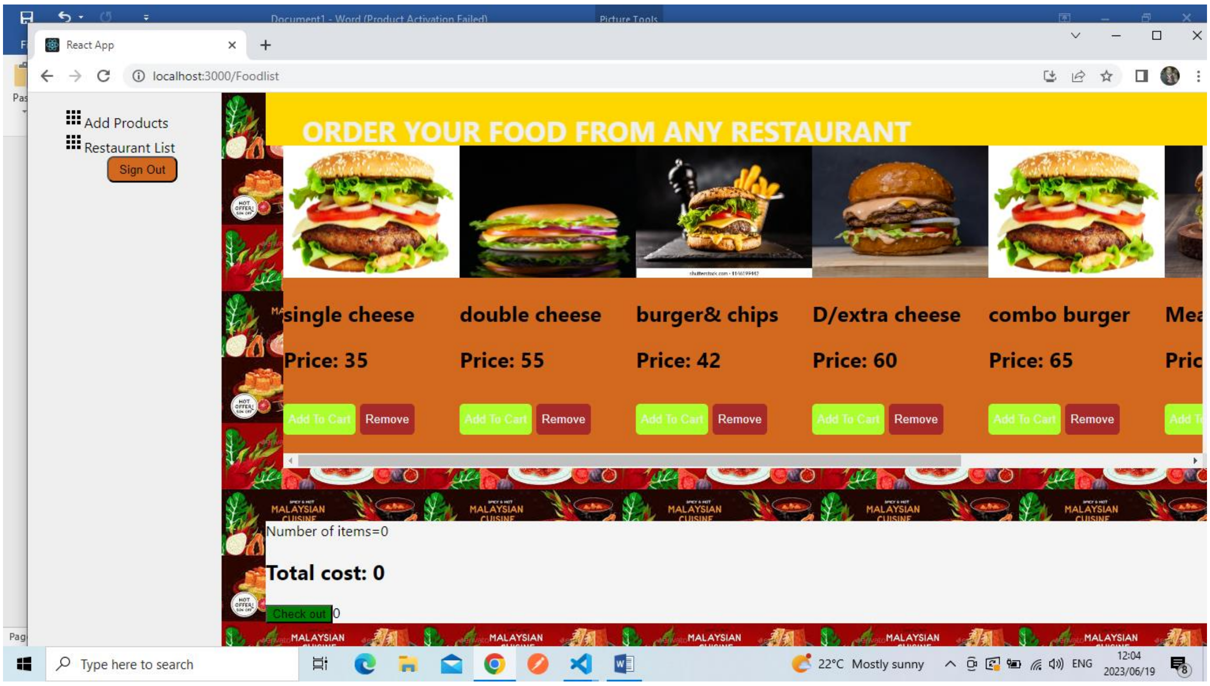
Add Product list and Upload box and buttons



□	Product Name	Quantity	Price	Description	Picture
□					
□					
□					

Rows per page: 5 1–3 of 3

Foodlist for one of the restaurants



ORDER YOUR FOOD FROM ANY RESTAURANT

single cheese
Price: 35

double cheese
Price: 55

burger& chips
Price: 42

D/extra cheese
Price: 60

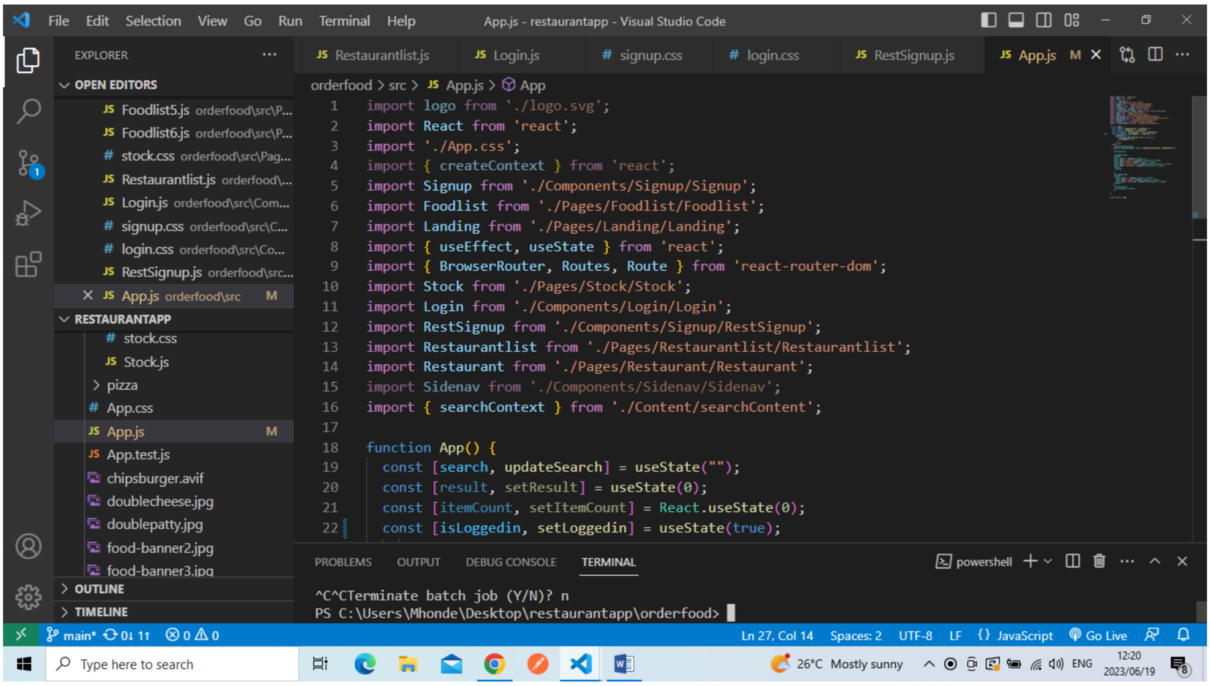
combo burger
Price: 65

Number of items=0

Total cost: 0

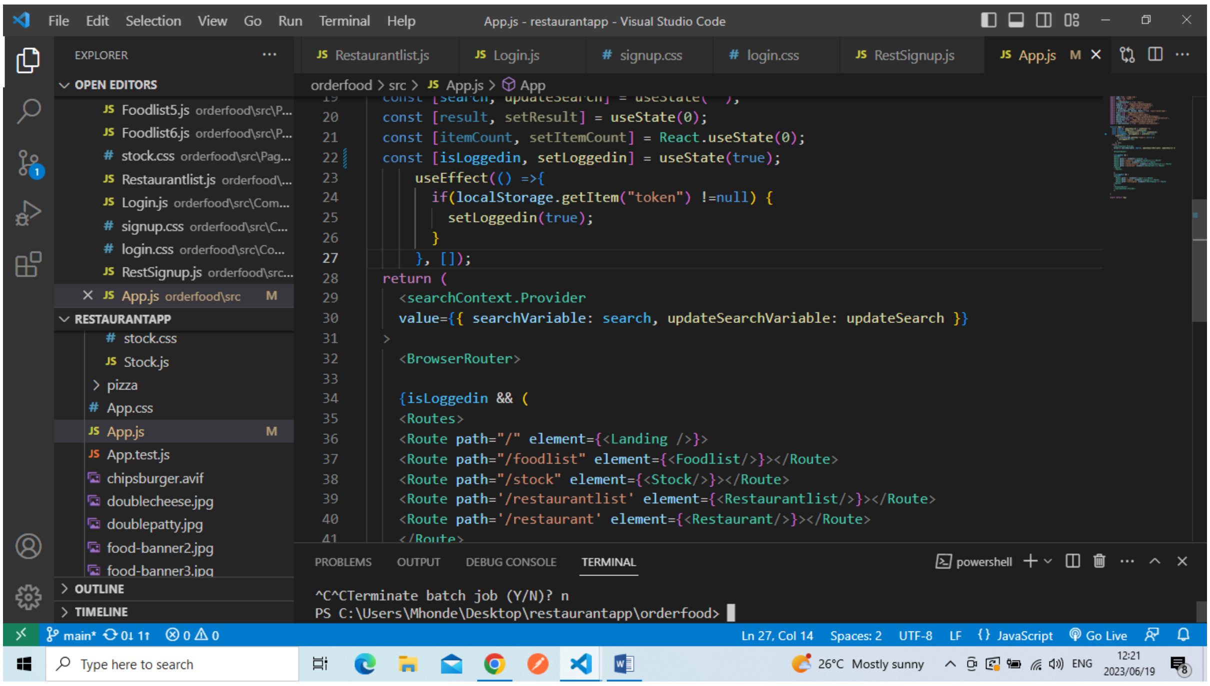
0

App.js



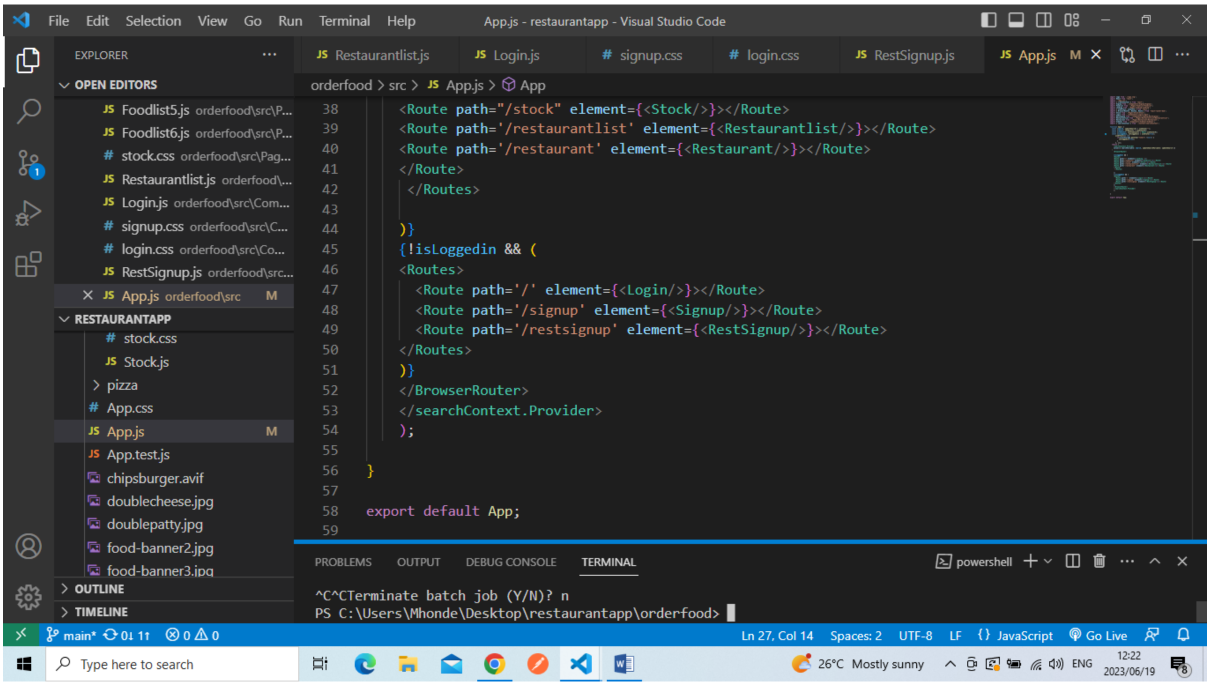
```
import logo from './logo.svg';
import React from 'react';
import './App.css';
import { createContext } from 'react';
import Signup from './Components/Signup/Signup';
import Foodlist from './Pages/Foodlist/Foodlist';
import Landing from './Pages/Landing/Landing';
import { useEffect, useState } from 'react';
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import Stock from './Pages/Stock/Stock';
import Login from './Components/Login/Login';
import RestSignup from './Components/Signup/RestSignup';
import Restaurantlist from './Pages/Restaurantlist/Restaurantlist';
import Restaurant from './Pages/Restaurant/Restaurant';
import Sidenav from './Components/Sidenav/Sidenav';
import { searchContext } from './Content/searchContent';

function App() {
  const [search, updateSearch] = useState("");
  const [result, setResult] = useState(0);
  const [itemCount, setItemCount] = React.useState(0);
  const [isLoggedin, setLoggedin] = useState(true);
```

```
^C^CTerminate batch job (Y/N)? n
PS C:\Users\Mhonde\Desktop\restaurantapp\orderfood>
```



```
const [search, updateSearch] = useState('');
const [result, setResult] = useState(0);
const [itemCount, setItemCount] = React.useState(0);
const [isLoggedin, setLoggedin] = useState(true);
    useEffect(() =>{
      if(localStorage.getItem("token") !=null) {
        setLoggedin(true);
      }
    }, []);
return (
  <searchContext.Provider
  value={{ searchVariable: search, updateSearchVariable: updateSearch }}
>
  <BrowserRouter>

  {isLoggedin && (
  <Routes>
  <Route path="/" element={<Landing />}>
  <Route path="/foodlist" element={<Foodlist/>}></Route>
  <Route path="/stock" element={<Stock/>}></Route>
  <Route path='/restaurantlist' element={<Restaurantlist/>}></Route>
  <Route path='/restaurant' element={<Restaurant/>}></Route>
  </Route>
```

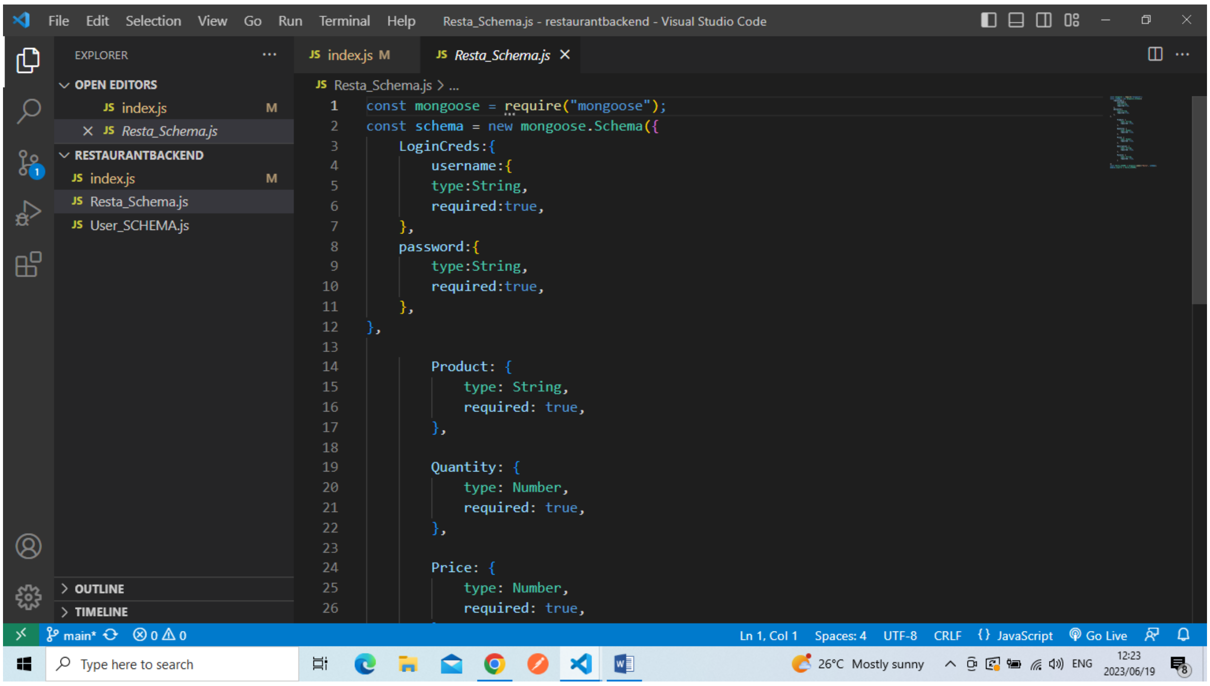
```
^C^CTerminate batch job (Y/N)? n
PS C:\Users\Mhonde\Desktop\restaurantapp\orderfood> 
```



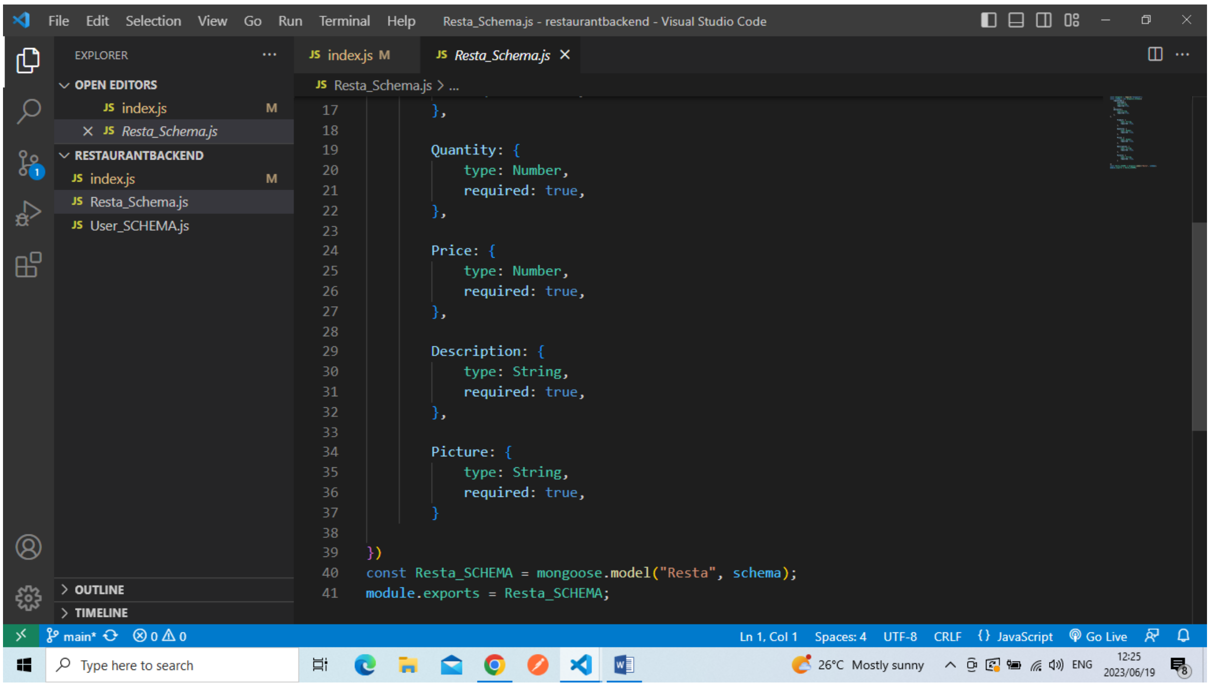
```
  <Route path="/stock" element={<Stock/>}></Route>
  <Route path='/restaurantlist' element={<Restaurantlist/>}></Route>
  <Route path='/restaurant' element={<Restaurant/>}></Route>
  </Route>
   </Routes>

  )}
  {!isLoggedin && (
  <Routes>
    <Route path='/' element={<Login/>}></Route>
    <Route path='/signup' element={<Signup/>}></Route>
    <Route path='/restsignup' element={<RestSignup/>}></Route>
  </Routes>
  )}
  </BrowserRouter>
  </searchContext.Provider>
  );

}

export default App;
```

```
^C^CTerminate batch job (Y/N)? n
PS C:\Users\Mhonde\Desktop\restaurantapp\orderfood> 
```

Product Schema

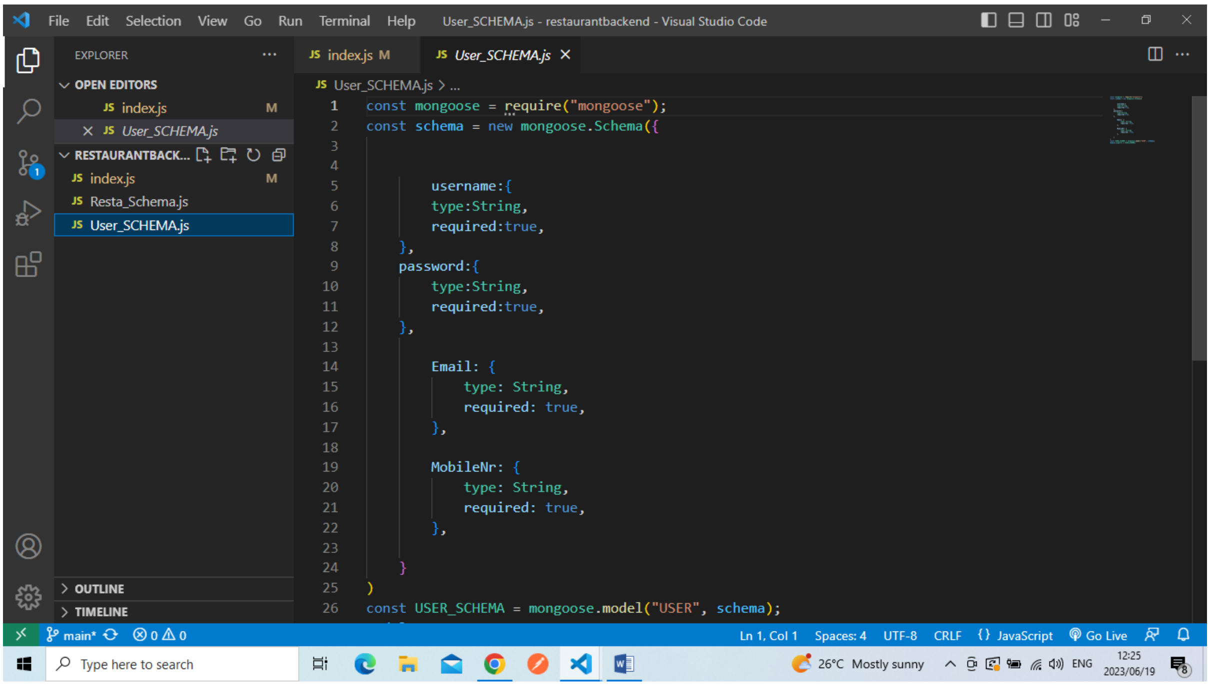


```
const mongoose = require("mongoose");
const schema = new mongoose.Schema({
    LoginCreds:{
        username:{
        type:String,
        required:true,
    },
    password:{
        type:String,
        required:true,
    },
},

        Product: {
            type: String,
            required: true,
        },

        Quantity: {
            type: Number,
            required: true,
        },

        Price: {
            type: Number,
            required: true,
```

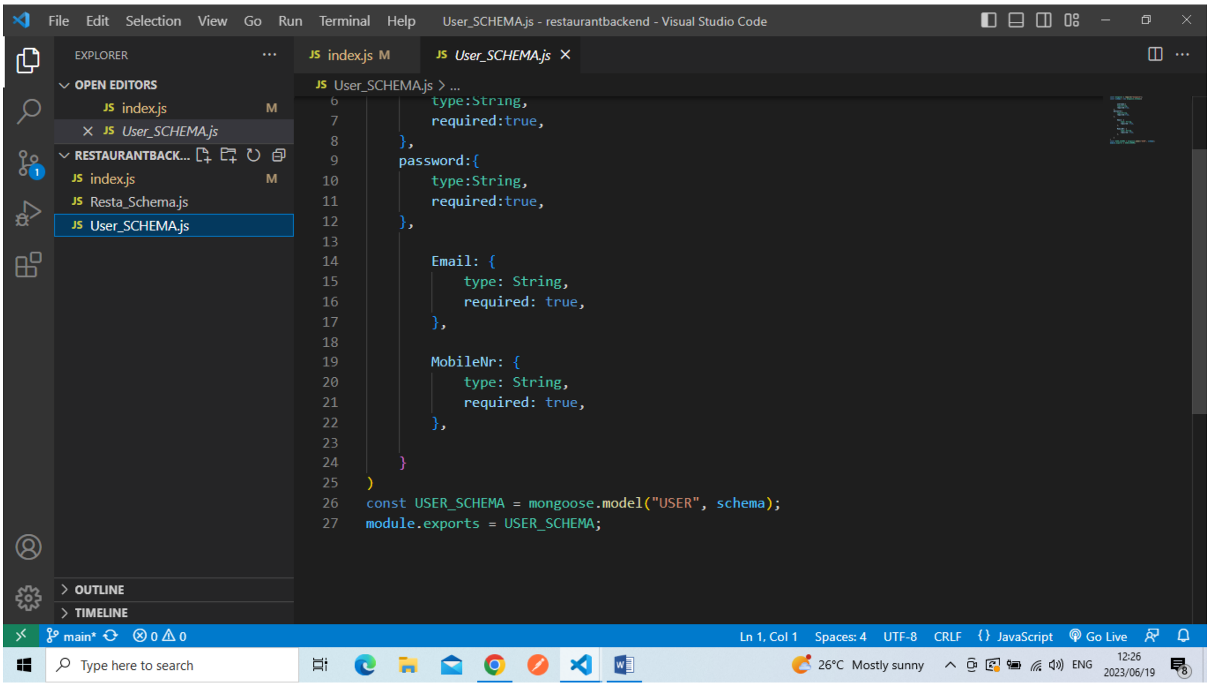


```
        },

        Quantity: {
            type: Number,
            required: true,
        },

        Price: {
            type: Number,
            required: true,
        },

        Description: {
            type: String,
            required: true,
        },

        Picture: {
            type: String,
            required: true,
        }

})
const Resta_SCHEMA = mongoose.model("Resta", schema);
module.exports = Resta_SCHEMA;
```

User Schema



```javascript
const mongoose = require("mongoose");
const schema = new mongoose.Schema({


        username:{
        type:String,
        required:true,
    },
    password:{
        type:String,
        required:true,
    },

        Email: {
            type: String,
            required: true,
        },

        MobileNr: {
            type: String,
            required: true,
        },

    }
)
const USER_SCHEMA = mongoose.model("USER", schema);
```



```javascript
        type:String,
        required:true,
    },
    password:{
        type:String,
        required:true,
    },

        Email: {
            type: String,
            required: true,
        },

        MobileNr: {
            type: String,
            required: true,
        },

    }
)
const USER_SCHEMA = mongoose.model("USER", schema);
module.exports = USER_SCHEMA;
```