

1. A pdf file which is a document where you explain how you approached the problem statement.

A pdf file which is a document where you explain how you approached the problem statement.

Frontend:

In these Food Ordering Web Application I started from basic things like how it should be look like and how it work as given in problem statement, so according that started from Frontend part to building

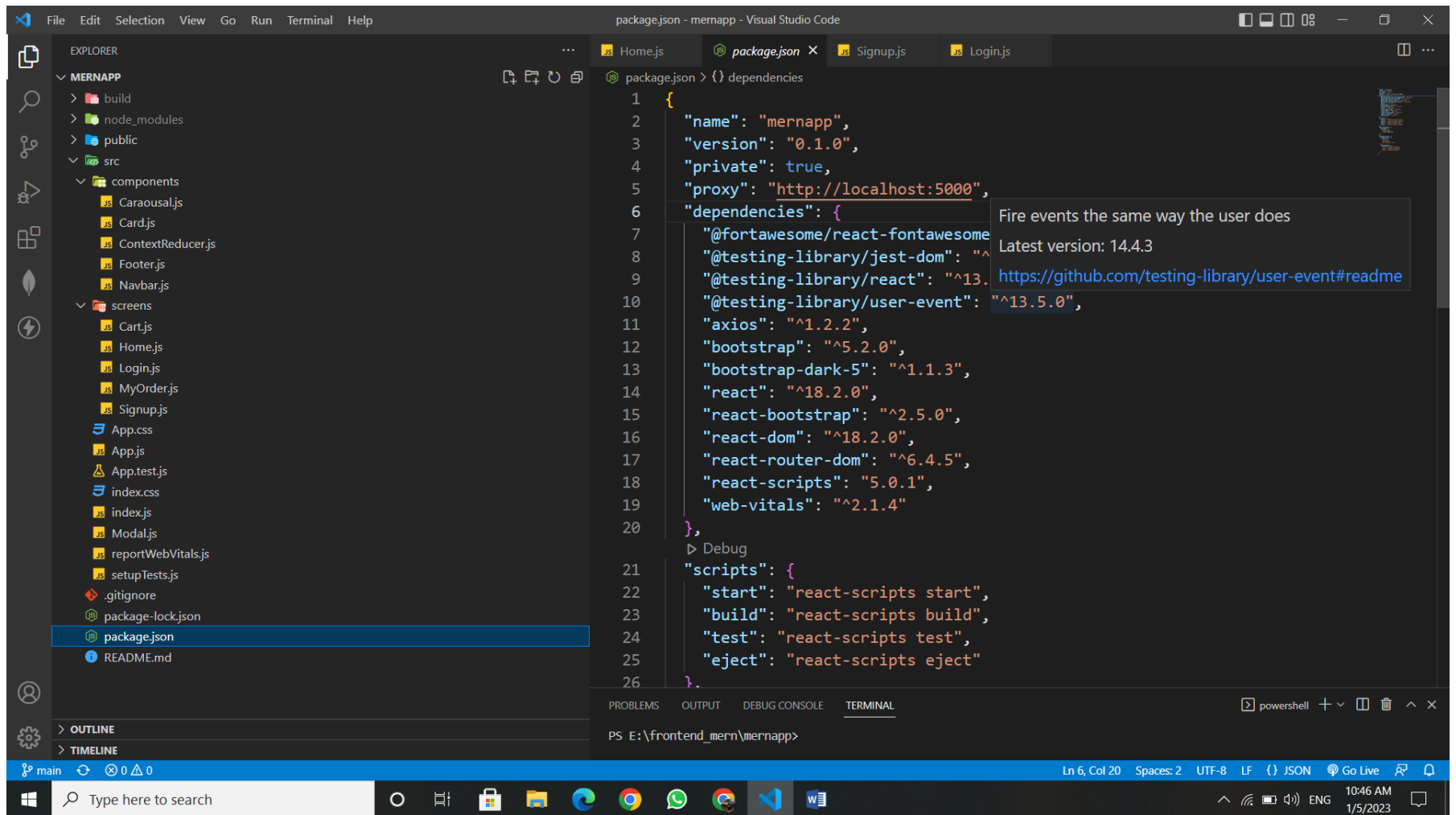
```
:installed all npm pkg
: "npx create-react-app mernapp"
: "npm i bootstrap"
: "npm i react-router-dom"           //for routing purpose
: etc packages...
```

: firstly create a frontend web application then it's file structure and folders and sub-folders and files

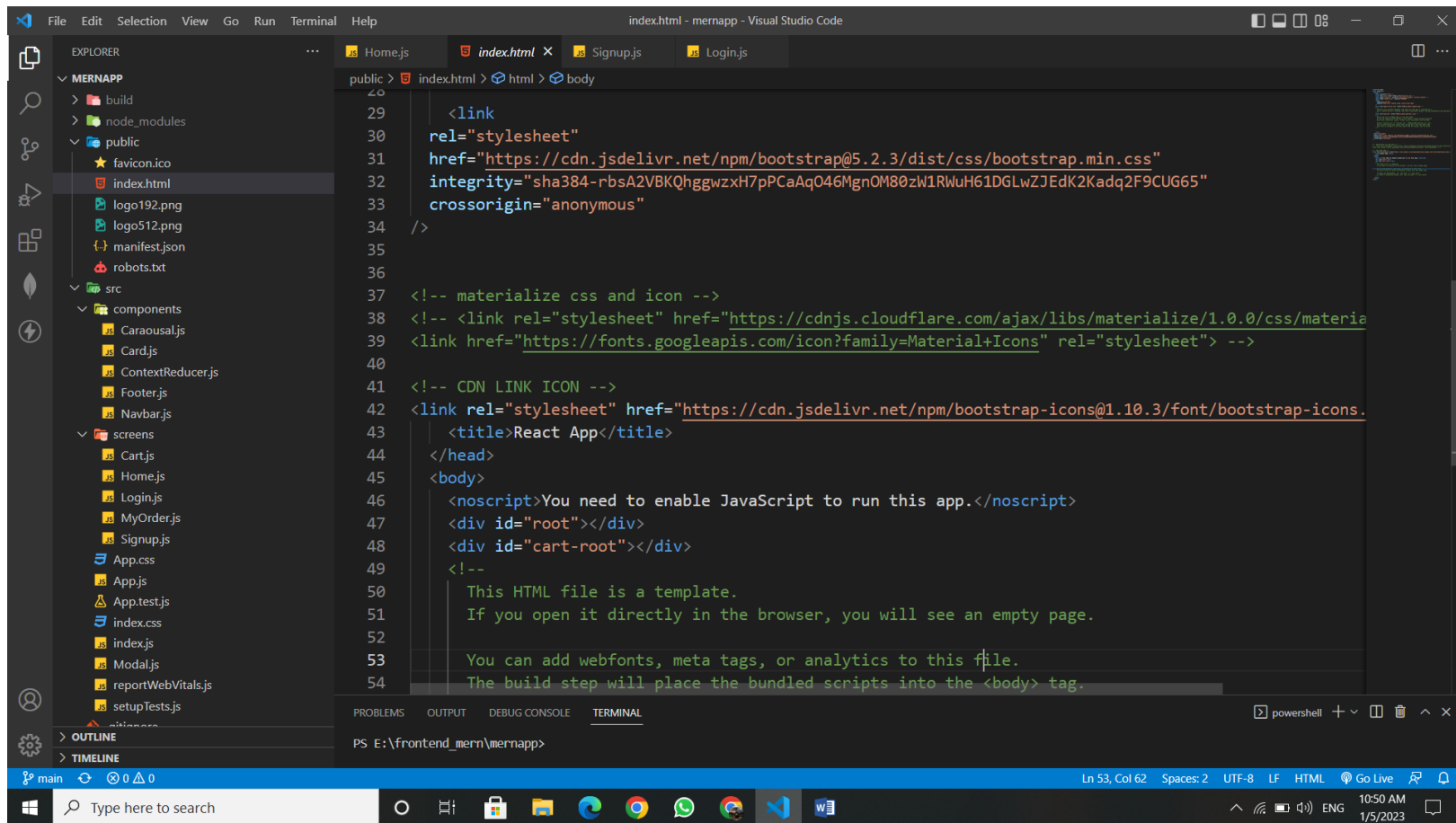
```
: In src components
  -components
    :Carousal.js
    :Card.js
    :ContextReducer.js
    :Footer.js
    :Navbar.js

  -screens
    :Cart.js
    :Home.js
    :Login.js
    :MyOrder.js
    :Signup.js
```

This is file and folder structure for project



Then also use cdn links in index.html page which are like for bootstrap and logo or styling
Inside head tag:



Then we started work on to build Navbar.js page
In that we use bootstrap Navbar code and make some changes and styling
And then created Footer.js page and make code for that then just render inside of
Home.js page.

To create Nabar.js

: I used and imported pkges like

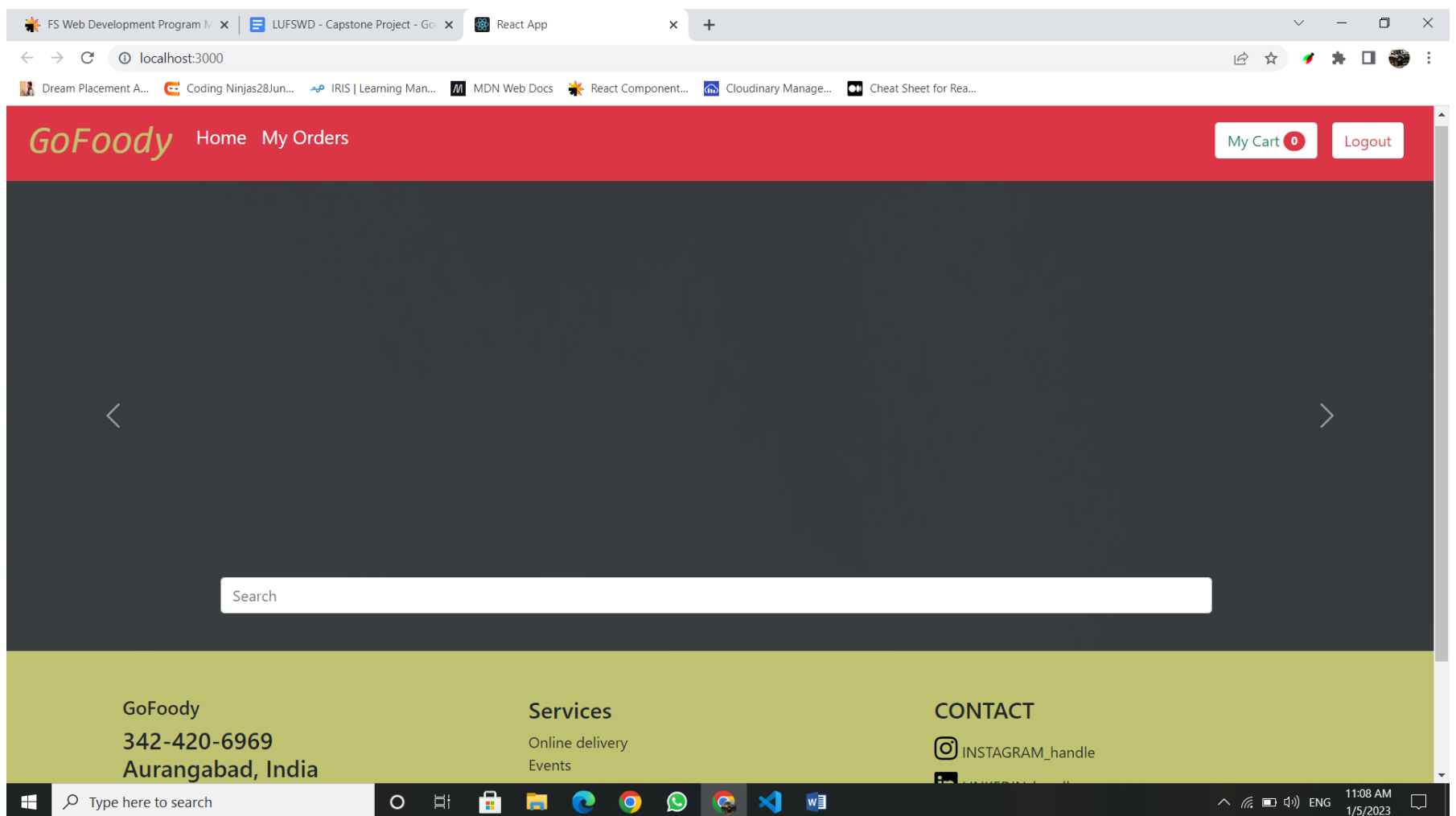
```
import React, { useState } from 'react'
import { Link, useNavigate } from 'react-router-dom'
import Badge from 'react-bootstrap/Badge';
import Modal from '../Modal';
import Cart from '../screens/Cart';
import { useCart } from './ContextReducer';
```

: where useState used to create Hooks or variabele whre we create initialState and
currentState
and then just update the state of initial by using currentState

:useNavigate and Link imported from react-router-dom which used to navigate from one component to another or it work like anchor tage but ,
: Link are advanced from anchor tag where Link tag never refresh our page or rerender page it directly navigate to particular path
: useNavigate is hook which used to navigate to another component or page...
:Modal is component we rendering here and Cart also rendering here and same..

After that I create Footer.js

: footer also we used bootstrap code to create these and just return xml code which looks UI friendly and make contact for our restaurant... and give there google map and social handles like Instagram , googleMap and LinkedIn Link we used bootstrap icon here...



: in backend I created server then install npm pkgs

```
"author": "",
```

```
"license": "ISC",
"dependencies": {
  "bcryptjs": "^2.4.3",
  "express": "^4.18.2",
  "express-validator": "^6.14.2",
  "jsonwebtoken": "^9.0.0",
  "mongoose": "^6.8.0",
  "nodemon": "^2.0.20"
}
}
```

: After that I work on backend where I created server and then join it with backend mongo database and use mongoURI and connected in .env file where I created

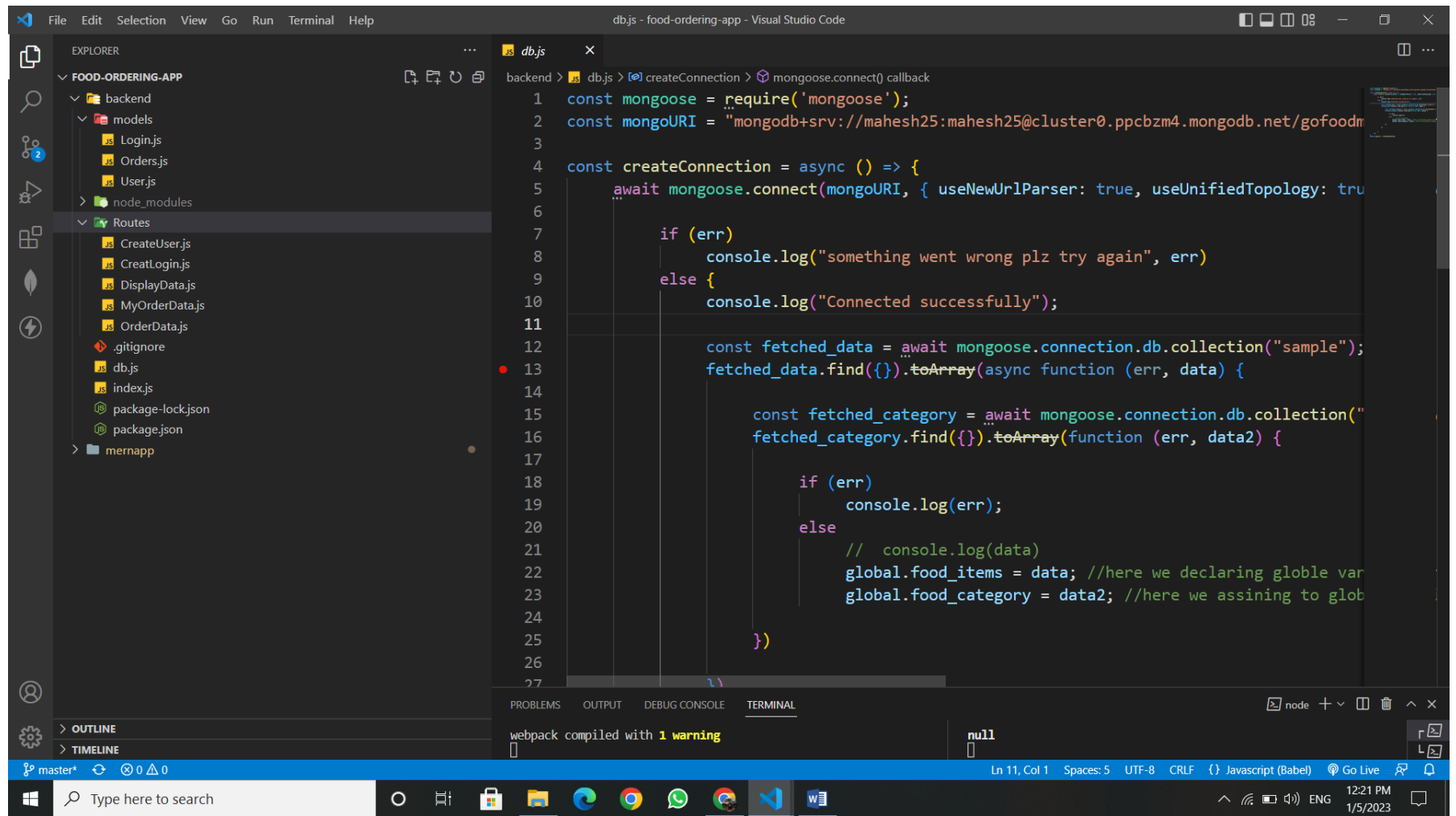
: then in file I use mongoURI and PORT which I access in db.js

```
const mongoURI =
"mongodb+srv://mahesh25:mahesh25@cluster0.ppcbzm4.mongodb.net/gofoodmern?retryWrites=true&w=majority"
```

then created methods in index.js where I create server and methods and also access mongoose routers in that page

once my server running on a port 5000 successfully I'm ready to go to create backend models in that I created Schema body for signin and signout pages and Home.js page

file structure of backend



Then in models folder I created Schema body for each frontend pages and Login.js

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

const {Schema} = mongoose //here we destructuring the mongoose which connected to
mongoURI online cloud we can access/fetch the online data or
//or we can send the data from here

const loginSchema = new Schema({ //now create the object for Schema and create
Schema body
  email:{
    type : String,
    require:true
  },
  password:{
```

Then same for logout page and Home.js page we created and connected backed and fronted to them.

Once backend code completed for each page I then create frontend pages for these routers or endpoints and using fetch method we access and join both backend and frontend part

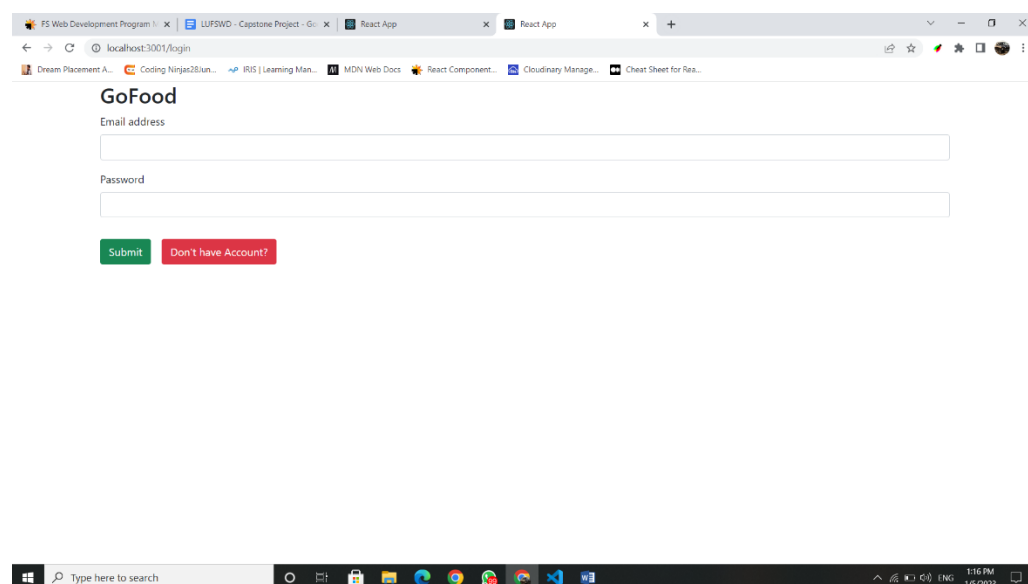
Signin.js

The screenshot displays a web browser window with the address bar showing 'localhost:3001/creatuser'. The page title is 'GoFood'. The form contains the following elements:

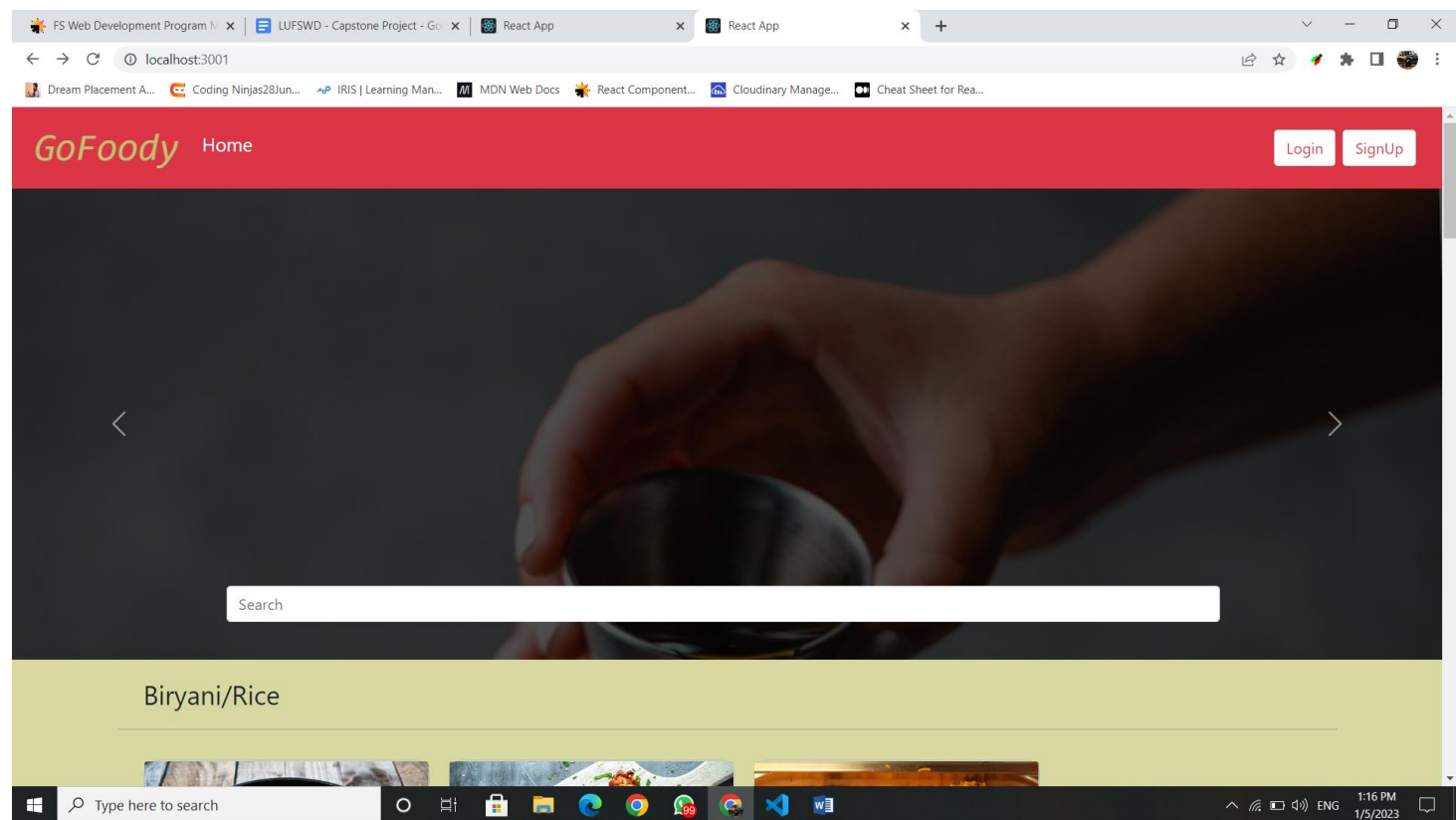
- Name:** A text input field.
- Email address:** A text input field.
- Mobile:** A text input field.
- Password:** A text input field.
- Submit:** A green button.
- Already Have an Account?:** A red button.

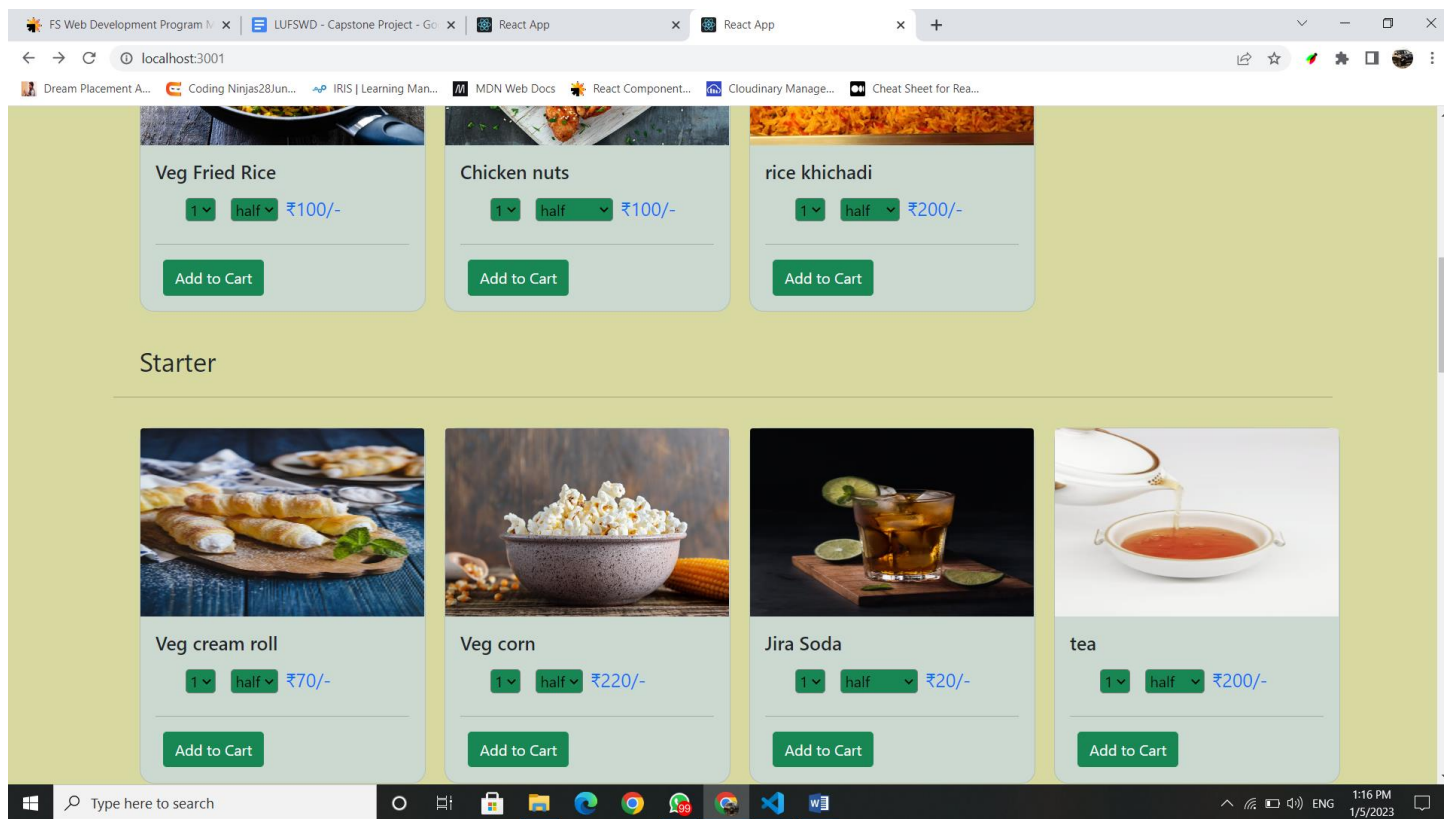
The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system clock in the bottom right corner shows '1:16 PM' and '1/5/2023'.

Create login page



Home.js page :

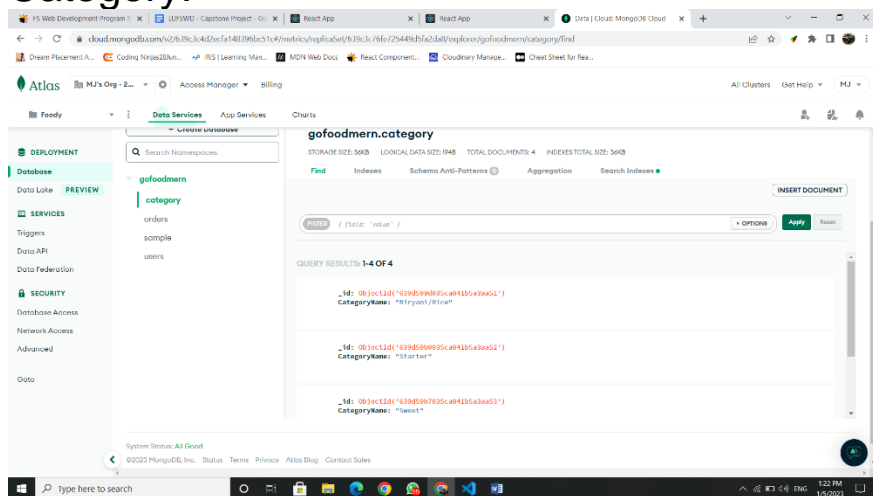




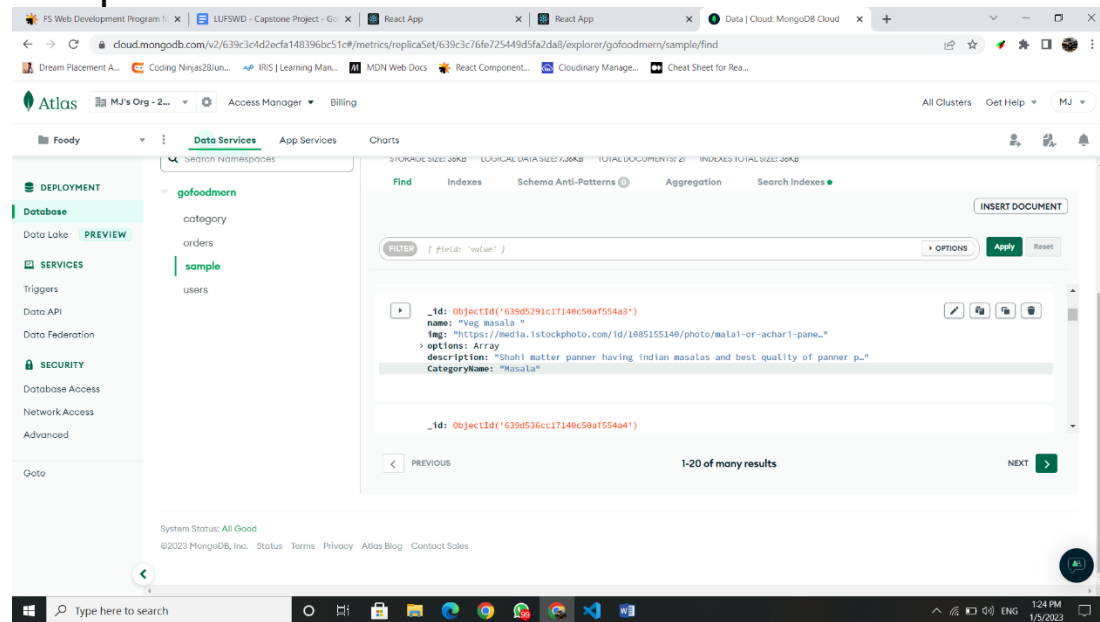
And then we whatever data we send or filed from frontend that saved on backend part i.e. mongodb cluster

Colleactions we created and data stored

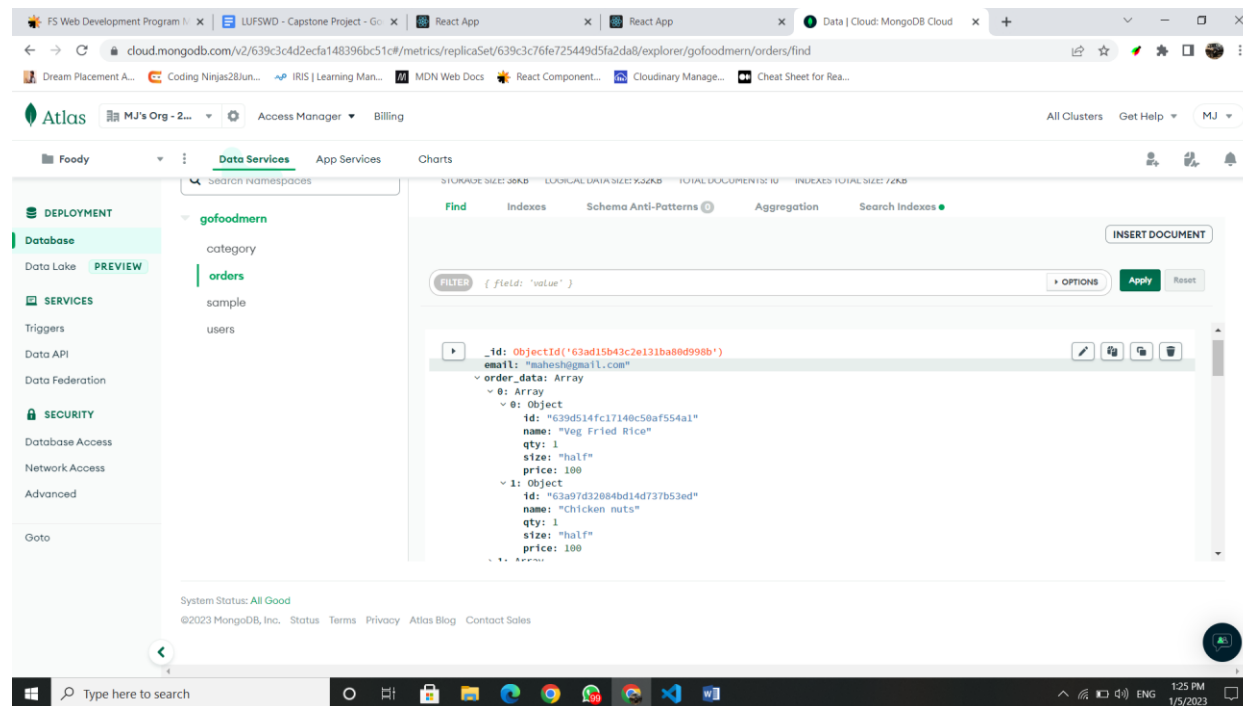
Category:



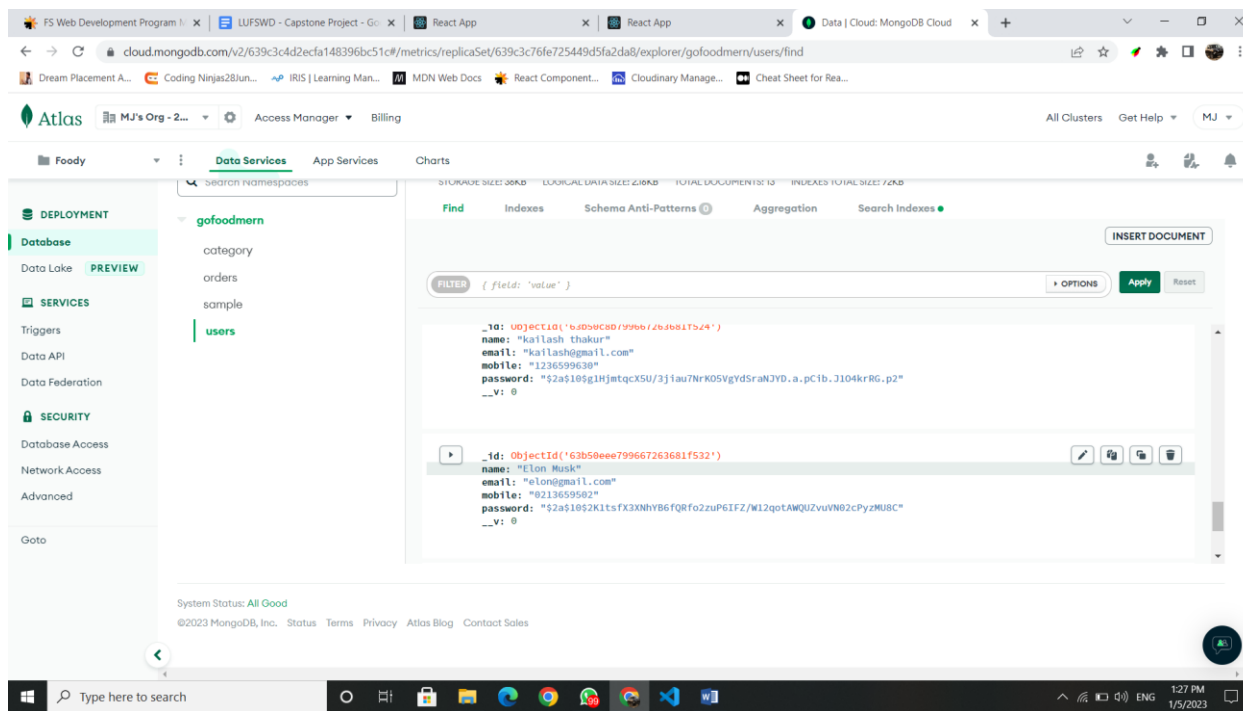
Sample collection used to stored all food-items and data



Orders:



Users collection used to stored user data :



These is approach I created MERN full stack web application and used many things here to create and also password will in the encrypted format

These is all about My project approach