**Full Stack Development with MERN**

**Frontend Development Report**

|  |  |
| --- | --- |
| Date | 9 July 2024 |
| Team ID | SWTID1720171927 |
| Project Name | SBFoods- Food Ordering Website |
| Maximum Marks | 10 |

**Project Title: SBFoods**

Date: 9 July 2024

Prepared by: SWTID1720171927

**Objective**

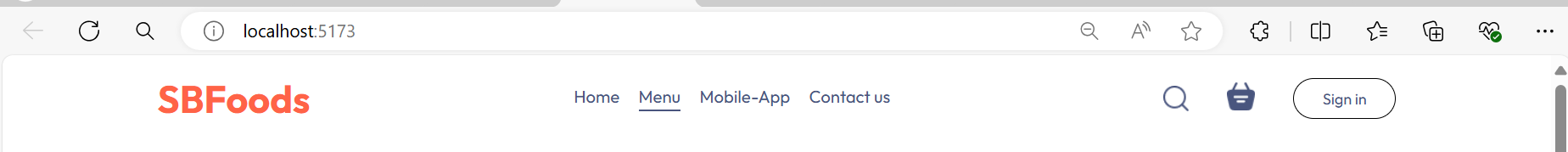
The objective of this report is to document the frontend development progress and key aspects of the user interface implementation for the SBFoods project.

**Technologies Used**

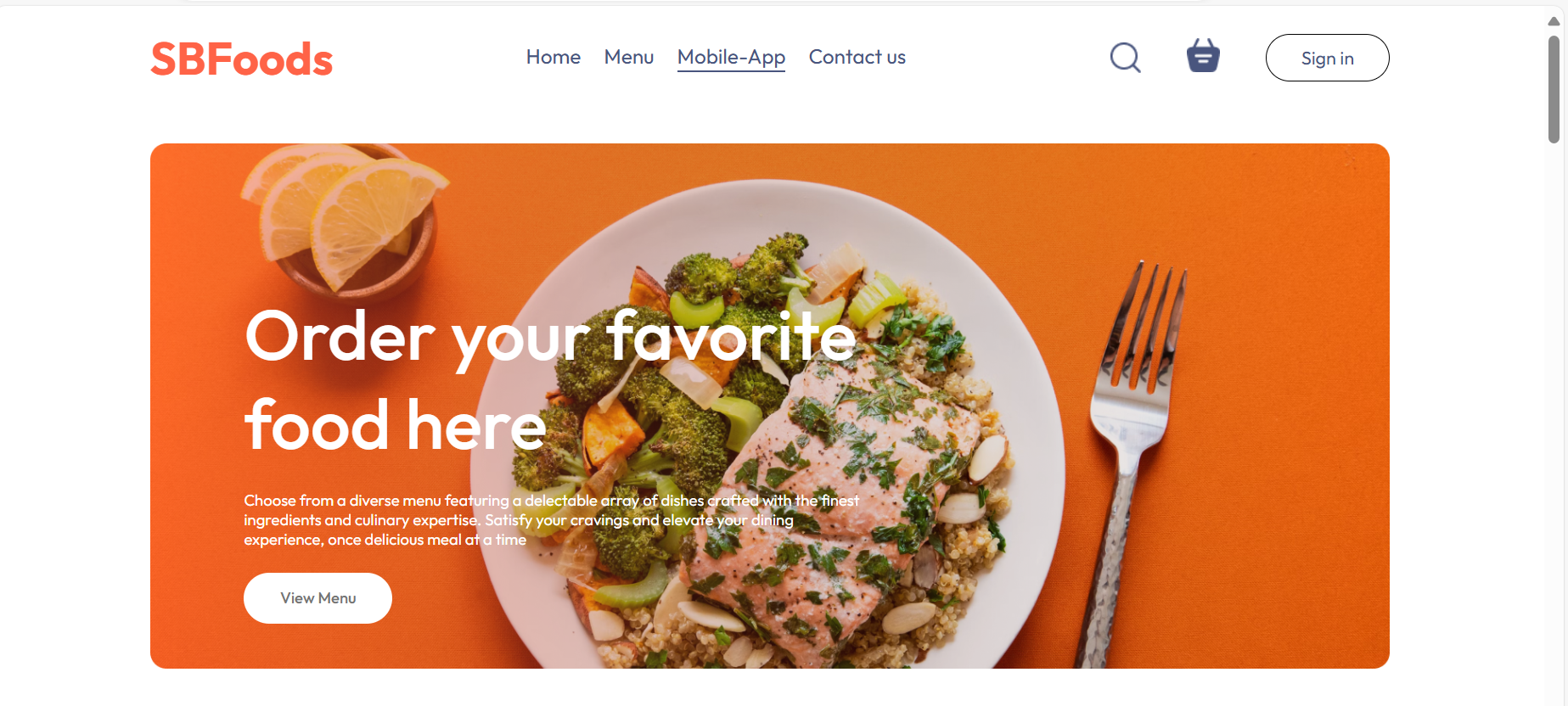
* **Frontend Framework:** React.js
* **UI Framework/Libraries:** Bootstrap
* **API Libraries:** Axios

**Project Structure**

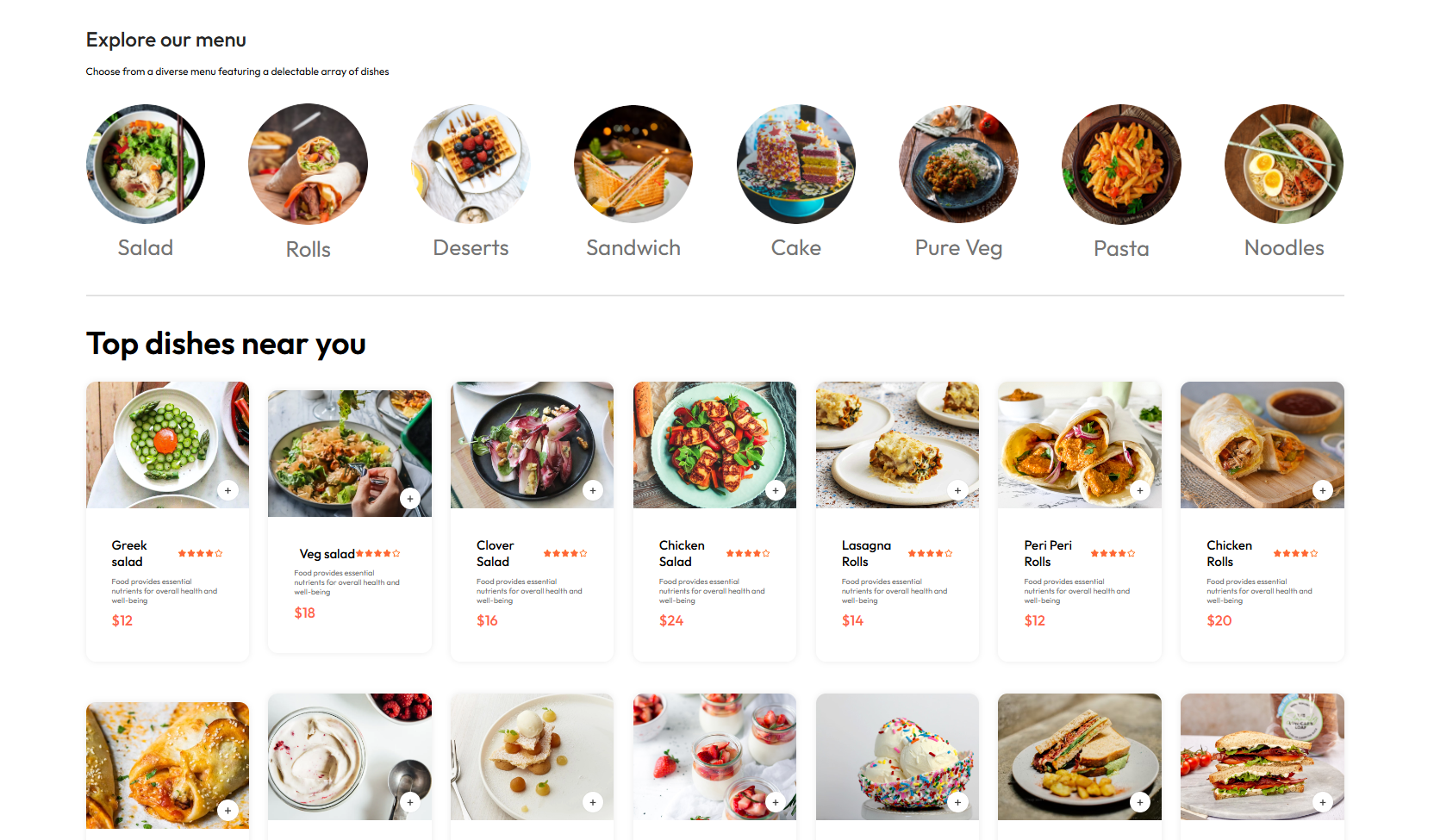
**Navbar**

****

**Home**

****

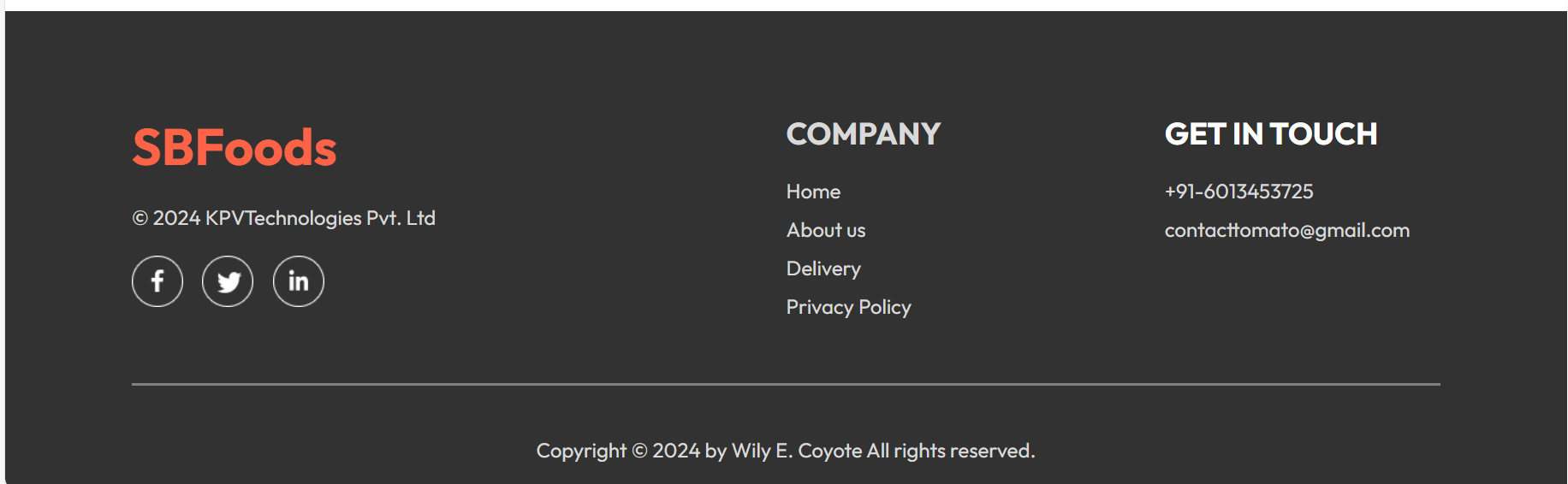
**Menu**

****

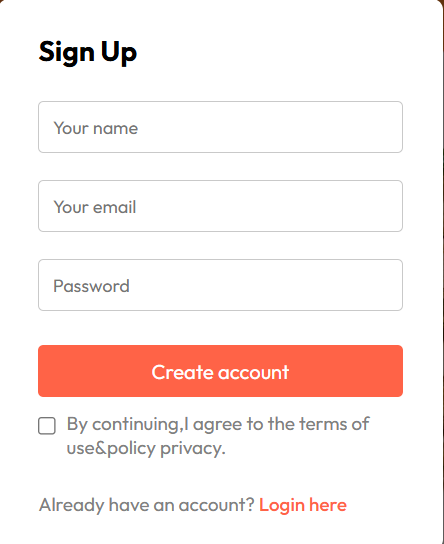
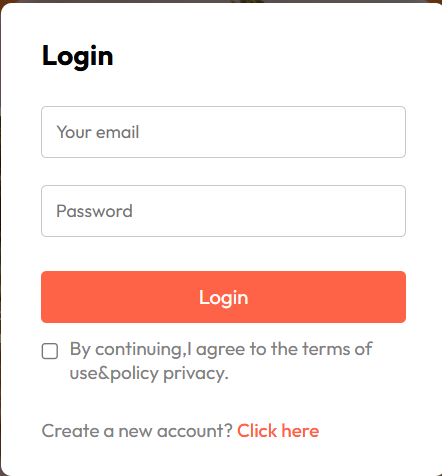
**Mobile-App**

****

**About us**

****

**Sign up/Login Functionality**

** **

**Key Components**

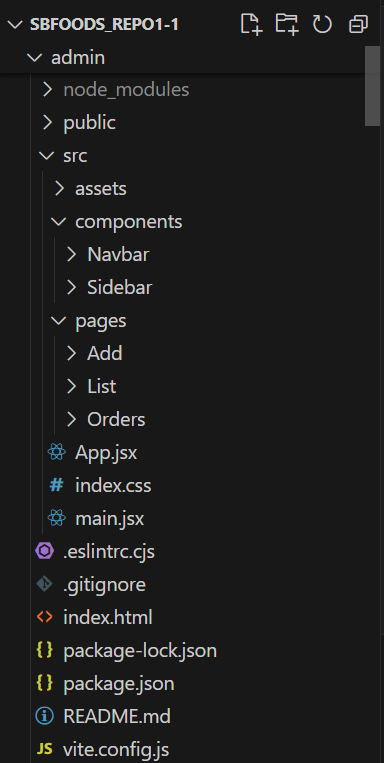
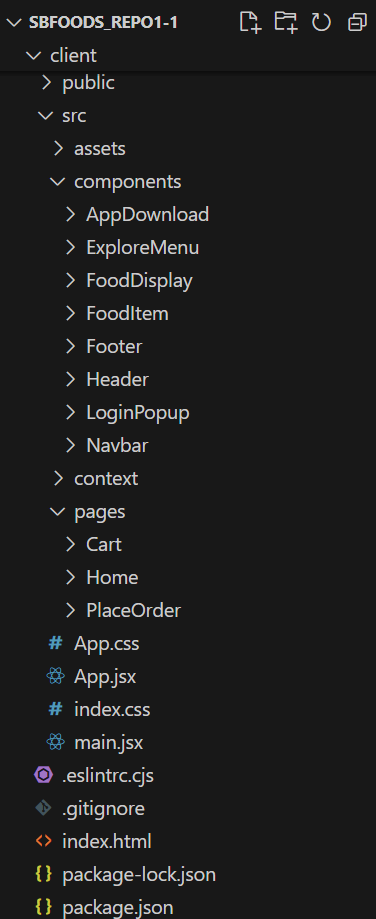
**App.jsx**

**Admin Client**

** **

**Components and Pages**

**­ Admin Client**

**** ­­­

**Routing**

Routing is managed using React Router. Here are the main routes:

Client-side:

* **/** - Landing page of the application.
* **/ cart**- displays the items added in to the cart
* **/order** – getting delivery address information and calculating the total amount to be paid
* **/verify** - verify the payment being processed
* **/myOrders** – allows user to track their order status(Processing/Out for Delivery/Delivered)

Admin-side

* **/add**- uploading new dish to the menu
* **/list**- the available dishes in the menu are displayed
* **/orders –** display user orders

**Integration with Backend**

The frontend communicates with the backend APIs hosted on backend URL. Key endpoints include:

* userRouter.post("/register", registerUser) – user registration
* userRouter.post("/login", loginUser) -user login
* foodRouter.post("/add",upload.single("image"),addFood) – updating menu by uploading image of new dish
* foodRouter.get("/list",listFood) – display dishes present in the database
* foodRouter.post("/remove",removeFood) – remove a dish from menu
* cartRouter.post('/add',authMiddleware,addToCart) – add the dish to cart
* cartRouter.post('/remove',authMiddleware,removeFromCart) – remove dish from the cart
* cartRouter.post('/get',authMiddleware,getCart) – display the items in cart
* orderRouter.post('/place',authMiddleware,placeOrder) – order being placed by user
* orderRouter.post('/verify',verifyOrder) - verify the order made by user
* orderRouter.post('/userorders',authMiddleware,userOrders) – get the orders made by a specific user
* orderRouter.get('/list',listOrders) - get the list of orders that were made
* orderRouter.post("/status",updateStatus) – update the order status(Processing/Out for Delivery/Delivered)

**User Interface (UI) Design**

The UI design of the SBFoods Food Ordering Website follows modern design principles, emphasizing usability and aesthetics to provide an optimal user experience. Here are the key design principles implemented in the project:

1. Consistency

* Bootstrap: We use BootStrap to ensure consistent styling across all components, providing a unified look and feel throughout the application.
* Typography and Colors: Consistent use of fonts, colors, and spacing enhances readability and visual harmony.

2. Responsiveness

* Responsive Design: The UI is designed to be fully responsive, ensuring that it looks and functions well on a variety of devices, including desktops, tablets, and smartphones.
* Media Queries: Utilized CSS media queries to adjust the layout and components based on the screen size, ensuring a seamless user experience across all devices.

3. Usability

* Intuitive Navigation: Clear and intuitive navigation ensures users can easily find what they are looking for without confusion.r

**Third-Party Integrations**

**Payment Gateway: Stripe**

Stripe is integrated as the payment gateway to securely handle transactions on the SBFoods Food Ordering Website. It provides a user-friendly checkout experience, supporting various payment methods including credit cards, debit cards, and digital wallets.