**REACT.JS**

Table of Contents:

    Reacts Fragments..................................................................................1

    React Portals......................................................................................... 1

    React Refs (References).........................................................................2

    What is an Effect?..................................................................................3

    React use Reducer () .............................................................................3

    React use State () ..................................................................................4

Project ………………………………………………………………………………………………5

Outputs……………………………………………………………………………………………...18

1.React Fragments:

* We can use <> and </> instead of <div> and </div>. Because, it is an empty wrapper component. It doesn’t render any real HTML element to the DOM (Document Object Model). But it fulfills React’s/JSX requirement.
* The Code Execution is faster with use of React Fragments as Compared to the div tag.
* Less utilization of memory with the use of react Fragments instead of div tag.
* You can declare Fragments in two types: -
* <> and </>
* <Fragment> and </Fragment>

2. React Portals:

* React Portal is a first-class way to render child components into a DOM node outside of the parent DOM hierarchy defined by the component hierarchy.

3. React Refs (References):

* Refs are a function provided by react to access the DOM element and react element that you might have created on your own.
* They are used in cases where we want to change the value of the child component, without using props and all.
* They also provide us with good functionality as we can use callbacks with them.
* Depending on the type of the node, the value of the ref can defer.
* The ref attribute doesn’t have instances, and can’t be used on Functional Components.

Example: Adding Refs to DOM elements.

**import** React, { Component } from 'react';     
import { render } from 'react-dom';     
class App **extends** React.Component {     
constructor(props) {     
super(props);     
this.callRef = React.createRef();     
this.addingRefInput = **this**.addingRefInput.bind(**this**);     
}     
addingRefInput() {     
this.callRef.current.focus();     
}     
render() {     
return (     
<div>     
<h1>Hello World</h1>     
<input     
type="text"     
ref={**this**.callRef} />     
<input     
type="button"     
value="Enter text"     
onClick={**this**.addingRefInput}     
/>     
</div>    
);     
}     
}     
export **default** App;

* **Output:**   
  

4. React use Effect ():

* The use Effect () hook allows you to performs side effects in your components.
* Some examples of side components are: fetching data, directly updating the DOM and timers.

5. React use Reducer ():

* The use Reducer () can be used as a replacement for use State (), if you need “more powerful state management”.
* It allows custom state logic.
* The use Reducer () hook accepts two arguments.

* How is use Reducer () is used?

Const [state, dispatch FN] = use Reducer (reducer FN, initial State, init FN);

* State= The state snapshot used in the component render
* Dispatch FN= A function that is used to dispatch a new action
* Reducer FN= A function that is triggered automatically once an action is dispatched.
* Intial State= Initial state
* Init FN= A function to set the initial state programmatically.

6. React use State ():

* The main state management tool
* Great for independent use of state.
* Great if state updates are easy and limited to a few kinds of updates.
* At the top of your component, import the use State () Hook:

import {use State} from "react";

7. Creating a new project using Handling side effects, using reducers, and using context API, Use State method, named as Cart Icon Button……

To Start with my second React.js project, I first downloaded node.js from my internet browser.

* Then I installed it in my PC and went into Command Prompt or Terminal.
* I then entered the following syntax to install npm directories:
* **npx create-react-app my-app**

* Instead of my-app, I used sample-app for the new project.
* Then I opened Visual Studio Code and Opened the My-First-App folder and then opened terminal in Visual studio code.
* Then I entered the following syntax to get work of npm:
* **Npm start**
* This above command starts the app in development mode.

**Folder Structure:**

Source: -

Assets: -

        Meals.jpg

Components: -

    Cart: -

Cart.js

        Cart.module.css

        CartIcon.js

        CartItem.js

CartItem.module.css

Layout: -

Header.js

Header.module.css

HeaderCartButton.js

HeaderCartButton.module.css

Meals: -

  Meals Item: -

AvailableMeals.js

AvailableMeals.module.css

MealItem.js

MealItem.module.css

MealItemForm.js

MealItemForm.module.css

Meals.js

MealsSummary.js

MealsSummary.module.css

UI: -

Card.js

Card.module.css

Input.js

Input.module.css

Modal.js

Modal.module.css

Store: -

Cart-context.js

CartProvider.js

Src: -

App.js

App.css

App.est.js

Index.css

Index.js

First creates the header bar Home

Header.js: -

import { Fragment } from 'react';

import HeaderCartButton from './HeaderCartButton';

import mealsImage from '../../assests/meals.jpg';

import classes from './Header.module.css';

const Header = (props) => {

  return (

    <Fragment>

      <header className={classes.header}>

        <h1>Five Star Meals!!</h1>

        <HeaderCartButton onClick={props.onShowCart} />

      </header>

      <div className={classes['main-image']}>

        <img src={mealsImage} alt='A table full of delicious food!' />

      </div>

    </Fragment>

  );

};

export default Header;

Header module.css:-

.header {

  position: fixed;

  top: 0;

  left: 0;

  width: 100%;

  height: 5rem;

  background-color: #F16E4B;

  color: white;

  display: flex;

  justify-content: space-between;

  align-items: center;

  padding: 0 10%;

  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.25);

  z-index: 10;

}

.main-image {

  width: 100%;

  height: 25rem;

  z-index: 0;

  overflow: hidden;

}

.main-image img {

  width: 110%;

  height: 100%;

  object-fit: cover;

  transform: rotateZ(-5deg) translateY(-4rem) translateX(-1rem);

}

HeaderCartButton.js:-

import { useContext, useEffect, useState } from 'react';

import CartIcon from '../Cart/CartIcon';

import CartContext from '../../Store/cart-context';

import classes from './Header';

const HeaderCartButton = (props) => {

  const [btnIsHighlighted, setBtnIsHighlighted] = useState(false);

  const cartCtx = useContext(CartContext);

  const { items } = cartCtx;

  const numberOfCartItems = items.reduce((curNumber, item) => {

    return curNumber + item.amount;

  }, 0);

  const btnClasses = `${classes.button} ${btnIsHighlighted ? classes.bump : ''}`;

  useEffect(() => {

    if (items.length === 0) {

      return;

    }

    setBtnIsHighlighted(true);

    const timer = setTimeout(() => {

      setBtnIsHighlighted(false);

    }, 300);

    return () => {

      clearTimeout(timer);

    };

  }, [items]);

  return (

    <button className={btnClasses} onClick={props.onClick}>

      <span className={classes.icon}>

        <CartIcon />

      </span>

      <span>Your Cart</span>

      <span className={classes.badge}>{numberOfCartItems}</span>

    </button>

  );

};

export default HeaderCartButton;

HeaderCartButton.module.css:

.button {

  cursor: pointer;

  font: inherit;

  border: none;

  background-color: #4d1601;

  color: white;

  padding: 0.75rem 3rem;

  display: flex;

  justify-content: space-around;

  align-items: center;

  border-radius: 25px;

  font-weight: bold;

}

.button:hover,

.button:active {

  background-color: #2c0d00;

}

.icon {

  width: 1.35rem;

  height: 1.35rem;

  margin-right: 0.5rem;

}

.badge {

  background-color: #b94517;

  padding: 0.25rem 1rem;

  border-radius: 25px;

  margin-left: 1rem;

  font-weight: bold;

}

.button:hover .badge,

.button:active .badge {

  background-color: #92320c;

}

.bump {

  animation: bump 300ms ease-out;

}

@keyframes bump {

  0% {

    transform: scale(1);

  }

  10% {

    transform: scale(0.9);

  }

  30% {

    transform: scale(1.1);

  }

  50% {

    transform: scale(1.15);

  }

  100% {

    transform: scale(1);

  }

}

Cart.js

import { useContext } from 'react';

import Modal from '../UI/Modal';

import CartItem from './CartItem';

import classes from './Cart.module.css';

import CartContext from '../../Store/cart-context';

const Cart = (props) => {

  const cartCtx = useContext(CartContext);

  const totalAmount = `$${cartCtx.totalAmount.toFixed(2)}`;

  const hasItems = cartCtx.items.length > 0;

  const cartItemRemoveHandler = (id) => {

    cartCtx.removeItem(id);

  };

  const cartItemAddHandler = (item) => {

    cartCtx.addItem({ ...item, amount: 1 });

  };

  const cartItems = (

    <ul className={classes['cart-items']}>

      {cartCtx.items.map((item) => (

        <CartItem

          key={item.id}

          name={item.name}

          amount={item.amount}

          price={item.price}

          onRemove={cartItemRemoveHandler.bind(null, item.id)}

          onAdd={cartItemAddHandler.bind(null, item)}

        />

      ))}

    </ul>

  );

  return (

    <Modal onClose={props.onClose}>

      {cartItems}

      <div className={classes.total}>

        <span>Total Amount</span>

        <span>{totalAmount}</span>

      </div>

      <div className={classes.actions}>

        <button className={classes['button--alt']} onClick={props.onClose}>

          Close

        </button>

        {hasItems && <button className={classes.button}>Order</button>}

      </div>

    </Modal>

  );

};

export default Cart;

Cart.module.css

.cart-items {

  list-style: none;

  margin: 0;

  padding: 0;

  max-height: 20rem;

  overflow: auto;

}

.total {

  display: flex;

  justify-content: space-between;

  align-items: center;

  font-weight: bold;

  font-size: 1.5rem;

  margin: 1rem 0;

}

.actions {

  text-align: right;

}

.actions button {

  font: inherit;

  cursor: pointer;

  background-color: transparent;

  border: 1px solid #8a2b06;

  padding: 0.5rem 2rem;

  border-radius: 25px;

  margin-left: 1rem;

}

.actions button:hover,

.actions button:active {

  background-color: #1872d2;

  border-color: #0070AC;

  color: white;

}

.actions .button--alt {

  color: #8a2b06;

}

.actions .button {

  background-color: #0070AC;

  color: white;

}

CartIcon.js:-

const CartIcon = () => {

  return (

    <svg

      xmlns='http://www.w3.org/2000/svg'

      viewBox='0 0 20 20'

      fill='currentColor'

    >

      <path d='M3 1a1 1 0 000 2h1.22l.305 1.222a.997.997 0 00.01.042l1.358 5.43-.893.892C3.74 11.846

      4.632 14 6.414 14H15a1 1 0 000-2H6.414l1-1H14a1 1 0 00.894-.553l3-6A1 1 0 0017 3H6.28l-.31-1.243A1

       1 0 005 1H3zM16 16.5a1.5 1.5 0 11-3 0 1.5 1.5 0 013 0zM6.5 18a1.5 1.5 0 100-3 1.5 1.5 0 000 3z' />

    </svg>

  );

};

export default CartIcon;

CartItem.js:-

import classes from './CartItem.module.css';

const CartItem = (props) => {

  const price = `$${props.price.toFixed(2)}`;

  return (

    <li className={classes['cart-item']}>

      <div>

        <h2>{props.name}</h2>

        <div className={classes.summary}>

          <span className={classes.price}>{price}</span>

          <span className={classes.amount}>x {props.amount}</span>

        </div>

      </div>

      <div className={classes.actions}>

        <button onClick={props.onRemove}>−</button>

        <button onClick={props.onAdd}>+</button>

      </div>

    </li>

  );

};

export default CartItem;

CartItem.module.css

.cart-item {

  display: flex;

  justify-content: space-between;

  align-items: center;

  border-bottom: 2px solid #8a2b06;

  padding: 1rem 0;

  margin: 1rem 0;

}

.cart-item h2 {

  margin: 0 0 0.5rem 0;

  color: #363636;

}

.summary {

  width: 10rem;

  display: flex;

  justify-content: space-between;

  align-items: center;

}

.price {

  font-weight: bold;

  color: #8a2b06;

}

.amount {

  font-weight: bold;

  border: 1px solid #ccc;

  padding: 0.25rem 0.75rem;

  border-radius: 6px;

  color: #363636;

}

.actions {

  display: flex;

  flex-direction: column;

}

@media (min-width: 768px) {

  .actions {

    flex-direction: row;

  }

}

.cart-item button {

  font: inherit;

  font-weight: bold;

  font-size: 1.25rem;

  color: #8a2b06;

  border: 1px solid #8a2b06;

  width: 3rem;

  text-align: center;

  border-radius: 6px;

  background-color: transparent;

  cursor: pointer;

  margin-left: 1rem;

  margin: 0.25rem;

}

.cart-item button:hover,

.cart-item button:active {

  background-color: #8a2b06;

  color: white;

}

MealItem.js

import { useContext } from 'react';

import MealItemForm from './MealItemForm';

import classes from './MealItem.module.css';

import CartContext from '../../../Store/cart-context';

const MealItem = (props) => {

  const cartCtx = useContext(CartContext);

  const price = `${props.price.toFixed(2)}`;

  const addToCartHandler = amount => {

    cartCtx.addItem({

      id: props.id,

      name: props.name,

      amount: amount,

      price: props.price

    });

  };

  return (

    <li className={classes.meal}>

      <div>

        <h3>{props.name}</h3>

        <div className={classes.description}>{props.description}</div>

        <div className={classes.price}>{price}</div>

      </div>

      <div>

        <MealItemForm id={props.id} onAddToCart={addToCartHandler} />

      </div>

    </li>

  );

};

export default MealItem;

MealItem.module.css

.meal {

  display: flex;

  justify-content: space-between;

  margin: 1rem;

  padding-bottom: 1rem;

  border-bottom: 1px solid #ccc;

}

.meal h3 {

  margin: 0 0 0.25rem 0;

}

.description {

  font-style: italic;

}

.price {

  margin-top: 0.25rem;

  font-weight: bold;

  color: #ad5502;

  font-size: 1.25rem;

}

MealItemForm.js:-

import { useRef, useState } from 'react';

import Input from '../../UI/Input';

import classes from './MealItemForm.module.css';

const MealItemForm = (props) => {

  const [amountIsValid, setAmountIsValid] = useState(true);

  const amountInputRef = useRef();

  const submitHandler = (event) => {

    event.preventDefault();

    const enteredAmount = amountInputRef.current.value;

    const enteredAmountNumber = +enteredAmount;

    if (

      enteredAmount.trim().length === 0 ||

      enteredAmountNumber < 1 ||

      enteredAmountNumber > 5

    ) {

      setAmountIsValid(false);

      return;

    }

    props.onAddToCart(enteredAmountNumber);

  };

  return (

    <form className={classes.form} onSubmit={submitHandler}>

      <Input

        ref={amountInputRef}

        label='Amount'

        input={{

          id: 'amount\_' + props.id,

          type: 'number',

          min: '1',

          max: '5',

          step: '1',

          defaultValue: '1',

        }}

      />

      <button>+ Add</button>

      {!amountIsValid && <p>Please enter a valid amount (1-5).</p>}

    </form>

  );

};

export default MealItemForm;

MealItemForm.module.css:-

.form {

  text-align: right;

}

.form button {

  font: inherit;

  cursor: pointer;

  background-color: #276bd8;

  border: 1px solid #8a2b06;

  color: white;

  padding: 0.25rem 2rem;

  border-radius: 20px;

  font-weight: bold;

}

.form button:hover,

.form button:active {

  background-color: #641e03;

  border-color: #641e03;

}

AvailableMeals.js:-

import Card from '../UI/Card';

import MealItem from './MealItem/MealItem';

import classes from './AvailableMeals.module.css';

const DUMMY\_MEALS = [

  {

    id: 'm1',

    name: 'Biryani',

    description: 'Chicken Dum Biryani',

    price: 180.00,

  },

  {

    id: 'm2',

    name: 'France Biryani',

    description: 'Nellore Special!',

    price: 250.00,

  },

  {

    id: 'm3',

    name: 'Mutton Biryani',

    description: 'Delicious Dum Biryani',

    price: 360.00,

  },

  {

    id: 'm4',

    name: 'Fish Gravy',

    description: 'Natural and Fresh Fish Meals...',

    price: 190.00,

  },

];

const AvailableMeals = () => {

  const mealsList = DUMMY\_MEALS.map((meal) => (

    <MealItem

      key={meal.id}

      id={meal.id}

      name={meal.name}

      description={meal.description}

      price={meal.price}

    />

  ));

  return (

    <section className={classes.meals}>

      <Card>

        <ul>{mealsList}</ul>

      </Card>

    </section>

  );

};

export default AvailableMeals;

AvailableMeals.module.css:-

.meals {

  max-width: 60rem;

  width: 90%;

  margin: 2rem auto;

  animation: meals-appear 1s ease-out forwards;

}

.meals ul {

  list-style: none;

  margin: 0;

  padding: 0;

}

@keyframes meals-appear {

  from {

    opacity: 0;

    transform: translateY(3rem);

  }

  to {

    opacity: 1;

    transform: translateY(0);

  }

}

Mels.js:-

import { Fragment } from 'react';

import MealsSummary from './MealsSummary';

import AvailableMeals from './AvailableMeals';

const Meals = () => {

  return (

    <Fragment>

      <MealsSummary />

      <AvailableMeals />

    </Fragment>

  );

};

export default Meals;

MealsSummary.js:-

import classes from './MealsSummary.module.css';

const MealsSummary = () => {

  return (

    <section className={classes.summary}>

      <h2>Delicious Food, Delivered To You</h2>

      <p>

        Choose your favorite meal from our broad selection of available meals

        and enjoy a delicious lunch or dinner at home.

      </p>

      <p>

        All our meals are cooked with high-quality ingredients, just-in-time and

        of course by experienced chefs!

      </p>

    </section>

  );

};

export default MealsSummary;

MealsSummary.module.css:-

.summary {

  text-align: center;

  max-width: 45rem;

  width: 90%;

  margin: auto;

  margin-top: -10rem;

  position: relative;

  background-color: #383838;

  color: white;

  border-radius: 14px;

  padding: 1rem;

  box-shadow: 0 1px 18px 10px rgba(0, 0, 0, 0.25);

}

.summary h2 {

  font-size: 2rem;

  margin-top: 0;

}

UI

Card.js:-

import classes from './Card.module.css';

const Card = props => {

  return <div className={classes.card}>{props.children}</div>

};

export default Card;

Card.module.css:-

.card {

  padding: 1rem;

  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.25);

  border-radius: 14px;

  background-color: white;

}

Input.js:-

import React from 'react';

import classes from './Input.module.css';

const Input = React.forwardRef((props, ref) => {

  return (

    <div className={classes.input}>

      <label htmlFor={props.input.id}>{props.label}</label>

      <input ref={ref} {...props.input} />

    </div>

  );

});

export default Input;

Input.module.css:-

.input {

  display: flex;

  align-items: center;

  margin-bottom: 0.5rem;

}

.input label {

  font-weight: bold;

  margin-right: 1rem;

}

.input input {

  width: 3rem;

  border-radius: 5px;

  border: 1px solid #ccc;

  font: inherit;

  padding-left: 0.5rem;

}

Modal.js:-

import { Fragment } from 'react';

import ReactDOM from 'react-dom';

import classes from './Modal.module.css';

const Backdrop = (props) => {

  return <div className={classes.backdrop} onClick={props.onClose}/>;

};

const ModalOverlay = (props) => {

  return (

    <div className={classes.modal}>

      <div className={classes.content}>{props.children}</div>

    </div>

  );

};

const portalElement = document.getElementById('overlays');

const Modal = (props) => {

  return (

    <Fragment>

      {ReactDOM.createPortal(<Backdrop onClose={props.onClose} />, portalElement)}

      {ReactDOM.createPortal(

        <ModalOverlay>{props.children}</ModalOverlay>,

        portalElement

      )}

    </Fragment>

  );

};

export default Modal;

Modal.module.css:-

.backdrop {

  position: fixed;

  top: 0;

  left: 0;

  width: 100%;

  height: 100vh;

  z-index: 20;

  background-color: rgba(0, 0, 0, 0.75);

}

.modal {

  position: fixed;

  top: 20vh;

  left: 5%;

  width: 90%;

  background-color: white;

  padding: 1rem;

  border-radius: 14px;

  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.25);

  z-index: 30;

  animation: slide-down 300ms ease-out forwards;

}

@media (min-width: 768px) {

  .modal {

    width: 40rem;

    left: calc(50% - 20rem);

  }

}

@keyframes slide-down {

  from {

    opacity: 0;

    transform: translateY(-3rem);

  }

  to {

    opacity: 1;

    transform: translateY(0);

  }

}

Store:-

Cart-context.js:

import React from 'react';

const CartContext = React.createContext({

  items: [],

  totalAmount: 0,

  addItem: (item) => {},

  removeItem: (id) => {}

});

export default CartContext;

Cartprovider.js:-

import { useReducer } from 'react';

import CartContext from './cart-context';

const defaultCartState = {

  items: [],

  totalAmount: 0,

};

const cartReducer = (state, action) => {

  if (action.type === 'ADD') {

    const updatedTotalAmount =

      state.totalAmount + action.item.price \* action.item.amount;

    const existingCartItemIndex = state.items.findIndex(

      (item) => item.id === action.item.id

    );

    const existingCartItem = state.items[existingCartItemIndex];

    let updatedItems;

    if (existingCartItem) {

      const updatedItem = {

        ...existingCartItem,

        amount: existingCartItem.amount + action.item.amount,

      };

      updatedItems = [...state.items];

      updatedItems[existingCartItemIndex] = updatedItem;

    } else {

      updatedItems = state.items.concat(action.item);

    }

    return {

      items: updatedItems,

      totalAmount: updatedTotalAmount,

    };

  }

  if (action.type === 'REMOVE') {

    const existingCartItemIndex = state.items.findIndex(

      (item) => item.id === action.id

    );

    const existingItem = state.items[existingCartItemIndex];

    const updatedTotalAmount = state.totalAmount - existingItem.price;

    let updatedItems;

    if (existingItem.amount === 1) {

      updatedItems = state.items.filter(item => item.id !== action.id);

    } else {

      const updatedItem = { ...existingItem, amount: existingItem.amount - 1 };

      updatedItems = [...state.items];

      updatedItems[existingCartItemIndex] = updatedItem;

    }

    return {

      items: updatedItems,

      totalAmount: updatedTotalAmount

    };

  }

  return defaultCartState;

};

const CartProvider = (props) => {

  const [cartState, dispatchCartAction] = useReducer(

    cartReducer,

    defaultCartState

  );

  const addItemToCartHandler = (item) => {

    dispatchCartAction({ type: 'ADD', item: item });

  };

  const removeItemFromCartHandler = (id) => {

    dispatchCartAction({ type: 'REMOVE', id: id });

  };

  const cartContext = {

    items: cartState.items,

    totalAmount: cartState.totalAmount,

    addItem: addItemToCartHandler,

    removeItem: removeItemFromCartHandler,

  };

  return (

    <CartContext.Provider value={cartContext}>

      {props.children}

    </CartContext.Provider>

  );

};

export default CartProvider;

App.js:-

import { useState } from 'react';

import Header from './Components/Layout/Header';

import Meals from './Components/Meals/Meals';

import Cart from './Components/Cart/Cart';

import CartProvider from './Store/CartProvider';

function App() {

  const [cartIsShown, setCartIsShown] = useState(false);

  const showCartHandler = () => {

    setCartIsShown(true);

  };

  const hideCartHandler = () => {

    setCartIsShown(false);

  };

  return (

    <CartProvider>

      {cartIsShown && <Cart onClose={hideCartHandler} />}

      <Header onShowCart={showCartHandler} />

      <main>

        <Meals />

      </main>

    </CartProvider>

  );

}

export default App;

App.css:-

.App {

  text-align: center;

}

.App-logo {

  height: 40vmin;

  pointer-events: none;

}

@media (prefers-reduced-motion: no-preference) {

  .App-logo {

    animation: App-logo-spin infinite 20s linear;

  }

}

.App-header {

  background-color: #282c34;

  min-height: 100vh;

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

  font-size: calc(10px + 2vmin);

  color: white;

}

.App-link {

  color: #61dafb;

}

@keyframes App-logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

Index.js:-

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

Index.css:-

@import url('https://fonts.googleapis.com/css2?family=Noto+Sans+JP:wght@400;700&display=swap');

\* {

  box-sizing: border-box;

}

html {

  font-family: 'Noto Sans JP', sans-serif;

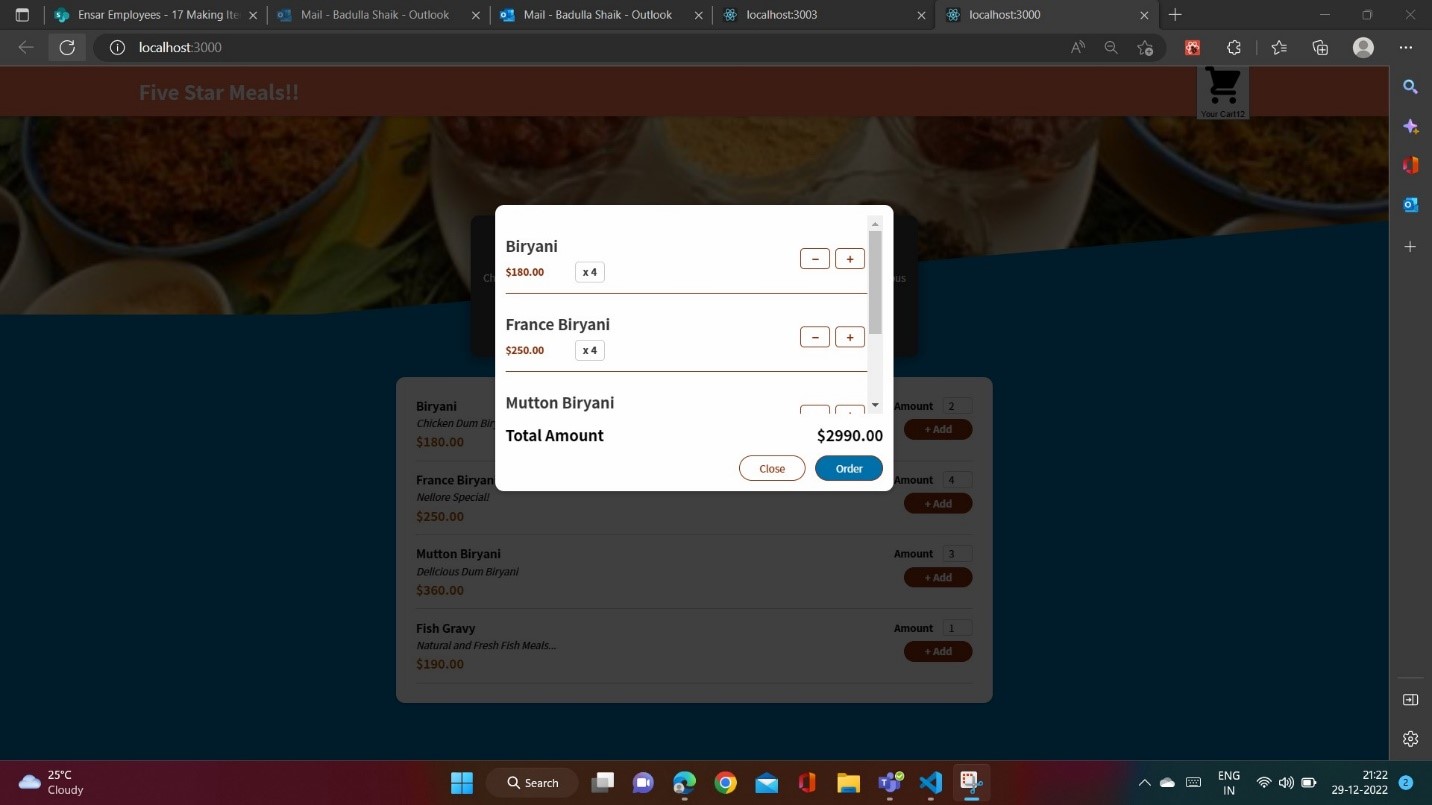
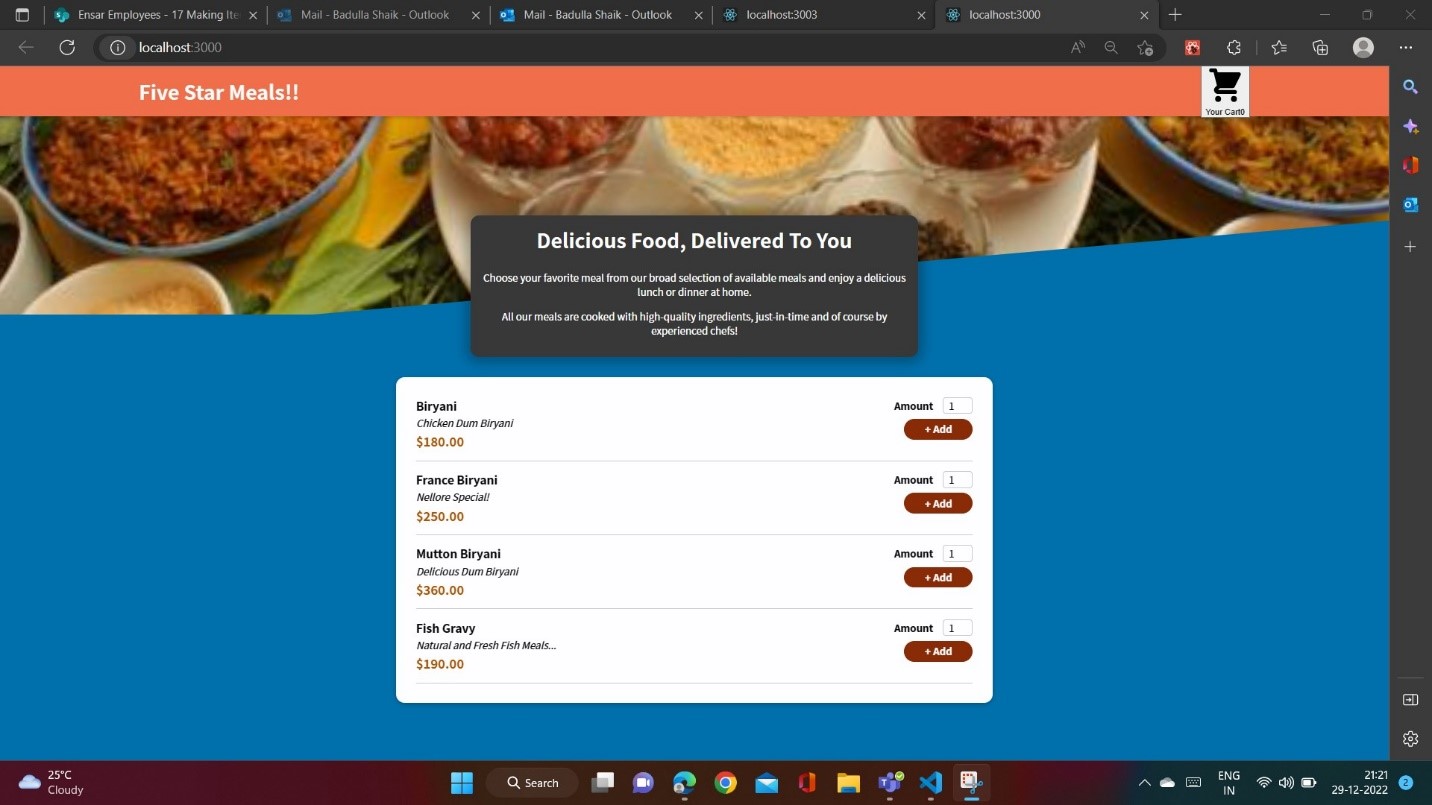
}

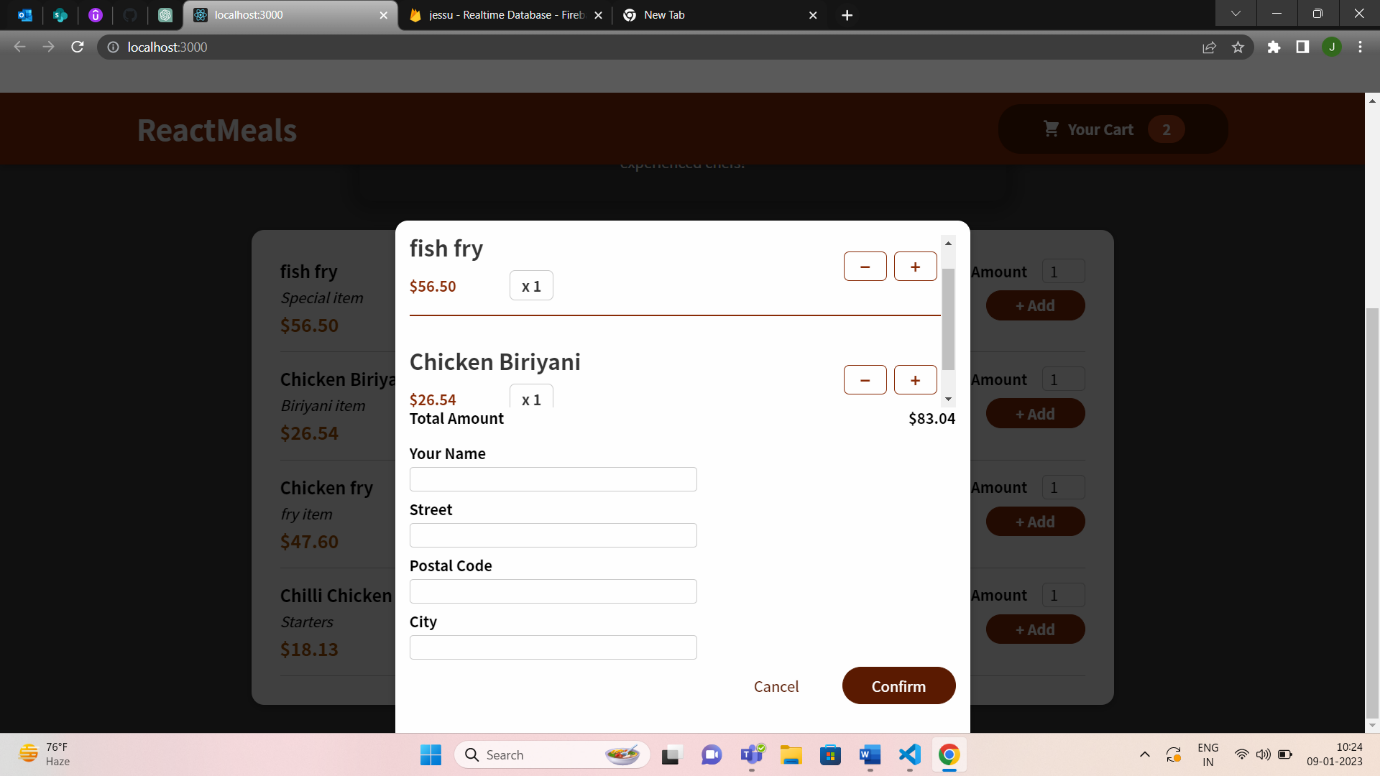
body {

  margin: 0;

  background-color: #0070AC;

}





Thank you