

Name: Nandhakumaran

Employee Id: 12225

Case Study: Coffee Shop Management System

W3H Analysis:

Case Study: Coffee Shop management	
1. What?	2. How?
<p>What are the modules required for Coffee shop management?</p> <p>Ans:</p> <ol style="list-style-type: none"><li>Admin Module</li><li>User Module</li><li>Shopkeeper Module</li></ol> <p>What are the functionalities required for Admin Module?</p> <p>Ans:</p> <ol style="list-style-type: none"><li>Login to the portal</li><li>Do CRUD Operations and Handle All Data</li></ol> <p>What are the functionalities required for User Module?</p> <p>Ans:</p> <ol style="list-style-type: none"><li>Login to the portal</li><li>Do CRUD Operations on User Data and Order</li><li>Order their Items</li></ol> <p>What are the functionalities required for Shopkeeper Module?</p> <ol style="list-style-type: none"><li>Login to the portal</li><li>Do CRUD Operations on Shopkeeper’s data and Menu Items</li><li>Provide Delivery Services</li></ol>	<p>Admin</p> <p>Login:</p> <p>Method 1: Login to the portal by email id and password</p> <p>Method 2: Login to the portal by Admin id and password</p> <p>CRUD:</p> <p>Method 1: Do CRUD Operations by non-Primary key value</p> <p>Method 2: Do CRUD Operations by Primary key values</p> <p>User</p> <p>Login:</p> <p>Method 1: Login to the portal by Username and password</p> <p>Method 2: Login to the portal by email Id and password</p> <p>CRUD:</p> <p>Method 1: Do Crud Operations by duplicatable values</p> <p>Method 2: Do Crud Operations by Unique values</p> <p>Order Items:</p> <p>Method 1: Dropdown the items and order it</p> <p>Method 2: See the Menu items in the menu and order the items</p> <p>Shopkeeper:</p> <p>Login:</p> <p>Method 1: Login to the portal by Shopkeeper name and password</p> <p>Method 2: Login to the portal by Shopkeeper id and password</p> <p>CRUD:</p> <p>Method 1: Do Crud by Duplicatable values</p> <p>Method 2: Do Crud by Unique values</p> <p>Provide Delivery Services:</p> <p>Method 1: Provide Delivery Services by Phone number</p> <p>Method 2: Provide Delivery Services by user given address.</p>
<p>Admin:</p> <p>Login:</p> <p>Method 2: Login to the portal by Admin id and password</p>	<p>Admin, User, Shopkeeper</p> <ol style="list-style-type: none"><li>Login with the Duplicatable values should be avoided.</li></ol>

<p>(Unique like Admin id is best for Login purposes)</p> <p>CRUD: Method 2: Do CRUD Operations by Primary key values (Unique values are best for CRUD Operations)</p> <p>User</p> <p>Login: Method 2: Login to the portal by email Id and password (Unique like email id is best for Login purposes)</p> <p>CRUD: Method 2: Do Crud Operations by Unique values (Unique values are best for CRUD Operations)</p> <p>Order Items: Method 2: See the Menu items in the menu and order the items (it will be easy to select the items rather than any other methods)</p> <p>Shopkeeper:</p> <p>Login: Method 2: Login to the portal by Shopkeeper id and password (Unique like Shopkeeper ID is best for Login purposes)</p> <p>CRUD: Method 2: Do Crud Operations by Unique values (Unique values are best for CRUD Operations)</p> <p>Provide Delivery Services: Method 2: Provide Delivery Services by user given address. (Provide delivery service by address will be the effective way the deliver the order)</p>	<p>2. Do CRUD Operations by Non-primary values also should be avoided</p> <p>3. Dropdown will not be a presentable way of show case the items</p> <p>4. Delivery services through the mobile numbers may irritate the users while asking location</p>
3. Why?	4. Why Not?

Algorithm for Admin:

Step 1: Start

**Step 2: Login to the Portal by Admin Id and password**

**Step 3: Do all CRUD Operations**

**Step 4: Handle all Data in Database**

**Step5: Logout**

**Step6: Stop**

**Algorithm for User:**

**Step1: Start**

**Step2: Login to the portal by email id and password**

**Step3: Do all crud Operations on their own data and order**

**Step4: Select the Items and place an order**

**Step5: give reviews and feedback**

**Step6: Logout**

**Step7: Stop**

**Algorithm for Shopkeeper:**

**Step1: Start**

**Step2: Login to the portal by shopkeeper id and password**

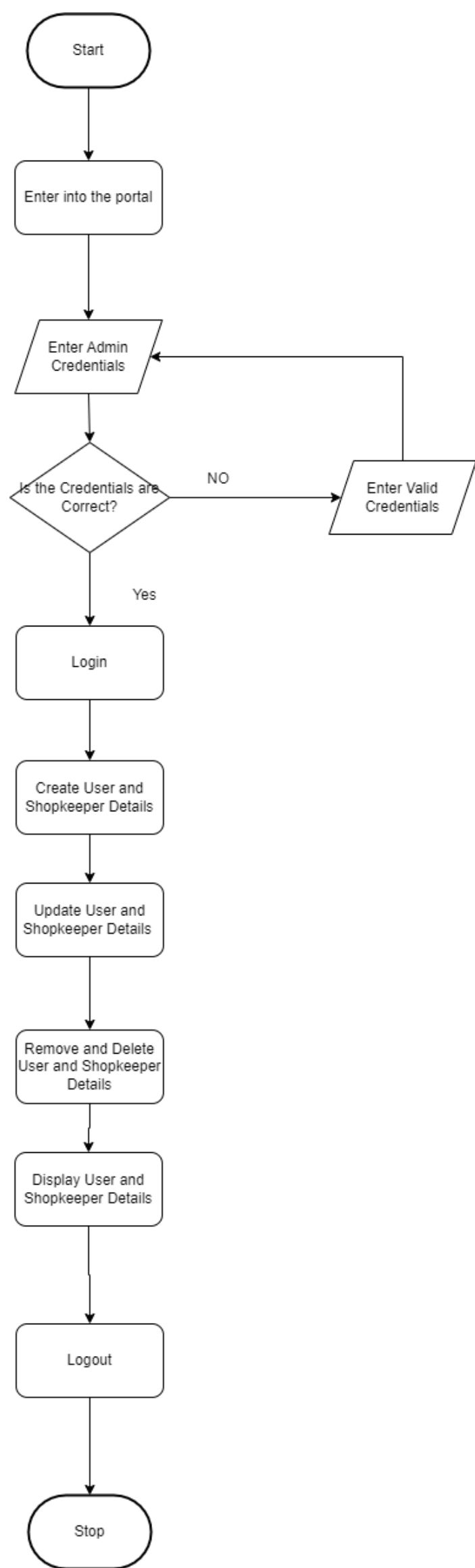
**Step3: Do all CRUD Operations on shopkeeper data and order**

**Step4: Provide Delivery Services**

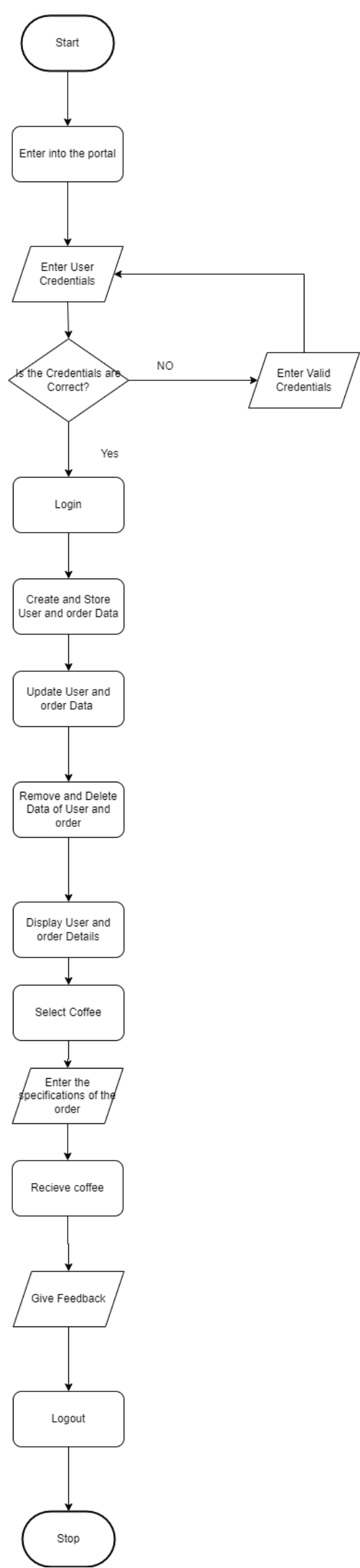
**Step5: Logout**

**Step6: Stop**

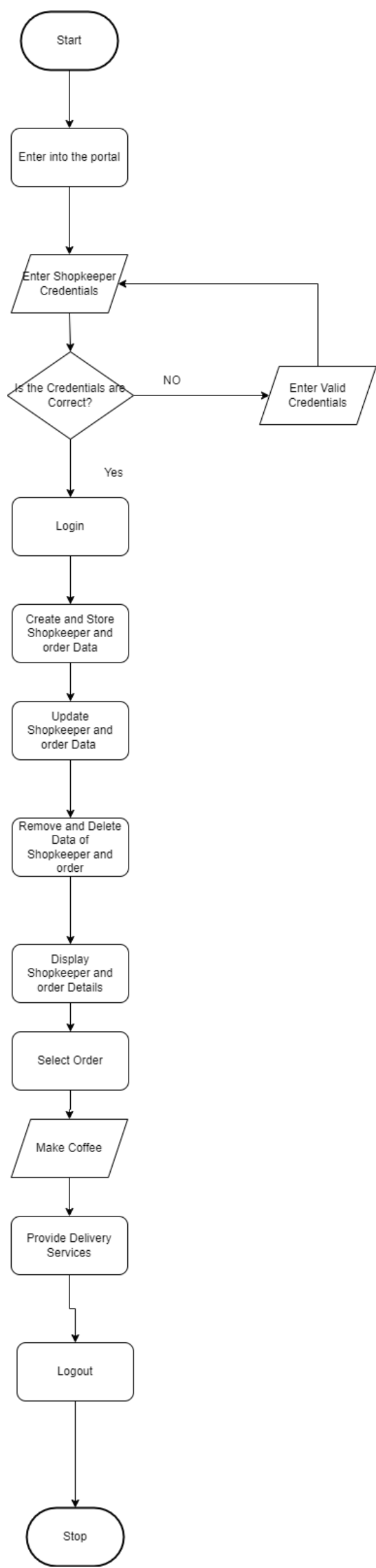
Flowchart for Admin:



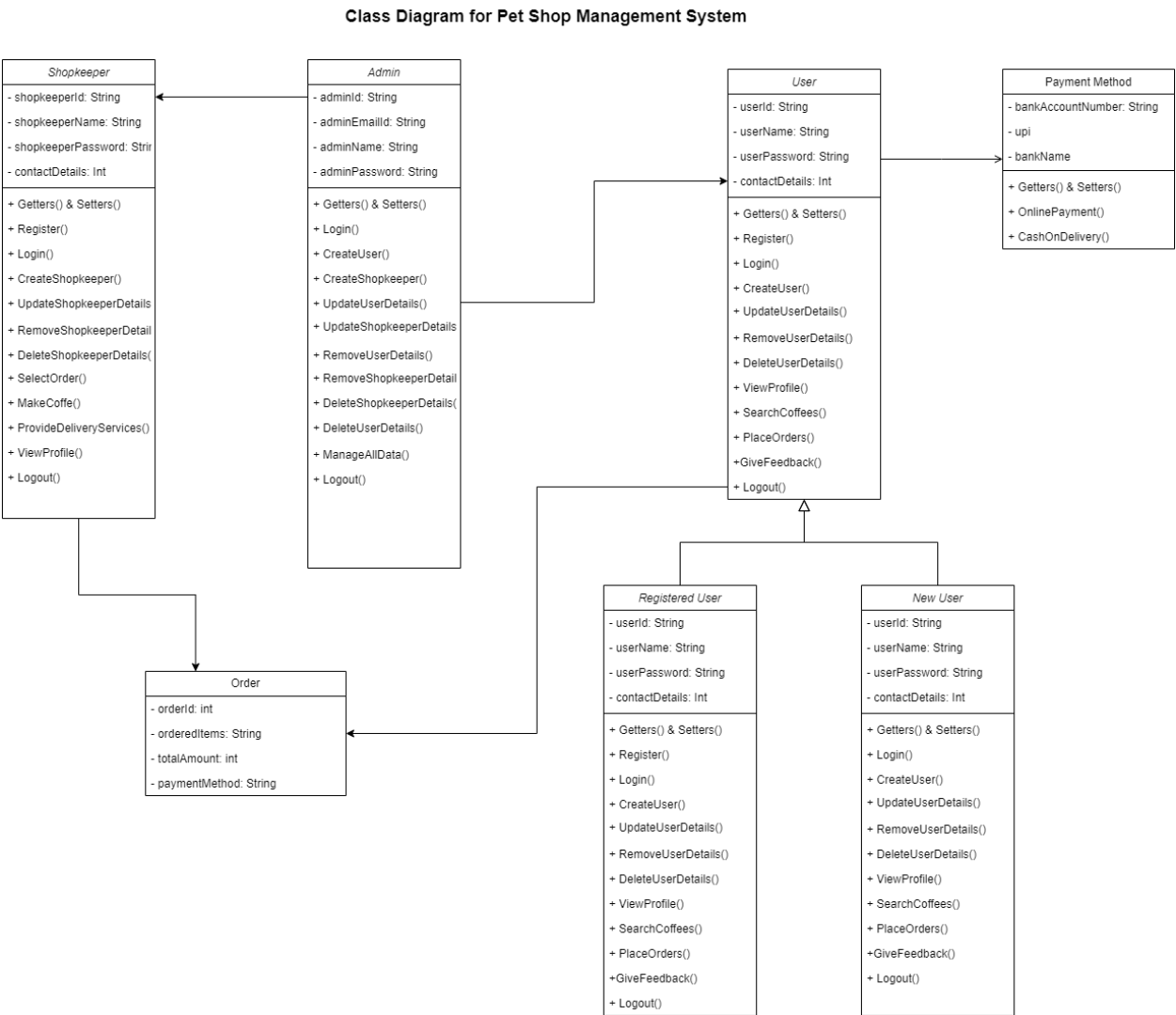
Flowchart for User:



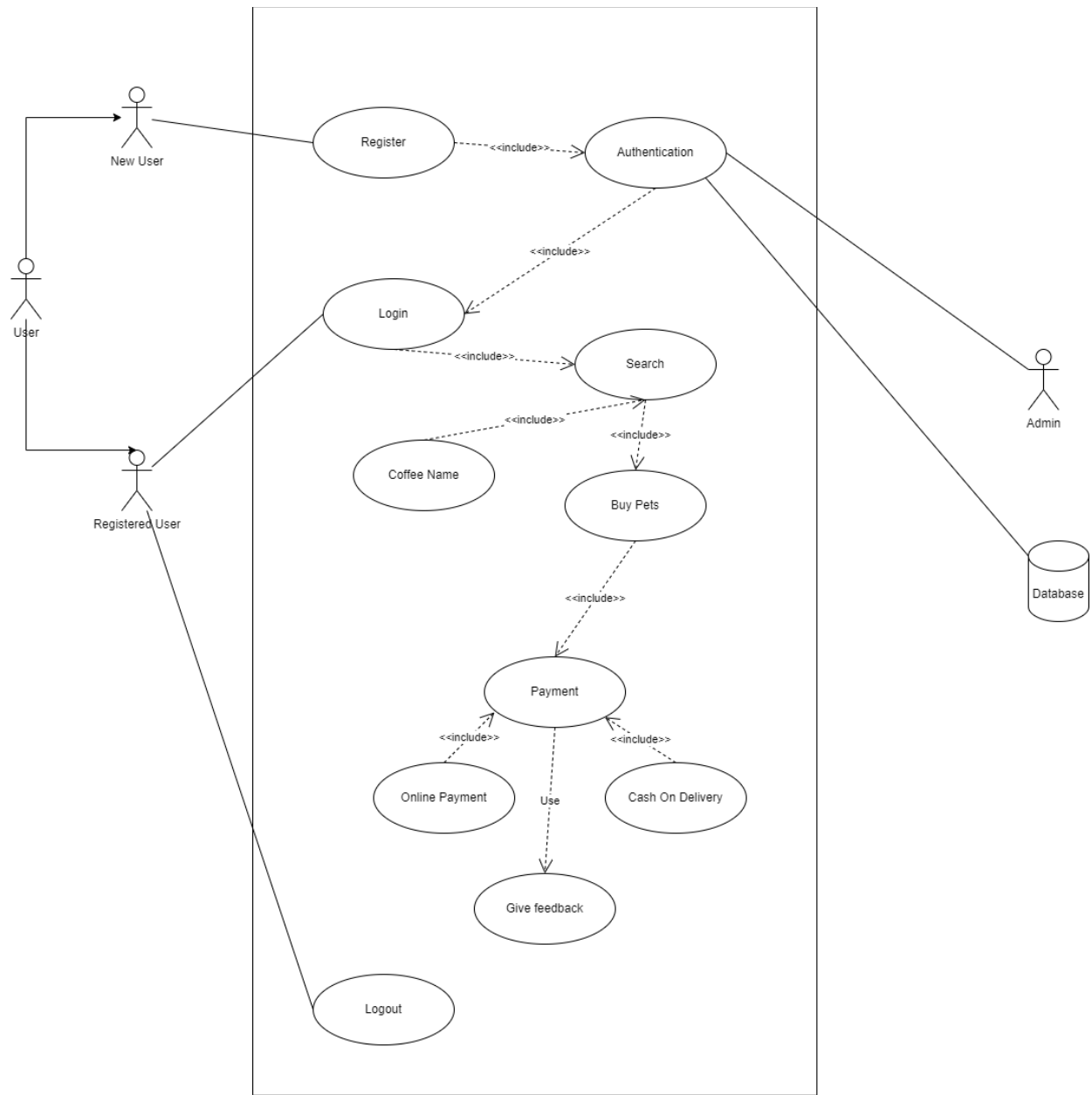
Flowchart for Shopkeeper:



Class Diagram for Coffee Shop Management System

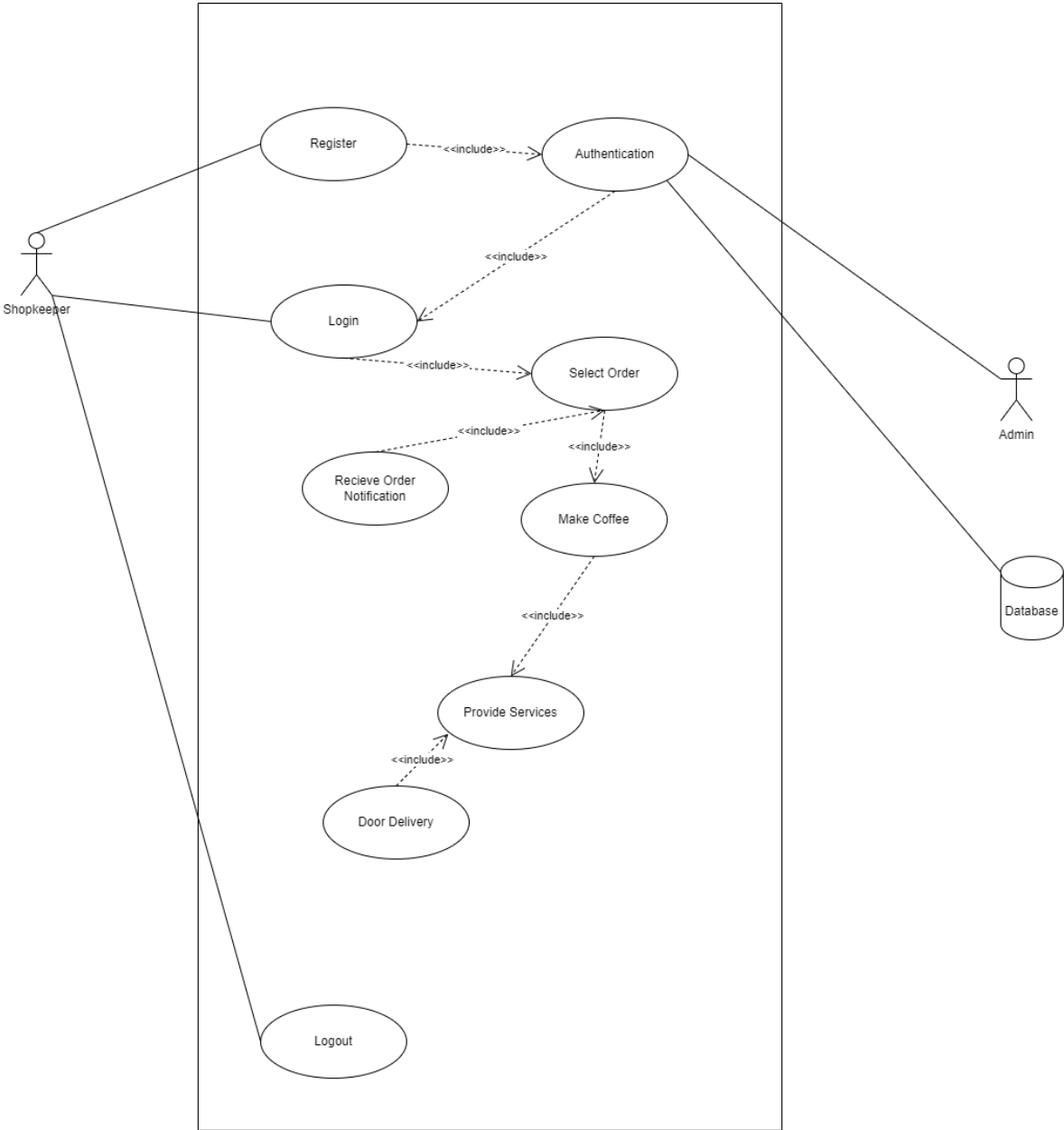


Use Case Diagram for User

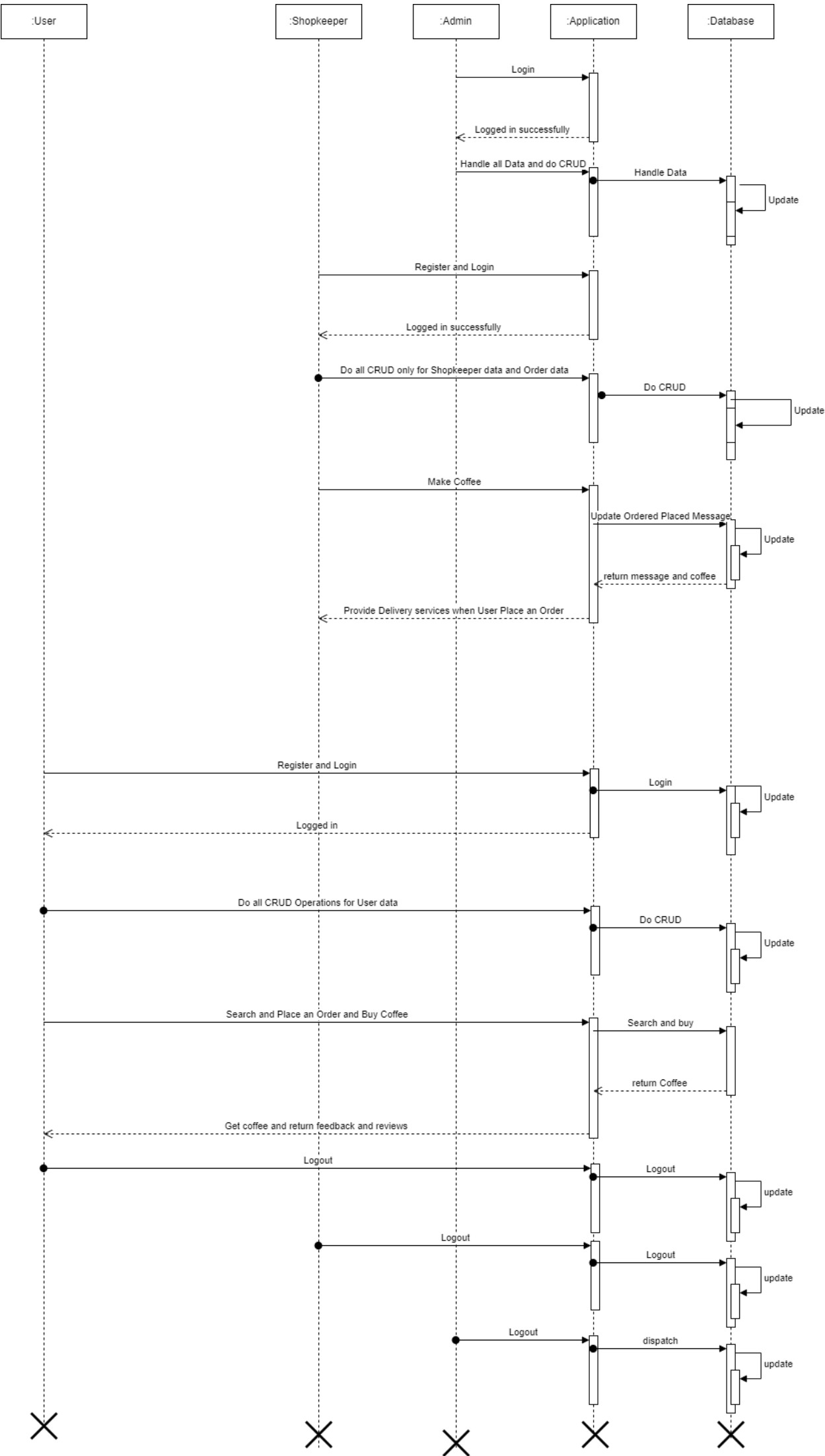




Use Case Diagram for Shopkeeper



Sequence Diagram for Coffee Shop Management System



Frontend Code

Pages:

```

import axios from "axios";
import React, { useState, useEffect } from "react";
import { useNavigate, useParams, Link } from "react-router-dom";
import "../App2.css";
import Navbar from "../components/Navbar";
import { Checkbox } from 'antd';

function Add() {
  const { id } = useParams();
  const [data, setData] = useState([]);
  const [selectedId, setSelectedId] = useState('');
  const [total, setTotal] = useState(0); // State to hold total amount
  const [selectedItems, setSelectedItems] = useState([]); // State to hold
    selected item labels
  const [paymentMethod, setPaymentMethod] = useState(''); // State to hold
    selected payment method

  const [inputData, setInputData] = useState({
    orderId: "",
    orderedItems: "", // Will store names of selected items
    totalAmount: 0, // Initialize total amount
    user: {
      id: "",
      name: "",
      emailid: "",
      address: "",
      phoneNumber: ""
    },
    payment: ""
  });

  const plainOptions = ['Cappuccino', 'Latte', 'Macho'];
  const options = [
    { label: 'Cappuccino', value: '100' },
    { label: 'Latte', value: '90' },
    { label: 'Macho', value: '80' }
  ];

  const naviget = useNavigate();

  useEffect(() => {
    axios
      .get("http://localhost:4444/user/all")
      .then((response) => setData(response.data))
      .catch((err) => console.log(err));
  }, []);

  const onChange = (checkedValues) => {
    let total = 0;
    let selectedItemsNames = [];
    checkedValues.forEach((value) => {
      const option = options.find((option) => option.label === value);
      if (option) {
        total += parseInt(option.value, 10);
        selectedItemsNames.push(option.label);
      }
    });

    setTotal(total); // Update total amount state
    setSelectedItems(selectedItemsNames); // Update selected items names
    setInputData({
      ...inputData,
      orderedItems: selectedItemsNames.join(', '),
      totalAmount: total
    }); // Update ordered items and total amount in form data
  };

  const handleChanges = (event) => {

```

```

        setSelectedId(event.target.value);

        axios
        .get("http://localhost:4444/user/" + event.target.value)
        .then((response) => {
            setInputData({
                ...inputData,
                user: response.data
            });
        })
        .catch((err) => {
            console.log(err);
        });
    });

    const handleChange = (event) => {
        setPaymentMethod(event.target.value); // Update selected payment method
        setInputData({
            ...inputData,
            payment: event.target.value
        }); // Update payment method in form data
    };

    const handleSubmit = (e) => {
        e.preventDefault();
        console.log(inputData); // Log input data before submission for verification

        // Validate form data
        if (!validateForm()) {
            alert("Please Enter Valid Inputs!!!");
            return;
        }

        axios
        .post("http://localhost:4444/order", inputData)
        .then((res) => {
            alert("Data added Successfully");
            naviget("/"); // Navigate to home page after successful submission
        })
        .catch((err) => console.error("Error submitting data:", err));
    };

    const validateForm = () => {
        const { user, payment } = inputData;
        if (user.name.trim() === "" || user.emailid.trim() === "" || payment.trim()
            === "") {
            return false;
        }
        return true;
    };

    return (
        <>
        <Navbar />

        <div
            id="add2"
            className="d-flex w-100 vh-100 justify-content-center align-items-center"
            style={{ backdropFilter: 'blur(5px)', backgroundColor: '#f9f9f9' }}
        >
        <div className="w-50 border bg-dark p-5" style={{ animation: 'fadeIn
            0.5s', border: '4px solid black', borderRadius: '30px' }}>
            <form onSubmit={handleSubmit}>
                <h1 style={{ marginBottom: '20px', color: 'fff' }}>Place Your
                    Order</h1>

                <div className="mb-3 bg-light p-3 " id="check">

```

```

<label htmlFor="items" className="form-label mr-4" style={{color:
    'black' }}>
    Items
</label>
<Checkbox.Group options={options} defaultValue={[]}
    onChange={onChange} />
</div>

<div className="mb-3">
<label htmlFor="selectId">Select a User ID:</label>
    <select id="selectId" onChange={handleChanges}>
        <option value="">Select an ID</option>
        {data.map((item) => (
            <option key={item.id} value={item.id}>
                {item.id}
            </option>
        ))}
    </select>
</div>

<div className="mb-3">
<label htmlFor="orderedItems" className="form-label">
    Ordered Items
</label>
    <input
        type="text"
        id="orderedItems"
        name="orderedItems"
        className="form-control"
        value={selectedItems.join(', ')} // Display selected items names
        readOnly
    />
</div>

<div className="mb-3">
<label htmlFor="totalAmount" className="form-label">
    Total Amount
</label>
    <input
        type="number"
        id="totalAmount"
        name="totalAmount"
        className="form-control"
        value={total}
        disabled
    />
</div>

<div className="mb-3">
<label htmlFor="payment" className="form-label">
    Payment
</label>
    <select
        id="payment"
        name="payment"
        className="form-control"
        onChange={handleChange}
        value={paymentMethod}
    >
        <option value="">Select Payment Method</option>
        <option value="cash">Cash</option>
        <option value="online_payment">Online Payment</option>
        <option value="coupons">Coupons</option>
    </select>
</div>

<button type="submit" className="btn btn-info" style={{ fontSize:
    '16px', padding: '10px 20px' }}>

```

```

        Submit
      </button>

      <Link to="/items" className="btn btn-primary">
        Menu
      </Link>
    </form>
  </div>
</div>
</>
);
}

export default Add;

```

```

import React, { useState } from "react";
import { useNavigate } from "react-router-dom";
import axios from "axios";
import Navbar from "../components/Navbar";

function AddUser() {
const [inputData, setInputData] = useState({
  name: "",
  emailid: "",
  address: "",
  phoneNumber: ""
});

const naviget = useNavigate();

const handleSubmit = (e) => {
  e.preventDefault();

  axios
    .post("http://localhost:4444/user", inputData)
    .then((res) => {
      alert("Data added Successfully");
      naviget("/"); // Navigate to home page
      console.log(res.data);
    })
    .catch((err) => {
      console.error("Error adding data:", err);
      alert("Failed to add data. Please try again.");
    });
});

return (
  <>
    <Navbar />
    <div
      id="add2"
      className="d-flex w-100 vh-100 justify-content-center align-items-center"
      style={{ backdropFilter: "blur(5px)", backgroundColor: "#f9f9f9" }}
    >
      <div
        className="w-50 border bg-dark p-5"
        style={{

```

```

        animation: "fadeIn 0.5s",
        border: "4px solid black",
        borderRadius: "30px"
      }}
    >
    <form onSubmit={handleSubmit}>
    <h1 style={{ marginBottom: "20px", color: "#fff" }}>
      ADD USER DATA
    </h1>
    <div className="mb-3">
    <label htmlFor="basicSalary" className="form-label">
      Name
    </label>
    <input
      type="text"
      name="name"
      className="form-control"
      value={inputData.name}
      onChange={(e) =>
        setInputData({ ...inputData, name: e.target.value })
      }
      required
    />
    </div>
    <div className="mb-3">
    <label htmlFor="hra" className="form-label">
      Email ID
    </label>
    <input
      type="email"
      name="emailid"
      className="form-control"
      value={inputData.emailid}
      onChange={(e) =>
        setInputData({ ...inputData, emailid: e.target.value })
      }
      required
    />
    </div>
    <div className="mb-3">
    <label htmlFor="ta" className="form-label">
      Address
    </label>
    <input
      type="text"
      name="address"
      className="form-control"
      value={inputData.address}
      onChange={(e) =>
        setInputData({ ...inputData, address: e.target.value })
      }
      required
    />
    </div>
    <div className="mb-3">
    <label htmlFor="ma" className="form-label">
      Phone Number
    </label>
    <input
      type="tel"
      name="phoneNumber"
      className="form-control"
      value={inputData.phoneNumber}
      onChange={(e) =>
        setInputData({ ...inputData, phoneNumber: e.target.value })
      }
      required
    />
    </div>
  
```



```
        <button
          type="submit"
          className="btn btn-info"
          style={{ fontSize: "16px", padding: "10px 20px" }}
        >
          Submit
        </button>
      </form>
    </div>
  </div>
</>
);
}
```

export default AddUser;

```
    div #app2{
      margin-top: 40px;
      text-align: center;

    }
    table{
      border-radius: 10px;
    }

    div #add2{
      background: url('https://images.pexels.com/photos/7130478/pexels-photo-7130478.jpeg?auto=compress&cs=tinysrgb&w=600');
      background:cover no-repeat fixed center;
      background-size: 100% 100%;

    }
    div #body{
      /* background: url('https://images.pexels.com/photos/1076885/pexels-photo-1076885.jpeg?auto=compress&cs=tinysrgb&w=600'); */
      background:cover no-repeat fixed center;
      background-size: 100% 100%;

    }
    div #edit2{

      background: url('https://images.pexels.com/photos/7130478/pexels-photo-7130478.jpeg?auto=compress&cs=tinysrgb&w=600');
      background:cover no-repeat fixed center;
      background-size: 100% 100%;
    }
    div{
      font-family: Verdana, Geneva, Tahoma, sans-serif;
    }

    label{
      color: #ffff;
    }

    #selectId{
      padding: 10px;
      border-radius: 5px;
    }
  }
```

```
    #check{
      border-radius: 10px;
    }
```

```
    #back{
      margin-left: 10%;
    }
```

```
import axios from "axios";
import React, { useState, useEffect } from "react";
import { useNavigate, useParams } from "react-router-dom";
import "../App2.css";
import Navbar from "../components/Navbar";
import { Checkbox } from 'antd';

function Update() {
  const { id } = useParams();
  const [data, setData] = useState([]);
  const [selectedId, setSelectedId] = useState('');
  const [total, setTotal] = useState(0); // State to hold total amount
  const [selectedItems, setSelectedItems] = useState([]); // State to hold
    selected item labels
  const [paymentMethod, setPaymentMethod] = useState(''); // State to hold
    selected payment method
  const [inputData, setInputData] = useState({
    id: "", // Initialize id state for the order ID
    orderedItems: "", // Will store names of selected items
    totalAmount: 0, // Initialize total amount
    user: {
      id: "",
      name: "",
      emailid: "",
      address: "",
      phoneNumber: ""
    },
    payment: ""
  });

  const plainOptions = ['Cappuccino', 'Latte', 'Macho'];
  const options = [
    { label: 'Cappuccino', value: '100' },
    { label: 'Latte', value: '90' },
    { label: 'Macho', value: '80' }
  ];

  const navigate = useNavigate();

  useEffect(() => {
    // Fetch order details based on ID when component mounts
    axios
      .get(`http://localhost:4444/order/${id}`) // Assuming endpoint to fetch
        order details
      .then((response) => {
        const { orderId, orderedItems, totalAmount, payment, user } =
          response.data;
        setInputData({
          orderId,
          orderedItems,
          totalAmount,
          payment,
          user: {
            id: user.id,
            name: user.name,
            emailid: user.emailid,
            address: user.address,
            phoneNumber: user.phoneNumber
          }
        });
      });
  });
}
```

```

        }
    });

    setTotal(totalAmount); // Set total amount from fetched data
    setSelectedItems(orderedItems.split(', ')); // Set selected items from
    fetched data
    setPaymentMethod(payment); // Set payment method from fetched data
    })
    .catch((err) => console.log(err));

    // Fetch all users to populate the dropdown
    axios
    .get("http://localhost:4444/user/all")
    .then((response) => setData(response.data))
    .catch((err) => console.log(err));
    }, [id]);

    const onChange = (checkedValues) => {
        let total = 0;
        let selectedItemsNames = [];
        checkedValues.forEach((value) => {
    const option = options.find((option) => option.label === value);
        if (option) {
            total += parseInt(option.value, 10);
            selectedItemsNames.push(option.label);
        }
    });

    setTotal(total); // Update total amount state
    setSelectedItems(selectedItemsNames); // Update selected items names
    setInputData({
        ...inputData,
        orderedItems: selectedItemsNames.join(', '),
        totalAmount: total
    }); // Update ordered items and total amount in form data
    });

    const handleChanges = (event) => {
        const selectedUserId = event.target.value;
        setSelectedId(selectedUserId); // Update selected user ID
        axios
        .get(`http://localhost:4444/user/${selectedUserId}`)
        .then((response) => {
    const { id, name, emailid, address, phoneNumber } = response.data;
        setInputData({
            ...inputData,
            user: { id, name, emailid, address, phoneNumber }
        });
    })
    .catch((err) => console.log(err));
    });

    const handleChange = (event) => {
    setPaymentMethod(event.target.value); // Update selected payment method
        setInputData({
            ...inputData,
            payment: event.target.value
        }); // Update payment method in form data
    });

    const handleSubmit = (e) => {
        e.preventDefault();
        console.log(inputData); // Log input data before submission for verification

        // Validate form data
        if (!validateForm()) {
            alert("Please Enter Valid Inputs!!!");
            return;
        }
    }

```

```

        axios
        .put(`http://localhost:4444/order/${id}`, inputData) // Assuming endpoint
        to update order
        .then((res) => {
            alert("Data updated Successfully");
            navigate("/"); // Navigate to home page after successful submission
        })
        .catch((err) => console.error("Error submitting data:", err));
    };

    const validateForm = () => {
        const { user, payment } = inputData;
        if (user.name.trim() === "" || user.emailid.trim() === "" || payment.trim()
            === "") {
            return false;
        }
        return true;
    };

    return (
        <>
        <Navbar />
        <div
            id="update"
            className="d-flex w-100 vh-100 justify-content-center align-items-center"
            style={{ backdropFilter: 'blur(5px)', backgroundColor: '#f9f9f9' }}
        >
            <div className="w-50 border bg-dark p-5" style={{ animation: 'fadeIn
                0.5s', border: '4px solid black', borderRadius: '30px' }}>
                <form onSubmit={handleSubmit}>
                    <h1 style={{ marginBottom: '20px', color: '#fff', fontFamily: 'Arial,
                        sans-serif' }}>
                        UPDATE ORDER DATA
                    </h1>

                    <div className="mb-3">
                        <label htmlFor="id">ID:</label>
                        <input
                            type="number"
                            disabled
                            name="orderId"
                            className="form-control"
                            value={inputData.orderId}
                        />
                    </div>

                    <div className="mb-3">
                        <label htmlFor="items" className="form-label">
                            Items
                        </label>
                        <Checkbox.Group options={plainOptions} value={selectedItems}
                            onChange={onChange} />
                    </div>

                    <div className="mb-3">
                        <label htmlFor="selectId">Select a User ID:</label>
                        <select id="selectId" onChange={handleChanges} value={selectedId}>
                            <option value="">Select an ID</option>
                            {data.map((item) => (
                                <option key={item.id} value={item.id}>
                                    {item.id}
                                </option>
                            ))}
                        </select>
                    </div>
                </form>
            </div>
        </>
    );

```

```

        <div className="mb-3">
          <label htmlFor="orderedItems" className="form-label">
            Ordered Items
          </label>
          <input
            type="text"
            id="orderedItems"
            name="orderedItems"
            className="form-control"
            value={selectedItems.join(', ')}
            readOnly
          />
        </div>

        <div className="mb-3">
          <label htmlFor="totalAmount" className="form-label">
            Total Amount
          </label>
          <input
            type="number"
            id="totalAmount"
            name="totalAmount"
            className="form-control"
            value={total}
            disabled
          />
        </div>

        <div className="mb-3">
          <label htmlFor="payment" className="form-label">
            Payment
          </label>
          <select
            id="payment"
            name="payment"
            className="form-control"
            onChange={handleChange}
            value={paymentMethod}
          >
            <option value="">Select Payment Method</option>
            <option value="cash">Cash</option>
            <option value="online_payment">Online Payment</option>
            <option value="coupons">Coupons</option>
          </select>
        </div>

        <button type="submit" className="btn btn-info" style={{ fontSize:
          '16px', padding: '10px 20px' }}>
          Update Order
        </button>
      </form>
    </div>
  </div>
  </>
  );
}

export default Update;

```

```

import axios from "axios";
import React, { useEffect, useState } from "react";
import { useNavigate, useParams } from "react-router-dom";
import "../App2.css";
import Navbar from "../components/Navbar";

function EditUser() {
  const { id } = useParams();
  const [selectedId, setSelectedId] = useState('');
  const [inputData, setInputData] = useState([]);
  const naviget = useNavigate();

  useEffect(() => {
    axios
      .get("http://localhost:4444/user/"+id)
      .then((response) => setInputData(response.data))
      .catch((err) => console.log(err));
  }, []);

  const handleSubmit = (e) => {
    e.preventDefault();
    // const result = validateValues(inputData);

    // if (result) {
    console.log(inputData);

    axios
      .post("http://localhost:4444/user", inputData)
      .then((res) => {
        alert("Data Updated Successfully");
        naviget("/"); // Navigate to home page
        console.log(res.data);
      })
      .catch((err) => console.log(err));
    // } else {
    //   alert("Please Enter Valid Inputs!!!");
    // }
  };

  // const validateValues = (inputData) => {
  //   if (inputData.name.length === 0) {
  //     alert("Please enter Employee Name !!! ");
  //     return false;
  //   } else if (inputData.department.length === 0) {
  //     alert("Please enter Employee Department !!!");
  //     return false;
  //   } else if (inputData.designation.length === 0) {
  //     alert("Please enter Employee Designation !!!");
  //     return false;
  //   } else {
  //     return true;
  //   }
  // };

  return (
    <div>
      <Navbar />

```

```

        <div
          id="edit2"
          className="d-flex w-100 vh-100 justify-content-center align-items-center"
          style={{ backdropFilter: 'blur(5px)', backgroundColor: '#f9f9f9', padding:
            '20px' }}
        >
<div className="w-50 border bg-dark p-5 mt-5" style={{ animation: 'fadeIn
  0.5s', border: '4px solid black', borderRadius: '30px' }}>
  <form onSubmit={handleSubmit}>
    <h1 style={{ marginBottom: '20px', color: '#fff', fontFamily: 'Arial,
      sans-serif' }}>
      UPDATE USER DATA
    </h1>
    <div className="mb-3">
      <label htmlFor="id"> Order ID:</label>
      <input
        type="number"
        disabled
        name="id"
        className="form-control"
        value={inputData.id}
        onChange={(e) => setInputData({ ...inputData, id:
          e.target.value })}
      />
    </div>
    <div className="mb-3">
      <label htmlFor="basicSalary" className="form-label">
        Name
      </label>
      <input
        type="text"
        name="name"
        className="form-control"
        value={inputData.name}
        onChange={(e) =>
          setInputData({ ...inputData, name: e.target.value })
        }
        required
      />
    </div>
    <div className="mb-3">
      <label htmlFor="hra" className="form-label">
        Email ID
      </label>
      <input
        type="email"
        name="emailid"
        className="form-control"
        value={inputData.emailid}
        onChange={(e) =>
          setInputData({ ...inputData, emailid: e.target.value })
        }
        required
      />
    </div>
    <div className="mb-3">
      <label htmlFor="ta" className="form-label">
        Address
      </label>
      <input
        type="text"
        name="address"
        className="form-control"
        value={inputData.address}
        onChange={(e) =>
          setInputData({ ...inputData, address: e.target.value })
        }
        required
      />
    </div>
  </form>
</div>

```

```

        </div>
        <div className="mb-3">
          <label htmlFor="ma" className="form-label">
            Phone Number
          </label>
          <input
            type="tel"
            name="phoneNumber"
            className="form-control"
            value={inputData.phoneNumber}
            onChange={(e) =>
              setInputData({ ...inputData, phoneNumber: e.target.value })
            }
            required
          />
        </div>

        <button className="btn btn-info" style={{ fontSize: '16px', padding:
          '10px 20px' }}>
          Update
        </button>
      </form>
    </div>
  </div>

  </div>
  );
}

export default EditUser;

```

```

import Navbar from "../components/Navbar";

function Signin() {
  return (
    <><Navbar/>
    <main className="w-full h-screen flex flex-col items-center justify-
      center px-4">
      <div className="max-w-sm w-full text-gray-600 space-y-5">
        <div className="text-center pb-8">
          
          <div className="mt-5">
            <h3 className="text-gray-800 text-2xl font-bold sm:text-
              3xl">Log in to your account</h3>
          </div>
        </div>
        <form
          onSubmit={(e) => e.preventDefault()}
          className="space-y-5"
        >
          <div>
            <label className="font-medium">
              Email
            </label>
            <input
              type="email"
              required

```



```

        className="w-full mt-2 px-3 py-2 text-gray-500 bg-
transparent outline-none border focus:border-indigo-600 shadow-sm rounded-lg"
      />
    </div>
    <div>
      <label className="font-medium">
        Password
      </label>
      <input
        type="password"
        required
        className="w-full mt-2 px-3 py-2 text-gray-500 bg-
transparent outline-none border focus:border-indigo-600 shadow-sm rounded-lg"
      />
    </div>
    <div className="flex items-center justify-between text-sm">
      <div className="flex items-center gap-x-3">
        <input type="checkbox" id="remember-me-checkbox"
className="checkbox-item peer hidden" />
        <label
          htmlFor="remember-me-checkbox"
          className="relative flex w-5 h-5 bg-white peer-
checked:bg-indigo-600 rounded-md border ring-offset-2 ring-indigo-600 duration-
150 peer-active:ring cursor-pointer after:absolute after:inset-x-0 after:top-
[3px] after:m-auto after:w-1.5 after:h-2.5 after:border-r-2 after:border-b-2
after:border-white after:rotate-45"
        >
          </label>
        <span>Remember me</span>
      </div>
      <a href="javascript:void(0)" className="text-center text-
indigo-600 hover:text-indigo-500">Forgot password?</a>
    </div><a href="/">
      <button
        className="w-full px-4 py-2 text-white font-medium bg-
indigo-600 hover:bg-indigo-500 active:bg-indigo-600 rounded-lg duration-150"
      >
        Sign in
      </button></a>
    </form>
    <button className="w-full flex items-center justify-center gap-x-
3 py-2.5 border rounded-lg text-sm font-medium hover:bg-gray-50 duration-150
active:bg-gray-100">
      <svg className="w-5 h-5" viewBox="0 0 48 48" fill="none"
xmlns="http://www.w3.org/2000/svg">
        <g clip-path="url(#clip0_17_40)">
          <path d="M47.532 24.5528C47.532 22.9214 47.3997
21.2811 47.1175 19.6761H24.48V28.9181H37.4434C36.9055 31.8988 35.177 34.5356
32.6461 36.2111V42.2078H40.3801C44.9217 38.0278 47.532 31.8547 47.532 24.5528Z"
fill="#4285F4" />
          <path d="M24.48 48.0016C30.9529 48.0016 36.4116
45.8764 40.3888 42.2078L32.6549 36.2111C30.5031 37.675 27.7252 38.5039 24.4888
38.5039C18.2275 38.5039 12.9187 34.2798 11.0139 28.6006H3.03296V34.7825C7.10718
42.8868 15.4056 48.0016 24.48 48.0016Z" fill="#34A853" />
          <path d="M11.0051 28.6006C9.99973 25.6199 9.99973
22.3922 11.0051 19.4115V13.2296H3.03298C-0.371021 20.0112 -0.371021 28.0009
3.03298 34.7825L11.0051 28.6006Z" fill="#FBBC04" />
          <path d="M24.48 9.49932C27.9016 9.44641 31.2086
10.7339 33.6866 13.0973L40.5387 6.24523C36.2 2.17101 30.4414 -0.068932 24.48
0.00161733C15.4055 0.00161733 7.10718 5.11644 3.03296 13.2296L11.005
19.4115C12.901 13.7235 18.2187 9.49932 24.48 9.49932Z" fill="#EA4335" />
        </g>
        <defs>
          <clipPath id="clip0_17_40">
            <rect width="48" height="48" fill="white" />
          </clipPath>
        </defs>
      </svg>

```

```

        </button>
        <p className="text-center">Don't have an account? <a
href="/signup" className="font-medium text-indigo-600 hover:text-indigo-500">Sign
up</a></p>
    </div>
</main></>
    )
}

export default Signin;

```

```

import Navbar from "../components/Navbar";

function Signup() {
    return (
        <><Navbar/>
        <main className="w-full flex">

            <div className="relative flex-1 hidden items-center justify-center h-
screen bg-gray-900 lg:flex">
                <div className="relative z-10 w-full max-w-md">
                    
                    <div className="mt-16 space-y-3">
                        <h3 className="text-white text-3xl font-bold">Start
Drinking and Do Nothing</h3>
                        <p className="text-gray-300">
Happy Customers and their Reviews
                        </p>
                        <div className="flex items-center -space-x-2 overflow-
hidden">
                            
                            
                            
                            
                            
                        <p className="text-sm text-gray-400 font-medium
translate-x-5">
Join 5.000+ users
                        </p>
                    </div>
                </div>
            </div>
            <div
className="absolute inset-0 my-auto h-[500px]"
style={{
background: "linear-gradient(152.92deg, rgba(192, 132,
252, 0.2) 4.54%, rgba(232, 121, 249, 0.26) 34.2%, rgba(192, 132, 252, 0.1)
77.55%)", filter: "blur(118px)"

```

```
    }}
  >

  </div>
</div>
<div className="flex-1 flex items-center justify-center h-screen">
  <div className="w-full max-w-md space-y-8 px-4 bg-white text-gray-600 sm:px-0">
    <div className="">
      
      <div className="mt-5 space-y-2">
        <h3 className="text-gray-800 text-2xl font-bold sm:text-3xl">Sign up</h3>
        <p className="">Already have an account? <a
href="/signin" className="font-medium text-indigo-600 hover:text-indigo-500">Log
in</a></p>
      </div>
    </div>
    <div className="grid grid-cols-3 gap-x-3">
      <button className="flex items-center justify-center py-2.5 border rounded-lg hover:bg-gray-50 duration-150 active:bg-gray-100">
        <svg className="w-5 h-5" viewBox="0 0 48 48"
          fill="none" xmlns="http://www.w3.org/2000/svg">
          <g clip-path="url(#clip0_17_40)">
            <path d="M47.532 24.5528C47.532 22.9214
47.3997 21.2811 47.1175 19.6761H24.48V28.9181H37.4434C36.9055 31.8988 35.177
34.5356 32.6461 36.2111V42.2078H40.3801C44.9217 38.0278 47.532 31.8547 47.532
24.5528Z" fill="#4285F4" />
            <path d="M24.48 48.0016C30.9529 48.0016
36.4116 45.8764 40.3888 42.2078L32.6549 36.2111C30.5031 37.675 27.7252 38.5039
24.4888 38.5039C18.2275 38.5039 12.9187 34.2798 11.0139
28.6006H3.03296V34.7825C7.10718 42.8868 15.4056 48.0016 24.48 48.0016Z"
          fill="#34A853" />
            <path d="M11.0051 28.6006C9.99973 25.6199
9.99973 22.3922 11.0051 19.4115V13.2296H3.03298C-0.371021 20.0112 -0.371021
28.0009 3.03298 34.7825L11.0051 28.6006Z" fill="#FBBC04" />
            <path d="M24.48 9.49932C27.9016 9.44641
31.2086 10.7339 33.6866 13.0973L40.5387 6.24523C36.2 2.17101 30.4414 -0.068932
24.48 0.00161733C15.4055 0.00161733 7.10718 5.11644 3.03296 13.2296L11.005
19.4115C12.901 13.7235 18.2187 9.49932 24.48 9.49932Z" fill="#EA4335" />
          </g>
          <defs>
            <clipPath id="clip0_17_40">
              <rect width="48" height="48" fill="white"
/>
            </clipPath>
          </defs>
        </svg>
      </button>
      <button className="flex items-center justify-center py-2.5 border rounded-lg hover:bg-gray-50 duration-150 active:bg-gray-100">
        <svg className="w-5 h-5" viewBox="0 0 48 48"
          fill="none" xmlns="http://www.w3.org/2000/svg">
          <path d="M15.095 43.5014C33.2083 43.5014 43.1155
28.4946 43.1155 15.4809C43.1155 15.0546 43.1155 14.6303 43.0867 14.2079C45.0141
12.8138 46.6778 11.0877 48 9.11033C46.2028 9.90713 44.2961 10.4294 42.3437
10.6598C44.3996 9.42915 45.9383 7.49333 46.6733 5.21273C44.7402 6.35994 42.6253
7.16838 40.4198 7.60313C38.935 6.02428 36.9712 4.97881 34.8324 4.6285C32.6935
4.27818 30.4988 4.64256 28.5879 5.66523C26.677 6.68791 25.1564 8.31187 24.2615
10.2858C23.3665 12.2598 23.1471 14.4737 23.6371 16.5849C19.7218 16.3885 15.8915
15.371 12.3949 13.5983C8.89831 11.8257 5.81353 9.33765 3.3408 6.29561C2.08146
8.4636 1.69574 11.0301 2.2622 13.4725C2.82865 15.9148 4.30468 18.0495 6.38976
19.4418C4.82246 19.3959 3.2893 18.9731 1.92 18.2092V18.334C1.92062 20.6077 2.7077
22.8112 4.14774 24.5707C5.58778 26.3303 7.59212 27.5375 9.8208 27.9878C8.37096
28.3832 6.84975 28.441 5.37408 28.1567C6.00363 30.1134 7.22886 31.8244 8.87848
33.0506C10.5281 34.2768 12.5197 34.9569 14.5747 34.9958C12.5329 36.6007 10.1946
37.7873 7.69375 38.4878C5.19287 39.1882 2.57843 39.3886 0 39.0777C4.50367 41.9677
9.74385 43.5007 15.095 43.4937" fill="#1DA1F2" />
        </button>
      </div>
    </div>
  </div>
</div>
```

```

        </svg>
    </button>
    <button className="flex items-center justify-center py-
2.5 border rounded-lg hover:bg-gray-50 duration-150 active:bg-gray-100">
        <svg className="w-5 h-5" viewBox="0 0 48 48"
fill="none" xmlns="http://www.w3.org/2000/svg">
            <g clip-path="url(#clip0_910_21)">
                <path fill-rule="evenodd" clip-rule="evenodd"
d="M24.0005 1C18.303 1.00296 12.7923 3.02092 8.45374 6.69305C4.11521 10.3652
1.23181 15.452 0.319089 21.044C-0.593628 26.636 0.523853 32.3684 3.47174
37.2164C6.41963 42.0643 11.0057 45.7115 16.4099 47.5059C17.6021 47.7272 18.0512
46.9883 18.0512 46.36C18.0512 45.7317 18.0273 43.91 18.0194 41.9184C11.3428
43.3608 9.93197 39.101 9.93197 39.101C8.84305 36.3349 7.26927 35.6078 7.26927
35.6078C5.09143 34.1299 7.43223 34.1576 7.43223 34.1576C9.84455 34.3275 11.1123
36.6194 11.1123 36.6194C13.2504 40.2667 16.7278 39.2116 18.0949 38.5952C18.3095
37.0501 18.9335 35.999 19.621 35.4023C14.2877 34.8017 8.68408 32.7548 8.68408
23.6108C8.65102 21.2394 9.53605 18.9461 11.156 17.2054C10.9096 16.6047 10.087
14.1785 11.3905 10.8829C11.3905 10.8829 13.4054 10.2427 17.9916 13.3289C21.9253
12.2592 26.0757 12.2592 30.0095 13.3289C34.5917 10.2427 36.6026 10.8829 36.6026
10.8829C37.9101 14.1706 37.0875 16.5968 36.8411 17.2054C38.4662 18.9464 39.353
21.2437 39.317 23.6187C39.317 32.7824 33.7015 34.8017 28.3602 35.3905C29.2186
36.1334 29.9856 37.5836 29.9856 39.8122C29.9856 43.0051 29.9578 45.5736 29.9578
46.36C29.9578 46.9962 30.391 47.7391 31.6071 47.5059C37.0119 45.7113 41.5984
42.0634 44.5462 37.2147C47.4941 32.3659 48.611 26.6326 47.6972 21.0401C46.7835
15.4476 43.8986 10.3607 39.5587 6.68921C35.2187 3.01771 29.7067 1.00108 24.0085
1H24.0005Z" fill="#191717" />
                <path d="M9.08887 35.264C9.03721 35.3826
8.84645 35.4181 8.69146 35.3351C8.53646 35.2522 8.42122 35.098 8.47686
34.9755C8.5325 34.853 8.71928 34.8214 8.87428 34.9044C9.02927 34.9874 9.14848
35.1455 9.08887 35.264Z" fill="#191717" />
                <path d="M10.0626 36.3428C9.98028 36.384
9.88612 36.3955 9.79622 36.3753C9.70632 36.3551 9.62629 36.3045 9.56979
36.2321C9.41479 36.0662 9.38298 35.837 9.50221 35.7342C9.62143 35.6315 9.83606
35.6789 9.99105 35.8449C10.146 36.0108 10.1818 36.24 10.0626 36.3428Z"
fill="#191717" />
                <path d="M11.0085 37.714C10.8614 37.8167
10.6111 37.714 10.472 37.5085C10.4335 37.4716 10.4029 37.4274 10.382
37.3785C10.3611 37.3297 10.3503 37.2771 10.3503 37.224C10.3503 37.1709 10.3611
37.1183 10.382 37.0694C10.4029 37.0205 10.4335 36.9763 10.472 36.9395C10.619
36.8407 10.8694 36.9395 11.0085 37.141C11.1476 37.3425 11.1516 37.6112 11.0085
37.714Z" fill="#191717" />
                <path d="M12.2921 39.0417C12.161 39.1879
11.8947 39.1484 11.6761 38.9508C11.4575 38.7532 11.4059 38.4845 11.537
38.3423C11.6682 38.2 11.9344 38.2395 12.161 38.4331C12.3875 38.6268 12.4312
38.8994 12.2921 39.0417Z" fill="#191717" />
                <path d="M14.0923 39.8162C14.0327 40.0019
13.7625 40.0849 13.4922 40.0059C13.222 39.9268 13.0432 39.7055 13.0948
39.5159C13.1465 39.3262 13.4207 39.2393 13.6949 39.3262C13.9691 39.4131 14.144
39.6226 14.0923 39.8162Z" fill="#191717" />
                <path d="M16.0557 39.9505C16.0557 40.1442
15.8331 40.3101 15.547 40.3141C15.2608 40.318 15.0264 40.16 15.0264
39.9663C15.0264 39.7727 15.2489 39.6067 15.535 39.6028C15.8212 39.5988 16.0557
39.753 16.0557 39.9505Z" fill="#191717" />
                <path d="M17.8838 39.6463C17.9196 39.8399
17.7208 40.0414 17.4347 40.0888C17.1486 40.1363 16.8982 40.0217 16.8624
39.832C16.8267 39.6423 17.0333 39.4368 17.3115 39.3855C17.5897 39.3341 17.848
39.4526 17.8838 39.6463Z" fill="#191717" />
            </g>
        </defs>
        <clipPath id="clip0_910_21">
            <rect width="48" height="48" fill="white"
/>
        </clipPath>
    </defs>
</svg>
</button>
</div>
<div className="relative">
    <span className="block w-full h-px bg-gray-300"></span>

```

```

        <p className="inline-block w-fit text-sm bg-white px-2
absolute -top-2 inset-x-0 mx-auto">Or continue with</p>
      </div>
      <form
        onSubmit={(e) => e.preventDefault()}
        className="space-y-5"
        >
        <div>
          <label className="font-medium">
            Name
          </label>
          <input
            type="text"
            required
            className="w-full mt-2 px-3 py-2 text-gray-500
bg-transparent outline-none border focus:border-indigo-600 shadow-sm rounded-lg"
            />
          </div>
          <div>
            <label className="font-medium">
              Email
            </label>
            <input
              type="email"
              required
              className="w-full mt-2 px-3 py-2 text-gray-500
bg-transparent outline-none border focus:border-indigo-600 shadow-sm rounded-lg"
              />
            </div>
            <div>
              <label className="font-medium">
                Password
              </label>
              <input
                type="password"
                required
                className="w-full mt-2 px-3 py-2 text-gray-500
bg-transparent outline-none border focus:border-indigo-600 shadow-sm rounded-lg"
                />
              </div>
              <button
                className="w-full px-4 py-2 text-white font-medium
bg-indigo-600 hover:bg-indigo-500 active:bg-indigo-600 rounded-lg duration-150"
                >
                Create account
              </button>
            </form>
          </div>
        </div>
      </main></>
    )
  }

  export default Signup;

```

```

import Footer from "../components/Footer"
import Navbar from "../components/Navbar"
import { Link } from "react-router-dom";
import "../App2.css";

const people = [

```

```

        {
          name: 'Cappuccino',
          role: '$100',
          imageUrl:
            'https://images.pexels.com/photos/302899/pexels-photo-302899.jpeg?auto=compress&cs=tinysrgb&w=600',
        },
        {
          name: 'Latte',
          role: '$90',
          imageUrl:
            'https://images.pexels.com/photos/3879802/pexels-photo-3879802.jpeg?auto=compress&cs=tinysrgb&w=600',
        },
        {
          name: 'Macho',
          role: '$80',
          imageUrl:
            'https://images.pexels.com/photos/20854092/pexels-photo-20854092/free-photo-of-close-up-of-a-glass-of-coffee-with-milk-foam-and-cinnamon-on-top.jpeg?auto=compress&cs=tinysrgb&w=600',
        },
      ],
    ],
  ],
]

```

```

export default function Team() {
  return (
    <><Navbar/>
    <div className="bg-white py-24 sm:py-32"> <div id="back"><Link to="/create"
      className="btn btn-primary">
        Go to Order
      </Link></div><br/>
    <div className="mx-auto grid max-w-7xl gap-x-8 gap-y-20 px-6 lg:px-8
      xl:grid-cols-3">

      <div className="max-w-2xl">
        <h2 className="text-3xl font-bold tracking-tight text-gray-900
          sm:text-4xl">Meet our Wonderful Menu</h2>
        <p className="mt-6 text-lg leading-8 text-gray-600">
          Libero fames augue nisl porttitor nisi, quis. Id ac elit odio vitae
          elementum enim vitae ullamcorper
          suspendisse.
        </p>
      </div>

      <ul role="list" className="grid gap-x-8 gap-y-12 sm:grid-cols-2 sm:gap-
        y-16 xl:col-span-2">
        {people.map((person) => (
          <li key={person.name}>
            <div className="flex items-center gap-x-6">
              <img className="h-16 w-16 rounded-full" src={person.imageUrl}
                alt="" />
              <div>
                <h3 className="text-base font-semibold leading-7 tracking-
                  tight text-gray-900">{person.name}</h3>
                <p className="text-sm font-semibold leading-6 text-indigo-
                  600">{person.role}</p>
              </div>
            </div>
          </li>
        ))}
      </ul>
    </div><Footer/></>
  )
}

```

```
import axios from "axios";
import React, { useEffect, useState } from "react";
import { useNavigate, Link } from "react-router-dom";
import "../App2.css";
import Navbar from "../components/Navbar";

import styled from 'styled-components';

const Container = styled.div`
  background-color: #f9f9f9;
  padding: 20px;
`;

const Title = styled.h1`
  text-align: center;
  background-color: #f9f9f9;
  padding: 10px;
  margin-bottom: 20px;
  color: #333;
`;

const StyledTable = styled.table`
  width: 100%;
  border-collapse: collapse;
  background-color: #fff;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
`;

const TableHead = styled.th`
  background-color: #f9f9f9;
  color: #333;
  padding: 10px;
  text-align: left;
`;

const TableRow = styled.tr`
  &:nth-child(even) {
    background-color: #f5f5f5;
  }
`;

const TableCell = styled.td`
  padding: 10px;
`;

const ActionButton = styled.button`
  background-color: #007bff;
  color: #fff;
  padding: 5px 10px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  margin-right: 5px;

  &:hover {
    background-color: #0056b3;
  }
`;
```

```

const SearchInput = styled.input`
  padding: 5px;
  font-size: 16px;
  border-radius: 4px;
  border: 1px solid #ccc;
  margin-right: 5px;
`;

function ViewOrder() {
  const [id, setId] = useState('');
  const [columns, setColumns] = useState([]);
  const [records, setRecords] = useState([]);
  const navigate = useNavigate();

  useEffect(() => {
    axios
      .get("http://localhost:4444/order/all")
      .then((response) => {
        setColumns(Object.keys(response.data[0]));
        setRecords(response.data);
      })
      .catch((err) => {
        console.log(err);
      });
  }, []);

  let handleSubmit = (id) => {
    const conf = window.confirm("Do you want to delete");
    if (conf) {
      axios
        .delete("http://localhost:4444/order/" + id)
        .then((res) => {
          alert("record has deleted");
          navigate("/");
          window.location.reload();
        })
        .catch((err) => console.log(err));
    }
  };

  return (
    <>
      <Navbar/>
      <Container>
        <Title>Order Details</Title>
        <div className="text-end">
          <Link to="/create" className="btn btn-primary">
            Add +
          </Link><br/><br/>
          <Link to="/items" className="btn btn-primary">
            Menu
          </Link><br/>
        </div>
        <br />
        <div className="text-end">
          <SearchInput type="number" id="id" placeholder="Enter Order Id"
            name="id" onChange={(e) => setId(e.target.value)}/>
          {/* <button onClick={handleSearch}>
            <ActionButton style={{ backgroundColor:
              'green' }}>Find</ActionButton>
          </button> */}
        </div>
        <br />
        <StyledTable>
          <thead>
            <tr>

```



```

        {columns.map((c, i) => (
        <TableHead key={i}>{c}</TableHead>
        )))
        <TableHead>Action</TableHead>
        </tr>
        </thead>
        <tbody>
        {records.filter((item)=>{ return id === ''?item :
item.orderId==parseInt(id)}).map((d, i) => (
        <TableRow key={i}>
        <TableCell>{d.orderId}</TableCell>
        <TableCell>{d.orderedItems}</TableCell>
        <TableCell>{d.totalAmount}</TableCell>

        <TableCell>{d.user.id}</TableCell>
        <TableCell>{d.payment}</TableCell>

        <TableCell>
        <Link to={` /update/${d.orderId}`}>
        <ActionButton style={{ backgroundColor:
'#228B22' }}>Update</ActionButton>
        </Link>
        <button   onClick={(e) => handleSubmit(d.orderId)}>
        <ActionButton style={{ backgroundColor:
'red' }}>Delete</ActionButton>
        </button>
        </TableCell>
        </TableRow>
        )))
        </tbody>
        </StyledTable>
        </Container>
        </>
        );
    }

    export default ViewOrder;

```

```

import axios from "axios";
import React, { useEffect, useState } from "react";
import { useNavigate, Link } from "react-router-dom";
import "../App2.css";
import Navbar from "../components/Navbar";

import styled from 'styled-components';

const Container = styled.div`
  background-color: #f9f9f9;
  padding: 20px;
  `;

const Title = styled.h1`
  text-align: center;
  background-color: #f9f9f9;
  padding: 10px;
  margin-bottom: 20px;
  color: #333;
  `;

const StyledTable = styled.table`
  width: 100%;

```

```

        border-collapse: collapse;
        background-color: #fff;
        box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
        border-radius: 8px;
        `;

const TableHead = styled.th`
    background-color: #f9f9f9;
    color: #333;
    padding: 10px;
    text-align: left;
    `;

const TableRow = styled.tr`
    &:nth-child(even) {
        background-color: #f5f5f5;
    }
    `;

const TableCell = styled.td`
    padding: 10px;
    `;

const ActionButton = styled.button`
    background-color: #007bff;
    color: #fff;
    padding: 5px 10px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    margin-right: 5px;

    &:hover {
        background-color: #0056b3;
    }
    `;

const SearchInput = styled.input`
    padding: 5px;
    font-size: 16px;
    border-radius: 4px;
    border: 1px solid #ccc;
    margin-right: 5px;
    `;

function ViewUser() {
    const [id, setId] = useState('');
    const [columns, setColumns] = useState([]);
    const [records, setRecords] = useState([]);
    const navigate = useNavigate();

    useEffect(() => {
        axios
        .get("http://localhost:4444/user/all")
        .then((response) => {
            setColumns(Object.keys(response.data[0]));
            setRecords(response.data);
        })
        .catch((err) => {
            console.log(err);
        });
    }, []);

    // const handleSearch = () => {
    //     axios
    //     .get("http://localhost:4444/payroll/"+id)
    //     .then((response) => {

```

```

//      setColumns(Object.keys(response.data));
//      setRecords([response.data]);
//    })
//    .catch((err) => console.log(err));
//  }

let handleSubmit = (id) => {
const conf = window.confirm("Do you want to delete");
  if (conf) {
    axios
      .delete("http://localhost:4444/user/" + id)
      .then((res) => {
        alert("record has deleted");
        navigate("/");
        window.location.reload();
      })
      .catch((err) => console.log(err));
  }
};

return (
  <>
    <Navbar/>
    <Container>
      <Title>Find User</Title>
      <div className="text-end">
        <Link to="/createuser" className="btn btn-primary">
          Add +
        </Link>
      </div>
      <br />
      <div className="text-end">
        <SearchInput type="number" id="id" placeholder="Enter User Id"
          name="id" onChange={(e) => setId(e.target.value)}>
          {/* <button onClick={handleSearch}>
            <ActionButton style={{ backgroundColor:
              'green' }}>Find</ActionButton>
          </button> */}
        </div>
        <br />
        <StyledTable>
          <thead>
            <tr>
              {columns.map((c, i) => (
                <TableHead key={i}>{c}</TableHead>
              ))}
              <TableHead>Action</TableHead>
            </tr>
          </thead>
          <tbody>
            {records.filter((item) => { return id === '' ? item :
              item.id === parseInt(id) }).map((d, i) => (
              <TableRow key={i}>
                <TableCell>{d.id}</TableCell>
                <TableCell>{d.name}</TableCell>
                <TableCell>{d.emailid}</TableCell>
                <TableCell>{d.address}</TableCell>
                <TableCell>{d.phoneNumber}</TableCell>

                <TableCell>
                  <Link to={`/updateuser/${d.id}`}>
                    <ActionButton style={{ backgroundColor:
                      '#228B22' }}>Update</ActionButton>
                  </Link>
                  <button   onClick={(e) => handleSubmit(d.id)}>
                    <ActionButton style={{ backgroundColor:
                      'red' }}>Delete</ActionButton>
                  </button>
                </TableCell>
              </TableRow>
            )
          )}
          </tbody>
        </StyledTable>
      </div>
    </div>
  </>
);

```

```
        </TableCell>
      </TableRow>
    )})
  </tbody>
</StyledTable>
</Container>
</>
);
}

export default ViewUser;
```

Components:

```
const links = [
  { name: 'Open roles', href: '#' },
  { name: 'Internship program', href: '#' },
  { name: 'Our values', href: '#' },
  { name: 'Meet our leadership', href: '#' },
]

const stats = [
  { name: 'Offices worldwide', value: '12' },
  { name: 'Full-time colleagues', value: '300+' },
  { name: 'Hours per week', value: '40' },
  { name: 'Paid time off', value: 'Unlimited' },
]

export default function About() {
  return (
    <div className="relative isolate overflow-hidden bg-gray-900 py-24 sm:py-32">
      
      <div
        className="hidden sm:absolute sm:-top-10 sm:right-1/2 sm:-z-10 sm:mr-10 sm:block sm:transform-gpu sm:blur-3xl"
        aria-hidden="true"
      >
        <div
          className="aspect-[1097/845] w-[68.5625rem] bg-gradient-to-tr from-[#ff4694] to-[#776fff] opacity-20"
          style={{
            clipPath:
              'polygon(74.1% 44.1%, 100% 61.6%, 97.5% 26.9%, 85.5% 0.1%, 80.7% 2%, 72.5% 32.5%, 60.2% 62.4%, 52.4% 68.1%, 47.5% 58.3%, 45.2% 34.5%, 27.5% 76.7%, 0.1% 64.9%, 17.9% 100%, 27.6% 76.8%, 76.1% 97.7%, 74.1% 44.1%)',
          }}
        />
      </div>
      <div
        className="absolute -top-52 left-1/2 -z-10 -translate-x-1/2 transform-gpu blur-3xl sm:top-[-28rem] sm:ml-16 sm:translate-x-0 sm:transform-gpu"
        aria-hidden="true"
      >
        <div
          className="aspect-[1097/845] w-[68.5625rem] bg-gradient-to-tr from-[#ff4694] to-[#776fff] opacity-20"
```

```

        style={{
          clipPath:
            'polygon(74.1% 44.1%, 100% 61.6%, 97.5% 26.9%, 85.5% 0.1%, 80.7%
2%, 72.5% 32.5%, 60.2% 62.4%, 52.4% 68.1%, 47.5% 58.3%, 45.2% 34.5%, 27.5% 76.7%,
0.1% 64.9%, 17.9% 100%, 27.6% 76.8%, 76.1% 97.7%, 74.1% 44.1%)',
        }}
      />
    </div>
    <div className="mx-auto max-w-7xl px-6 lg:px-8">
      <div className="mx-auto max-w-2xl lg:mx-0">
        <h2 className="text-4xl font-bold tracking-tight text-white sm:text-
6xl">Drink with us</h2>
        <p className="mt-6 text-lg leading-8 text-gray-300">
          Anim aute id magna aliqua ad ad non deserunt sunt. Qui irure qui
          lorem cupidatat commodo. Elit sunt amet
          fugiat veniam occaecat fugiat aliqua.
        </p>
      </div>
      <div className="mx-auto mt-10 max-w-2xl lg:mx-0 lg:max-w-none">
        <div className="grid grid-cols-1 gap-x-8 gap-y-6 text-base font-
semibold leading-7 text-white sm:grid-cols-2 md:flex lg:gap-x-10">
          {links.map((link) => (
            <a key={link.name} href={link.href}>
              {link.name} <span aria-hidden="true">&rarr;</span>
            </a>
          ))}
        </div>
        <dl className="mt-16 grid grid-cols-1 gap-8 sm:mt-20 sm:grid-cols-2
lg:grid-cols-4">
          {stats.map((stat) => (
            <div key={stat.name} className="flex flex-col-reverse">
              <dt className="text-base leading-7 text-gray-
300">{stat.name}</dt>
              <dd className="text-2xl font-bold leading-9 tracking-tight
text-white">{stat.value}</dd>
            </div>
          ))}
        </dl>
      </div>
    </div>
  )
}

```

```

const posts = [
  {
    id: 1,
    title: 'Boost your conversion rate with our founder',
    href: '#',
    description:
      'Illo sint voluptas. Error voluptates culpa eligendi. Hic vel totam vitae
illo. Non aliquid explicabo necessitatibus unde. Sed exercitationem placeat
consectetur nulla deserunt vel. Iusto corrupti dicta.',
    date: 'Mar 16, 2020',
    datetime: '2020-03-16',
    category: { title: 'Growing with Coffee', href: '#' },
    author: {
      name: 'NandhaKumaran',
      role: 'Founder',
      href: '#',
      imageUrl:
        'https://images.pexels.com/photos/1438081/pexels-photo-
1438081.jpeg?auto=compress&cs=tinysrgb&w=600',
    },
  },

```

```

    },
    },

]

export default function Blog() {
  return (
    <div className="bg-white py-24 sm:py-32">
      <div className="mx-auto max-w-7xl px-6 lg:px-8">
        <div className="mx-auto max-w-2xl lg:mx-0">
          <h2 className="text-3xl font-bold tracking-tight text-gray-900 sm:text-4xl">From the blog</h2>
          <p className="mt-2 text-lg leading-8 text-gray-600">
            Learn how to make coffee with smile.
          </p>
        </div>
        <div className="mx-auto mt-10 grid max-w-2xl grid-cols-1 gap-x-8 gap-y-16 border-t border-gray-200 pt-10 sm:mt-16 sm:pt-16 lg:mx-0 lg:max-w-none lg:grid-cols-3">
          {posts.map((post) => (
            <article key={post.id} className="flex max-w-xl flex-col items-start justify-between">
              <div className="flex items-center gap-x-4 text-xs">
                <time dateTime={post.datetime} className="text-gray-500">
                  {post.date}
                </time>
                <a
                  href={post.category.href}
                  className="relative z-10 rounded-full bg-gray-50 px-3 py-1.5 font-medium text-gray-600 hover:bg-gray-100"
                >
                  {post.category.title}
                </a>
              </div>
              <div className="group relative">
                <h3 className="mt-3 text-lg font-semibold leading-6 text-gray-900 group-hover:text-gray-600">
                  <a href={post.href}>
                    <span className="absolute inset-0" />
                    {post.title}
                  </a>
                </h3>
                <p className="mt-5 line-clamp-3 text-sm leading-6 text-gray-600">{post.description}</p>
              </div>
              <div className="relative mt-8 flex items-center gap-x-4">
                <img src={post.author.imageUrl} alt="" className="h-10 w-10 rounded-full bg-gray-50" />
                <div className="text-sm leading-6">
                  <p className="font-semibold text-gray-900">
                    <a href={post.author.href}>
                      <span className="absolute inset-0" />
                      {post.author.name}
                    </a>
                  </p>
                  <p className="text-gray-600">{post.author.role}</p>
                </div>
              </div>
            </article>
          )))
        </div>
      </div>
    </div>
  )
}

```

```

import React from "react";
import {
  FaFacebook,
  FaGithub,
  FaInstagram,
  FaTwitter,
  FaTwitch,
} from "react-icons/fa";

const sections = [
  {
    title: "Coffee",
    items: ["Making", "Smiling", "Billing", "Price", "Delivery"]
  },
];

const items = [
  { name: "Facebook", icon: FaFacebook, link: "https://www.facebook.com/" },
  { name: "Instagram", icon: FaInstagram, link: "https://www.instagram.com/" },
  { name: "Twitter", icon: FaTwitter, link: "https://twitter.com/" },
  { name: "Twitch", icon: FaTwitch, link: "https://www.twitch.tv/" },
  { name: "Github", icon: FaGithub, link: "https://github.com/" },
];

const Footer = () => {
  return (
    <div className="w-full mt-24 bg-slate-900 text-gray-300 py-y px-2">
      <div className="max-w-[1240px] mx-auto grid grid-cols-2 md:grid-cols-6
        border-b-2 border-gray-600 py-8">
        {sections.map((section, index) => (
          <div key={index}>
            <h6 className="font-bold uppercase pt-2">{section.title}</h6>
            <ul>
              {section.items.map((item, i) => (
                <li key={i} className="py-1 text-gray-500 hover:text-white">
                  {item}
                </li>
              ))}
            </ul>
          </div>
        ))}

        <div className="col-span-2 pt-8 md:pt-2">
          <p className="font-bold uppercase">Subscribe to our newsletter</p>
          <p className="py-4">
            The latest news, articles, and resources, sent to your inbox weekly.
          </p>
          <form className="flex flex-col sm:flex-row">
            <input
              className="w-full p-2 mr-4 rounded-md mb-4"
              type="email"
              placeholder="Enter email.."
            />
            <button className="p-2 mb-4">Subscribe</button>
          </form>
        </div>
      </div>

      <div className="flex flex-col max-w-[1240px] px-2 py-4 mx-auto justify-
        between sm:flex-row text-center text-gray-500">
        <p className="py-4">2022 Workflow, LLC. All rights reserved</p>
        <div className="flex justify-between sm:w-[300px] pt-4 text-2xl">
          {items.map((x, index) => {

```

```

        return <x.icon key={index} className="hover:text-white" />;
      }
    }
  </div>
</div>
</div>
);
};

export default Footer;

```

```

import { useState } from 'react'
import { Dialog, DialogPanel } from '@headlessui/react'
import { Bars3Icon, XMarkIcon } from '@heroicons/react/24/outline'

import Blog from './Blog';
import About from './About';
import Footer from './Footer';

const navigation = [
  { name: 'Place Your Order', href: '/create' },
  { name: 'User Adder', href: '/createuser' },

  { name: 'OrderDetails', href: '/view' },
  { name: 'UserDetails', href: '/viewuser' },

]

export default function Hero() {
  const [mobileMenuOpen, setMobileMenuOpen] = useState(false)

  return (
    <div className="bg-white">
      <header className="absolute inset-x-0 top-0 z-50">
        <nav className="flex items-center justify-between p-6 lg:px-8" aria-label="Global">
          <div className="flex lg:flex-1">
            <a href="/" className="-m-1.5 p-1.5">
              <span className="sr-only">Your Company</span>
              
            </a>
          </div>
          <div className="flex lg:hidden">
            <button
              type="button"
              className="-m-2.5 inline-flex items-center justify-center rounded-md p-2.5 text-gray-700"
              onClick={() => setMobileMenuOpen(true)}
            >
              <span className="sr-only">Open main menu</span>
              <Bars3Icon className="h-6 w-6" aria-hidden="true" />
            </button>
          </div>
        </nav>
      </header>
    </div>
  )
}

```



```

        <div className="hidden lg:flex lg:gap-x-12">
            {navigation.map((item) => (
                <a key={item.name} href={item.href} className="text-sm font-
                    semibold leading-6 text-gray-900">
                        {item.name}
                    </a>
                )))}
        </div>

        <div className="hidden lg:flex lg:flex-1 lg:justify-end">
            <a href="/signup" className="text-sm font-semibold leading-6 text-
                gray-900">
                Sign Up <span aria-hidden="true">&rarr;</span>
            </a>
        </div>
    </nav>

    <Dialog className="lg:hidden" open={mobileMenuOpen}
        onClose={setMobileMenuOpen}>
        <div className="fixed inset-0 z-50" />
        <DialogPanel className="fixed inset-y-0 right-0 z-50 w-full overflow-y-
            auto bg-white px-6 py-6 sm:max-w-sm sm:ring-1 sm:ring-gray-900/10">
            <div className="flex items-center justify-between">
                <a href="#" className="-m-1.5 p-1.5">
                    <span className="sr-only">Your Company</span>
                    
                </a>
                <button
                    type="button"
                    className="-m-2.5 rounded-md p-2.5 text-gray-700"
                    onClick={() => setMobileMenuOpen(false)}
                >
                    <span className="sr-only">Close menu</span>
                    <XMarkIcon className="h-6 w-6" aria-hidden="true" />
                </button>
            </div>
            <div className="mt-6 flow-root">
                <div className="-my-6 divide-y divide-gray-500/10">
                    <div className="space-y-2 py-6">
                        {navigation.map((item) => (
                            <a
                                key={item.name}
                                href={item.href}
                                className="-mx-3 block rounded-lg px-3 py-2 text-base font-
                                    semibold leading-7 text-gray-900 hover:bg-gray-50"
                            >
                                {item.name}
                            </a>
                        )))}
                    </div>
                    <div className="py-6">
                        <a
                            href="/signup"
                            className="-mx-3 block rounded-lg px-3 py-2.5 text-base font-
                                semibold leading-7 text-gray-900 hover:bg-gray-50"
                        >
                            Sign Up
                        </a>
                    </div>
                </div>
            </div>
        </DialogPanel>
    </Dialog>
</header>

```

```

        <div className="relative isolate px-6 pt-14 lg:px-8">
            <div
                className="absolute inset-x-0 -top-40 -z-10 transform-gpu overflow-
                    hidden blur-3xl sm:-top-80"
                aria-hidden="true"
            >
                <div
                    className="relative left-[calc(50%-11rem)] aspect-[1155/678] w-
[36.125rem] -translate-x-1/2 rotate-[30deg] bg-gradient-to-tr from-[#ff80b5] to-
[#9089fc] opacity-30 sm:left-[calc(50%-30rem)] sm:w-[72.1875rem]"
                    style={{
                        clipPath:
                            'polygon(74.1% 44.1%, 100% 61.6%, 97.5% 26.9%, 85.5% 0.1%, 80.7%
2%, 72.5% 32.5%, 60.2% 62.4%, 52.4% 68.1%, 47.5% 58.3%, 45.2% 34.5%, 27.5% 76.7%,
0.1% 64.9%, 17.9% 100%, 27.6% 76.8%, 76.1% 97.7%, 74.1% 44.1%)',
                    }}
                />
            </div>
            <div className="mx-auto max-w-2xl py-32 sm:py-48 lg:py-56">
                <div className="hidden sm:mb-8 sm:flex sm:justify-center">
                    <div className="relative rounded-full px-3 py-1 text-sm leading-6
text-gray-600 ring-1 ring-gray-900/10 hover:ring-gray-900/20">
                        What are the Operations are available{' '}
                        <a href="#" className="font-semibold text-indigo-600">
                            <span className="absolute inset-0" aria-hidden="true" />
                            Read more <span aria-hidden="true">&rarr;</span>
                        </a>
                    </div>
                </div>
                <div className="text-center">
                    <h1 className="text-4xl font-bold tracking-tight text-gray-900
sm:text-6xl">
                        Enjoy Your Coffee
                    </h1>
                    <p className="mt-6 text-lg leading-8 text-gray-600">
                        Anim aute id magna aliqua ad ad non deserunt sunt. Qui irure qui
                        lorem cupidatat commodo. Elit sunt amet
                        fugiat veniam occaecat fugiat aliqua.
                    </p>
                    <div className="mt-10 flex items-center justify-center gap-x-6">
                        <a
                            href="/show"
                            className="rounded-md bg-indigo-600 px-3.5 py-2.5 text-sm font-
                                semibold text-white shadow-sm hover:bg-indigo-500 focus-visible:outline focus-
                                visible:outline-2 focus-visible:outline-offset-2 focus-visible:outline-indigo-
                                    600"
                        >
                            Get started
                        </a>
                        <a href="#" className="text-sm font-semibold leading-6 text-gray-
900">
                            Learn more <span aria-hidden="true">></span>
                        </a>
                    </div>
                </div>
            </div>
            <div
                className="absolute inset-x-0 top-[calc(100%-13rem)] -z-10 transform-
                    gpu overflow-hidden blur-3xl sm:top-[calc(100%-30rem)]"
                aria-hidden="true"
            >
                <div
                    className="relative left-[calc(50%+3rem)] aspect-[1155/678] w-
[36.125rem] -translate-x-1/2 bg-gradient-to-tr from-[#ff80b5] to-[#9089fc]
opacity-30 sm:left-[calc(50%+36rem)] sm:w-[72.1875rem]"
                    style={{
                        clipPath:

```

```

        'polygon(74.1% 44.1%, 100% 61.6%, 97.5% 26.9%, 85.5% 0.1%, 80.7%
2%, 72.5% 32.5%, 60.2% 62.4%, 52.4% 68.1%, 47.5% 58.3%, 45.2% 34.5%, 27.5% 76.7%,
0.1% 64.9%, 17.9% 100%, 27.6% 76.8%, 76.1% 97.7%, 74.1% 44.1%)',
    })
  />
</div>
</div>

<About/>
<Blog/>
<Footer/>

</div>
)
}

```

```

import {
  Disclosure,
  DisclosureButton,
  DisclosurePanel,
  Menu,
  MenuButton,
  MenuItem,
  MenuItems,
  Transition,
} from '@headlessui/react'
import { Bars3Icon, BellIcon, XMarkIcon } from '@heroicons/react/24/outline'

const navigation = [
  { name: 'Home', href: '/', current: false },
  { name: 'Place Your Order', href: '/create', current: false },
  { name: 'UserAdder', href: '/createuser', current: false },

  // { name: 'EmployeeViewer', href: '/show', current: false },
  // { name: 'PayrollViewer', href: '/showpay', current: false },

  { name: 'OrderDetails', href: '/view', current: false },
  { name: 'UserDetails', href: '/viewuser', current: false },

]

function classNames(...classes) {
  return classes.filter(Boolean).join(' ')
}

export default function Navbar() {
  return (
    <Disclosure as="nav" className="bg-gray-800">
      {( { open } ) => (
        <>
          <div className="mx-auto max-w-7xl px-2 sm:px-6 lg:px-8">
            <div className="relative flex h-16 items-center justify-between">
              <div className="absolute inset-y-0 left-0 flex items-center sm:hidden">
                {/* Mobile menu button*/}

```

```

<DisclosureButton className="relative inline-flex items-center
justify-center rounded-md p-2 text-gray-400 hover:bg-gray-700 hover:text-white
focus:outline-none focus:ring-2 focus:ring-inset focus:ring-white">
  <span className="absolute -inset-0.5" />
  <span className="sr-only">Open main menu</span>
  {open ? (
    <XMarkIcon className="block h-6 w-6" aria-hidden="true" />
  ) : (
    <Bars3Icon className="block h-6 w-6" aria-hidden="true" />
  )}
</DisclosureButton>
</div>
<div className="flex flex-1 items-center justify-center sm:items-
stretch sm:justify-start">
  <div className="flex flex-shrink-0 items-center">
    <a href="/">
      </a>
    </div>
    <div className="hidden sm:ml-6 sm:block">
      <div className="flex space-x-4">
        {navigation.map((item) => (
          <a
            key={item.name}
            href={item.href}
            className={classNames(
              item.current ? 'bg-gray-900 text-white' : 'text-gray-
300 hover:bg-gray-700 hover:text-white',
              'rounded-md px-3 py-2 text-sm font-medium',
            )}
            aria-current={item.current ? 'page' : undefined}
          >
            {item.name}
          </a>
        ))}
      </div>
    </div>
  </div>
  <div className="absolute inset-y-0 right-0 flex items-center pr-2
sm:static sm:inset-auto sm:ml-6 sm:pr-0">
    <button
      type="button"
      className="relative rounded-full bg-gray-800 p-1 text-gray-
400 hover:text-white focus:outline-none focus:ring-2 focus:ring-white focus:ring-
offset-2 focus:ring-offset-gray-800"
    >
      <span className="absolute -inset-1.5" />
      <span className="sr-only">View notifications</span>
      <BellIcon className="h-6 w-6" aria-hidden="true" />
    </button>

    {/* Profile dropdown */}
    <Menu as="div" className="relative ml-3">
      <div>
        <MenuButton className="relative flex rounded-full bg-gray-
800 text-sm focus:outline-none focus:ring-2 focus:ring-white focus:ring-offset-2
focus:ring-offset-gray-800">
          <span className="absolute -inset-1.5" />
          <span className="sr-only">Open user menu</span>
          

```

```

        </MenuButton>
      </div>
      <Transition
        enter="transition ease-out duration-100"
        enterFrom="transform opacity-0 scale-95"
        enterTo="transform opacity-100 scale-100"
        leave="transition ease-in duration-75"
        leaveFrom="transform opacity-100 scale-100"
        leaveTo="transform opacity-0 scale-95"
      >
        <MenuItems className="absolute right-0 z-10 mt-2 w-48
origin-top-right rounded-md bg-white py-1 shadow-lg ring-1 ring-black ring-
opacity-5 focus:outline-none">
          <MenuItem>
            {{{ focus }}} => (
              <a
                href="#"
                className={classNames(focus ? 'bg-gray-100' : '',
'block px-4 py-2 text-sm text-gray-700')}
              >
                Your Profile
              </a>
            )}
          </MenuItem>
          <MenuItem>
            {{{ focus }}} => (
              <a
                href="#"
                className={classNames(focus ? 'bg-gray-100' : '',
'block px-4 py-2 text-sm text-gray-700')}
              >
                Settings
              </a>
            )}
          </MenuItem>
          <MenuItem>
            {{{ focus }}} => (
              <a
                href="#"
                className={classNames(focus ? 'bg-gray-100' : '',
'block px-4 py-2 text-sm text-gray-700')}
              >
                Sign out
              </a>
            )}
          </MenuItem>
        </MenuItems>
      </Transition>
    </Menu>
  </div>
</div>

<DisclosurePanel className="sm:hidden">
<div className="space-y-1 px-2 pb-3 pt-2">
  {navigation.map((item) => (
    <DisclosureButton
      key={item.name}
      as="a"
      href={item.href}
      className={classNames(
        item.current ? 'bg-gray-900 text-white' : 'text-gray-300
hover:bg-gray-700 hover:text-white',
        'block rounded-md px-3 py-2 text-base font-medium',
      )}
      aria-current={item.current ? 'page' : undefined}
    >
      {item.name}
    </DisclosureButton>
  ))}

```

```

    )))
  </div>
</DisclosurePanel>
</>
  })
</Disclosure>
)
}

```

Approuter:

```

import React from 'react';
import { BrowserRouter, Routes, Route } from "react-router-dom";

import Add from './pages/Add';
import AddPayroll from './pages/AddUser';

import Edit from './pages/Edit';
import Hero from './components/Hero';
import EditUser from './pages/EditUser';
import ViewOrder from './pages/ViewOrder';
import ViewUser from './pages/ViewUser';

import Signin from './pages/Signin';
import Signup from './pages/Signup';
import AddUser from './pages/AddUser';
import Team from './pages/Team';

function AppRouter() {

  return (
    <BrowserRouter>

      <Routes>
        <Route path="/" element={<Hero/>}></Route>

        <Route path="/create" element={<Add/>}></Route>
        <Route path="/createuser" element={<AddUser/>}></Route>
        <Route path="/update/:id" element={<Edit/>}></Route>
        <Route path="/updateuser/:id" element={<EditUser/>}></Route>

        <Route path="/view" element={<ViewOrder/>}></Route>
        <Route path="/viewuser" element={<ViewUser/>}></Route>

        <Route path="/signin" element={<Signin/>}></Route>

        <Route path="/signup" element={<Signup/>}></Route>
        <Route path="/items" element={<Team/>}></Route>

      </Routes>
    </BrowserRouter>
  )
}

export default AppRouter;

```

Backend Code:

Controllers:

```
package com.assessment.App.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import com.assessment.App.model.Items;
import com.assessment.App.model.Order;
import com.assessment.App.service.ItemsService;
import com.assessment.App.service.OrderService;

import java.util.List;

@CrossOrigin("*")
@RestController
@RequestMapping("/items")
public class ItemsController {

    @Autowired
    private ItemsService itemsService;

    @GetMapping("/all")
    public List<Items> getAllItemss() {
        return itemsService.getAllItemss();
    }

    @GetMapping("/{id}")
    public Items getItemsById(@PathVariable int id) {
        return itemsService.getItemsById(id);
    }

    @PostMapping
    public void addItem(@RequestBody Items items) {
        itemsService.saveItems(items);
    }

    @PutMapping("/{id}")
```

```

        public void updateItems(@RequestBody Items items) {
            itemsService.updateItems(items);
        }

        @DeleteMapping("/{id}")
        public void deleteItems(@PathVariable("id") int id) {
            itemsService.deleteItems(id);
        }
    }

    package com.assessment.App.controller;

    import org.springframework.beans.factory.annotation.Autowired;
    import org.springframework.web.bind.annotation.*;

    import com.assessment.App.model.Order;
    import com.assessment.App.service.OrderService;

    import java.util.List;

    @CrossOrigin("*")
    @RestController
    @RequestMapping("/order")
    public class OrderController {

        @Autowired
        private OrderService orderService;

        @GetMapping("/all")
        public List<Order> getAllOrders() {
            return orderService.getAllOrders();
        }

        @GetMapping("/{id}")
        public Order getOrderById(@PathVariable int id) {
            return orderService.getOrderById(id);
        }

        @PostMapping
        public void addOrder(@RequestBody Order order) {
            orderService.saveOrder(order);
        }

        @PutMapping("/{id}")
        public void updateOrder(@RequestBody Order order) {
            orderService.updateOrder(order);
        }

        @DeleteMapping("/{id}")
        public void deleteOrder(@PathVariable("id") int id) {
            orderService.deleteOrder(id);
        }
    }

```



```

    }

    package com.assessment.App.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import com.assessment.App.model.User;
import com.assessment.App.service.UserService;

import java.util.List;

    @CrossOrigin("*")
    @RestController
    @RequestMapping("/user")
    public class UserController {

        private final UserService userService;

        @Autowired
        public UserController(UserService userService) {
            this.userService = userService;
        }

        @GetMapping("/all")
        public List<User> getAllUsers() {
            return userService.getAllUsers();
        }

        @GetMapping("/{id}")
        public User getUserById(@PathVariable int id) {
            return userService.getUserById(id);
        }

        @PostMapping
        public void createUser(@RequestBody User user) {
            userService.saveUser(user);
        }

        @PutMapping("/{id}")
        public void updateUser(@RequestBody User user) {
            userService.updateUser(user);
        }

        @DeleteMapping("/{id}")
        public void deleteUser(@PathVariable("id") int id) {
            userService.deleteUser(id);
        }
    }

```

Models:

```
package com.assessment.App.model;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Entity
@Table(name = "itemtable")
public class Items {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "items_id")
    private int itemsId;
    private int Cappuccino;
    private int Latte;
    private int Mocha;

    public Items() {
        super();
        // TODO Auto-generated constructor stub
    }

    public Items(int itemsId, int cappuccino, int latte, int mocha)
    {
        super();
        this.itemsId = itemsId;
        Cappuccino = cappuccino;
        Latte = latte;
        Mocha = mocha;
    }

    public int getItemsId() {
        return itemsId;
    }

    public void setItemsId(int itemsId) {
        this.itemsId = itemsId;
    }

    public int getCappuccino() {
        return Cappuccino;
    }

    public void setCappuccino(int cappuccino) {
        Cappuccino = cappuccino;
    }

    public int getLatte() {
        return Latte;
    }

    public void setLatte(int latte) {
        Latte = latte;
    }

    public int getMocha() {
```

```

        return Mocha;
    }

    public void setMocha(int mocha) {
        Mocha = mocha;
    }

    @Override
    public String toString() {
return "Items [itemsId=" + itemsId + ", Cappuccino=" +
        Cappuccino + ", Latte=" + Latte + ", Mocha=" + Mocha
        + " ]";
    }

}

```

```

package com.assessment.App.model;

```

```

import java.util.List;

```

```

import jakarta.persistence.Column;

```

```

import jakarta.persistence.ElementCollection;

```

```

import jakarta.persistence.Entity;

```

```

import jakarta.persistence.GeneratedValue;

```

```

import jakarta.persistence.GenerationType;

```

```

import jakarta.persistence.Id;

```

```

import jakarta.persistence.JoinColumn;

```

```

import jakarta.persistence.ManyToOne;

```

```

import jakarta.persistence.OneToOne;

```

```

import jakarta.persistence.Table;

```

```

@Entity

```

```

@Table(name = "ordertable")

```

```

public class Order {

```

```

    @Id

```

```

    @GeneratedValue(strategy = GenerationType.IDENTITY)

```

```

    @Column(name = "order_id")

```

```

    private int orderId;

```

```

    private String orderedItems;

```

```

    private Double totalAmount;

```

```
private String Payment;
```

```
@ManyToOne
```

```
@JoinColumn(name = "user_id")
```

```
private User user;
```

```
public Order() {
```

```
    super();
```

```
// TODO Auto-generated constructor stub
```

```
}
```

```
public Order(int orderId, String orderedItems, Double totalAmount, String payment, User  
    user) {
```

```
    super();
```

```
    this.orderId = orderId;
```

```
    this.orderedItems = orderedItems;
```

```
    this.totalAmount = totalAmount;
```

```
    Payment = payment;
```

```
    this.user = user;
```

```
}
```

```
public int getOrderId() {
```

```
    return orderId;
```

```
}
```

```
public void setOrderId(int orderId) {
```

```
    this.orderId = orderId;
```

```
}
```

```
public String getOrderedItems() {
```

```
    return orderedItems;
```

```
}
```

```
public void setOrderedItems(String orderedItems) {
```

```
        this.orderedItems = orderedItems;
    }
}
```

```
    public Double getTotalAmount() {
        return totalAmount;
    }
}
```

```
public void setTotalAmount(Double totalAmount) {
    this.totalAmount = totalAmount;
}
}
```

```
    public String getPayment() {
        return Payment;
    }
}
```

```
public void setPayment(String payment) {
    Payment = payment;
}
}
```

```
    public User getUser() {
        return user;
    }
}
```

```
    public void setUser(User user) {
        this.user = user;
    }
}
```

```
    @Override
    public String toString() {
        return "Order [orderId=" + orderId + ", orderedItems=" + orderedItems + ", totalAmount=" +
            totalAmount
            + ", Payment=" + Payment + ", user=" + user + "];"
    }
}
```

```
}
```

```
package com.assessment.App.model;

import java.util.List;

import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.OneToMany;
import jakarta.persistence.OneToOne;
import jakarta.persistence.Table;

@Entity
@Table(name = "Usertable")
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "user_id")
    private int id;

    private String name;
    private String emailid;
    private String Address;
    private int PhoneNumber;
    public User() {
        super();
    }
    // TODO Auto-generated constructor stub

    public User(int id, String name, String emailid, String address,
        int phoneNumber) {
        super();
        this.id = id;
        this.name = name;
        this.emailid = emailid;
        Address = address;
        PhoneNumber = phoneNumber;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
```

```

        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmailid() {
        return emailid;
    }

    public void setEmailid(String emailid) {
        this.emailid = emailid;
    }

    public String getAddress() {
        return Address;
    }

    public void setAddress(String address) {
        Address = address;
    }

    public int getPhoneNumber() {
        return PhoneNumber;
    }

    public void setPhoneNumber(int phoneNumber) {
        PhoneNumber = phoneNumber;
    }

    @Override
    public String toString() {
        return "User [id=" + id + ", name=" + name + ", emailid=" +
            emailid + ", Address=" + Address + ", PhoneNumber="
                + PhoneNumber + "];"
    }

}

```

Repository:

```
package com.assessment.App.repository;
```

```

import java.util.List;

import org.springframework.stereotype.Repository;

import com.assessment.App.model.Items;
import com.assessment.App.model.Order;

import jakarta.transaction.Transactional;

@Repository
public interface ItemsRepository {

    Items findById(int id);

    void save(Items items);

    void delete(int id);

    void update(Items items);

    List<Items> findAllItems();

}

package com.assessment.App.repository;

import java.util.List;

import org.springframework.stereotype.Repository;

import com.assessment.App.model.Order;

import jakarta.transaction.Transactional;

@Repository
public interface OrderRepository {

    Order findById(int id);
    void save(Order order);
    void delete(int id);
    void update(Order order);
    List<Order> findAllOrders();

}

```



```

package com.assessment.App.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.assessment.App.model.User;

import jakarta.transaction.Transactional;

@Repository
public interface UserRepository{

    User findUById(int id);
    void saveU(User user);
    void deleteU(int id);
    void updateU(User user);
    List<User> findAllUsers();

}

```

#### Repository Implementation:

```

package com.assessment.App.repositoryImplementation;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

import com.assessment.App.model.Items;
import com.assessment.App.model.Order;
import com.assessment.App.repository.ItemsRepository;
import com.assessment.App.repository.OrderRepository;

import jakarta.persistence.EntityManager;
import jakarta.persistence.Query;
import jakarta.transaction.Transactional;

@Repository
@Transactional

public class ItemsRepositoryImplementation implements ItemsRepository {

    @Autowired

```

```

        EntityManager eManager;

        @Override
        public void save(Item item) {
            // TODO Auto-generated method stub
            eManager.persist(item);
        }

        @Override
        public List<Item> findAllItems() {
            String hql = "from Item";
            Query query = eManager.createQuery(hql);
            return query.getResultList();
        }

        @Override
        public Item findById(int id) {
            // TODO Auto-generated method stub
            return eManager.find(Item.class, id);
        }

        @Override
        public void update(Item item) {
            // TODO Auto-generated method stub
            eManager.merge(item);
        }

        @Override
        public void delete(int id) {
            Item itemToDelete = findById(id);
            if (itemToDelete != null) {
                eManager.remove(itemToDelete);
            }
        }

    }

```

```

package com.assessment.App.repositoryImplementation;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

```

```

import com.assessment.App.model.Order;
import com.assessment.App.repository.OrderRepository;

import jakarta.persistence.EntityManager;
import jakarta.persistence.Query;
import jakarta.transaction.Transactional;

@Repository
@Transactional
public class OrderRepositoryImplementation implements
    OrderRepository {

    @Autowired
    EntityManager eManager;

    @Override
    public void saveO(Order order) {
        // TODO Auto-generated method stub
        eManager.persist(order);
    }

    @Override
    public List<Order> findAllOrders() {
        String hql = "from Order";
        Query query = eManager.createQuery(hql);
        return query.getResultList();
    }

    @Override
    public Order findById(int id) {
        // TODO Auto-generated method stub
        return eManager.find(Order.class, id);
    }

    @Override
    public void updateO(Order order) {
        // TODO Auto-generated method stub
        eManager.merge(order);
    }

    @Override
    public void deleteO(int id) {
        Order orderToDelete = findById(id);
        if (orderToDelete != null) {
            eManager.remove(orderToDelete);
        }
    }

}

package com.assessment.App.repositoryImplementation;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

import com.assessment.App.model.User;

```

```

import com.assessment.App.repository.UserRepository;

import jakarta.persistence.EntityManager;
import jakarta.persistence.Query;
import jakarta.transaction.Transactional;

@Repository
@Transactional
public class UserRepositoryImplementation implements
    UserRepository {

    @Autowired
    EntityManager eManager;

    public void saveU(User user) {
        // TODO Auto-generated method stub
        eManager.persist(user);
    }

    public List<User> findAllUsers() {
        String hql = "from User";
        Query query = eManager.createQuery(hql);
        return query.getResultList();
    }

    @Override
    public User findUById(int id) {
        // TODO Auto-generated method stub
        return eManager.find(User.class, id);
    }

    @Override
    public void updateU(User user) {
        // TODO Auto-generated method stub
        eManager.merge(user);
    }

    @Override
    public void deleteU(int id) {
        User userToDelete = findUById(id);
        if (userToDelete != null) {
            eManager.remove(userToDelete);
        }
    }

}

```

Services:

```

package com.assessment.App.service;

import java.util.List;

import com.assessment.App.model.Items;

public interface ItemsService {

```

```
        Items getItemById(int id);  
void saveItems(Items items);  
        void deleteItems(int id);  
void updateItems(Items items);  
        List<Items> getAllItemss();  
  
    }
```

```
package com.assessment.App.service;
```

```
import java.util.List;
```

```
import com.assessment.App.model.Order;
```

```
public interface OrderService {
```

```
    Order getOrderById(int id);  
    void saveOrder(Order order);  
    void deleteOrder(int id);  
    void updateOrder(Order order);  
    List<Order> getAllOrders();  
  
}
```

```
package com.assessment.App.service;
```

```
import java.util.List;
```

```
import com.assessment.App.model.User;
```

```
public interface UserService {
```

```
    User getUserById(int id);  
    void saveUser(User user);  
    void deleteUser(int id);  
    void updateUser( User user);
```

```
List<User> getAllUsers();  
}
```

### Services Implementation:

```
package com.assessment.App.serviceImplementation;  
  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import com.assessment.App.model.Items;  
import com.assessment.App.model.Order;  
import com.assessment.App.repository.ItemsRepository;  
import com.assessment.App.repository.OrderRepository;  
import com.assessment.App.service.ItemsService;  
import com.assessment.App.service.OrderService;  
  
@Service  
public class ItemsServiceImpl implements ItemsService {  
  
    private final ItemsRepository itemsRepo;  
  
    @Autowired  
    public ItemsServiceImpl(ItemsRepository itemsRepo) {  
        this.itemsRepo = itemsRepo;  
    }  
  
    @Override  
    public void saveItems(Items items) {  
        itemsRepo.save(items);  
    }  
  
    @Override  
    public Items getItemsById(int id) {
```

```
        return itemsRepo.findIdById(id);  
    }  
}
```

```
public List<Items> getAllItemss() {  
  
    return itemsRepo.findAllItemss();  
}
```

```
    @Override  
    public void updateItems(Items items) {  
        // TODO Auto-generated method stub  
  
        itemsRepo.updateId(items);  
    }  
}
```

```
    @Override  
    public void deleteItems(int id) {  
        // TODO Auto-generated method stub  
  
        itemsRepo.deleteId(id);  
    }  
}
```

```
}
```

```
package com.assessment.App.serviceImplementation;  
  
    import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
    import org.springframework.stereotype.Service;  
  
    import com.assessment.App.model.Order;  
import com.assessment.App.repository.OrderRepository;  
    import com.assessment.App.service.OrderService;  
  
    @Service  
public class OrderServiceImplementation implements OrderService  
    {
```

```

        private final OrderRepository orderRepo;

        @Autowired
        public OrderServiceImpl(OrderRepository orderRepo) {
            this.orderRepo = orderRepo;
        }

        @Override
        public void saveOrder(Order order) {
            orderRepo.saveO(order);
        }

        @Override
        public Order getOrderById(int id) {

            return orderRepo.findOById(id);
        }

        public List<Order> getAllOrders() {

            return orderRepo.findAllOrders();
        }

        @Override
        public void updateOrder(Order order) {
            // TODO Auto-generated method stub

            orderRepo.updateO(order);

        }

        @Override
        public void deleteOrder(int id) {
            // TODO Auto-generated method stub
            orderRepo.deleteO(id);

        }

    }

}

package com.assessment.App.serviceImplementation;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.assessment.App.model.Order;
import com.assessment.App.repository.OrderRepository;
import com.assessment.App.service.OrderService;

```



```

        @Service
public class OrderServiceImpl implements OrderService
    {

        private final OrderRepository orderRepo;

        @Autowired
public OrderServiceImpl(OrderRepository orderRepo) {
            this.orderRepo = orderRepo;
        }

        @Override
public void saveOrder(Order order) {
            orderRepo.saveO(order);
        }

        @Override
public Order getOrderById(int id) {

            return orderRepo.findOById(id);
        }

public List<Order> getAllOrders() {

            return orderRepo.findAllOrders();
        }

        @Override
public void updateOrder(Order order) {
            // TODO Auto-generated method stub

            orderRepo.updateO(order);

        }

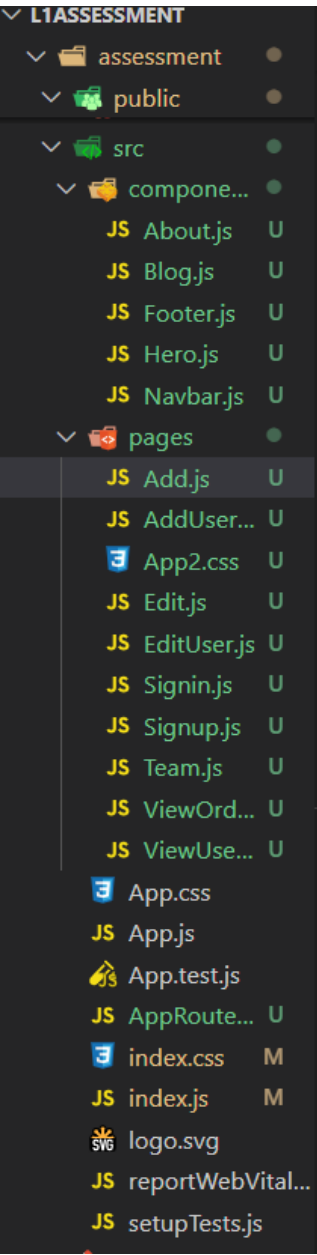
        @Override
public void deleteOrder(int id) {
            // TODO Auto-generated method stub
            orderRepo.deleteO(id);

        }

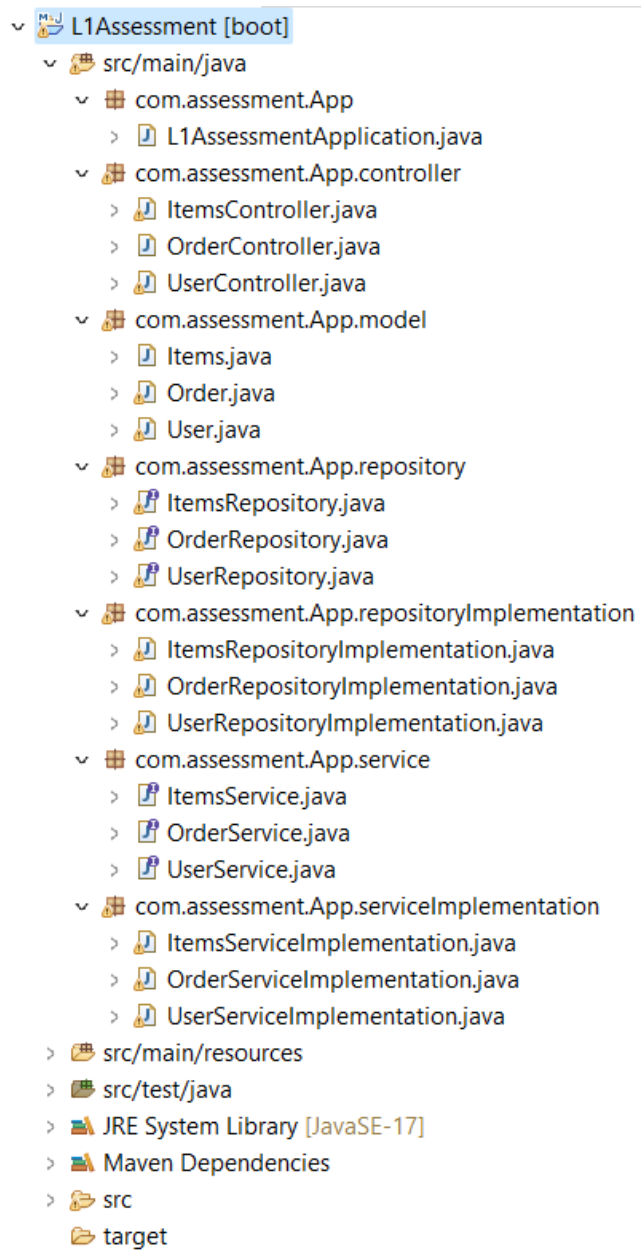
    }
}

```

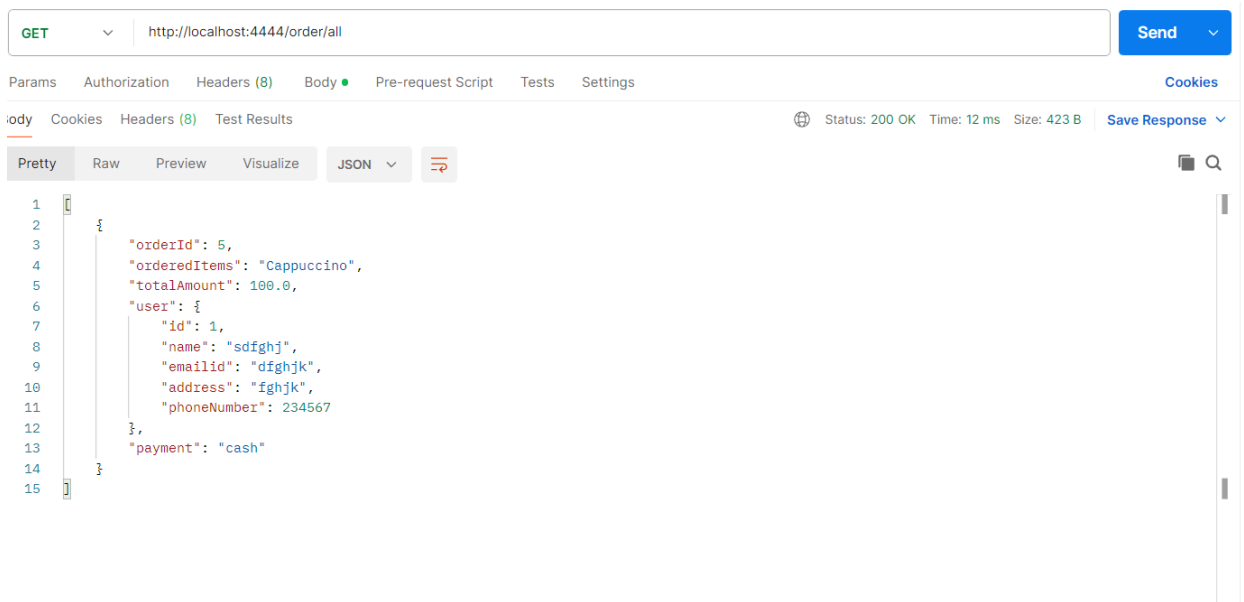
## Frontend Folder Structure



Backend Structure



## Screenshots:





<

PUT http://

GET http://

GET http://

[CONFLICT]

GET http://

PUT http://

GET http://

POST http://

GET http://

GET http://

POST http://


DEL http://

DEL http://

>

+

...

 http://localhost:4444/order/5

DELETE

http://localhost:4444/order/5

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

JSON

Beautify

```
1  {}
2  ----"orderedItems": "Cappuccino,-Latte",
3  ----"totalAmount": 190.0,
4  ----"user": {}
5  ----"id": 1,
6  ----"name": "nandha",
7  ----"emailid": "Itghjk@fghj",
8  ----"address": "ityuiop",
9  ----"phoneNumber": 12345678
10 ----,
11 ----"payment": "cash"
12 ----}
```

<

PUT http://

GET http://

GET http://

[CONFLICT]

GET http://

PUT http://

GET http://

POST http://

GET http://

GET http://

POST http://


DEL http://

GET http://

>

+

...

 http://localhost:4444/order/all

GET

http://localhost:4444/order/all

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 9 ms

Size: 430 B

Save Response

Pretty


Raw

Preview

Visualize

JSON

```
1  {}
2  {
3    "orderId": 6,
4    "orderedItems": "Cappuccino, Latte",
5    "totalAmount": 190.0,
6    "user": {
7      "id": 1,
8      "name": "sdfghj",
9      "emailid": "dfghjk",
10     "address": "fghjk",
11     "phoneNumber": 234567
12   },
13   "payment": "cash"
14 }
15 }
```



Place Your Order

User Adder

OrderDetails

UserDetails

Sign Up

What are the Operations are available

Read more

# Enjoy Your Coffee

Anim aute id magna aliqua ad ad non deserunt sunt. Qui irure qui lorem cupidatat commodo. Elit sunt amet fugiat veniam occaecat fugiat aliqua.

Get started

Learn more

# Drink with us

Anim aute id magna aliqua ad ad non deserunt sunt. Qui irure qui lorem cupidatat commodo. Elit sunt amet fugiat veniam occaecat fugiat aliqua.

Open roles

Internship program

Our values

Meet our leadership

12

Offices worldwide

300+

Full-time colleagues

40

Hours per week

Unlimited

Paid time off


# From the blog

Learn how to make coffee with smile.

Mar 16, 2020 Growing with Coffee

## Boost your conversion rate with our founder

Ilo sint voluptas. Error voluptates culpa eligendi. Hic vel totam vitae illo. Non aliquid explicabo necessitatibus unde. Sed exercitationem placeat...



**NandhaKumaran**  
Founder

React App

localhost:3000/viewuser


HomePlace Your OrderUserAdderOrderDetailsUserDetails

Find User

Add +

Enter User Id

id	name	emailid	address	phoneNumber	Action
1	sdfghj	dfghjk	fghjk	234567	<div>UpdateDelete</div>
7	rtgyhujiko	ftghjk@fghj	rtyuio	12345678	<div>UpdateDelete</div>
8	tyuhjik	ghjk@dfgh	poiujhgf	9876543	<div>UpdateDelete</div>



**NandhaKumaran**  
Founder

- COFFEE
- Making

Smiling

Billing

Price

Delivery

### SUBSCRIBE TO OUR NEWSLETTER

The latest news, articles, and resources, sent to your inbox weekly.

Enter email..

Subscribe

