**MAOTH -> public**

public->index.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <title>MaTh App</title>

  </head>

  <body>

    <div id="root"></div>

  </body>

</html>

**MAOTH -> src**

**src -> components**

**components -> UI**

**UI -> Card.js**

import React from 'react';

export const Card = ({ title, value }) => {

  return (

    <div className="bg-white p-4 rounded-lg shadow-md flex flex-col items-center">

      <h3 className="text-lg font-medium text-gray-600">{title}</h3>

      <p className="text-2xl font-bold mt-2">{value}</p>

    </div>

  );

};

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**components -> Dashboard.js**

// src/components/Dashboard.js

import React, { useContext } from 'react';

import { UserContext } from '../context/UserContext';

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

import {

  BarChart,

  Bar,

  XAxis,

  YAxis,

  Tooltip,

  CartesianGrid,

  ResponsiveContainer,

} from 'recharts';

// Example “dummy” weekly progress data

const sampleData = [

  { week: 'Week 1', value: 40 },

  { week: 'Week 2', value: 55 },

  { week: 'Week 3', value: 60 },

  { week: 'Week 4', value: 75 },

];

const Dashboard = () => {

  const { userData } = useContext(UserContext);

  // (Optional) If you wanted to fetch real “card” values from a dummy API:

  // const [stats, setStats] = useState({ teamCount: 0, activeProjects: 0, notifications: 0 });

  // useEffect(() => {

  //   axios.get('https://jsonplaceholder.typicode.com/todos?userId=1') // for example

  //     .then(resp => {

  //       setStats({

  //         teamCount: 12,         // ...or derive from resp.data

  //         activeProjects: 5,

  //         notifications: 3,

  //       });

  //     });

  // }, []);

  return (

    <div className="p-6">

      <h1 className="text-2xl font-bold mb-4">

        Welcome, {userData.name}

      </h1>

      <div className="mb-6 text-gray-700">

        <p>Email: {userData.email}</p>

        <p>Company: {userData.company}</p>

      </div>

      <div className="grid grid-cols-1 sm:grid-cols-3 gap-6 mb-6">

        <Card title="Team Members Count" value="12" />

        <Card title="Active Projects" value="5" />

        <Card title="Notifications" value="3" />

      </div>

      <div className="bg-white p-4 rounded-lg shadow-lg">

        <h2 className="text-xl font-semibold mb-4">Weekly Progress</h2>

        <ResponsiveContainer width="100%" height={300}>

          <BarChart data={sampleData}>

            <CartesianGrid strokeDasharray="3 3" />

            <XAxis dataKey="week" />

            <YAxis />

            <Tooltip />

            <Bar dataKey="value" />

          </BarChart>

        </ResponsiveContainer>

      </div>

    </div>

  );

};

export default Dashboard;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**components -> OnboardingWizard.js**

// src/components/OnboardingWizard.js

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

import axios from 'axios';                        // ← import Axios

import Step1 from './Step1';

import Step2 from './Step2';

import Step3 from './Step3';

import ProgressBar from './ProgressBar';

import { UserContext } from '../context/UserContext';

const OnboardingWizard = () => {

  const [step, setStep] = useState(1);

  const { setUserData } = useContext(UserContext);

  const [formData, setFormData] = useState({

    name: '',

    email: '',

    company: '',

    industry: '',

    size: '',

    theme: 'light',

    layout: 'grid',

  });

  const [isSubmitting, setIsSubmitting] = useState(false); // ← to disable button while posting

  const [error, setError] = useState(null);                // ← to show any POST error

  const validateStep = () => {

    if (step === 1) {

      return formData.name && formData.email;

    }

    if (step === 2) {

      return formData.company && formData.industry && formData.size;

    }

    return true;

  };

  const handleNext = () => {

    if (validateStep() && step < 3) {

      setStep(prev => prev + 1);

    }

  };

  const handleBack = () => {

    if (step > 1) {

      setStep(prev => prev - 1);

    }

  };

  // ↓ UPDATED handleSubmit to POST to dummy API first

  const handleSubmit = async () => {

    setIsSubmitting(true);

    setError(null);

    try {

      // JSONPlaceholder ignores most fields and returns an "id" in the response

      const response = await axios.post(

        'https://jsonplaceholder.typicode.com/users',

        formData

      );

      console.log('Dummy API response:', response.data);

      // Only after the POST succeeds do we save to Context/localStorage

      setUserData(formData);

    } catch (err) {

      console.error('Error posting to dummy API:', err);

      setError('Failed to save data. Please try again.');

      setIsSubmitting(false);

      return;

    }

    // No need to call setIsSubmitting(false) here, because once userData is set,

    // App.js’s redirect sends us to /dashboard immediately.

  };

  return (

    <div className="max-w-xl mx-auto p-6 bg-white mt-10 rounded-lg shadow-lg">

      <ProgressBar step={step} />

      <div className="mt-6">

        {step === 1 && <Step1 formData={formData} setFormData={setFormData} />}

        {step === 2 && <Step2 formData={formData} setFormData={setFormData} />}

        {step === 3 && <Step3 formData={formData} setFormData={setFormData} />}

      </div>

      {error && (

        <p className="text-red-500 mt-4">

          {error}

        </p>

      )}

      <div className="flex justify-between mt-6">

        <button

          onClick={handleBack}

          disabled={step === 1 || isSubmitting}

          className="px-4 py-2 bg-gray-300 text-gray-700 rounded disabled:opacity-50"

        >

          Back

        </button>

        {step < 3 && (

          <button

            onClick={handleNext}

            disabled={isSubmitting}

            className="px-4 py-2 bg-blue-500 text-white rounded disabled:opacity-50"

          >

            Next

          </button>

        )}

        {step === 3 && (

          <button

            onClick={handleSubmit}

            disabled={isSubmitting}

            className="px-4 py-2 bg-green-500 text-white rounded disabled:opacity-50"

          >

            {isSubmitting ? 'Submitting…' : 'Submit'}

          </button>

        )}

      </div>

    </div>

  );

};

export default OnboardingWizard;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**components -> ProgressBar.js**

import React from 'react';

const ProgressBar = ({ step }) => {

  // We have 3 steps total, so percentage = (current step / 3) \* 100

  const progress = (step / 3) \* 100;

  return (

    <div className="w-full bg-gray-200 h-2 rounded-full">

      <div

        className="h-2 bg-blue-500 rounded-full"

        style={{ width: `${progress}%` }}

      ></div>

    </div>

  );

};

export default ProgressBar;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**components -> Step3.js**

import React from 'react';

const Step3 = ({ formData, setFormData }) => {

  return (

    <div>

      <h2 className="text-xl font-semibold mb-4">Preferences</h2>

      <div className="mb-4">

        <label className="block text-gray-700">Theme</label>

        <select

          value={formData.theme}

          onChange={e => setFormData({ ...formData, theme: e.target.value })}

          className="w-full mt-1 p-2 border rounded"

        >

          <option value="light">Light</option>

          <option value="dark">Dark</option>

        </select>

      </div>

      <div>

        <label className="block text-gray-700">Default Dashboard Layout</label>

        <select

          value={formData.layout}

          onChange={e => setFormData({ ...formData, layout: e.target.value })}

          className="w-full mt-1 p-2 border rounded"

        >

          <option value="grid">Grid</option>

          <option value="list">List</option>

        </select>

      </div>

    </div>

  );

};

export default Step3;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**components -> Step2.js**

import React from 'react';

const Step2 = ({ formData, setFormData }) => {

  return (

    <div>

      <h2 className="text-xl font-semibold mb-4">Business Info</h2>

      <div className="mb-4">

        <label className="block text-gray-700">Company Name</label>

        <input

          type="text"

          value={formData.company}

          onChange={e => setFormData({ ...formData, company: e.target.value })}

          className="w-full mt-1 p-2 border rounded"

          required

        />

      </div>

      <div className="mb-4">

        <label className="block text-gray-700">Industry</label>

        <input

          type="text"

          value={formData.industry}

          onChange={e => setFormData({ ...formData, industry: e.target.value })}

          className="w-full mt-1 p-2 border rounded"

          required

        />

      </div>

      <div>

        <label className="block text-gray-700">Company Size</label>

        <select

          value={formData.size}

          onChange={e => setFormData({ ...formData, size: e.target.value })}

          className="w-full mt-1 p-2 border rounded"

          required

        >

          <option value="">Select Size</option>

          <option value="1-10">1-10</option>

          <option value="11-50">11-50</option>

          <option value="51-200">51-200</option>

          <option value="201-500">201-500</option>

          <option value="500+">500+</option>

        </select>

      </div>

    </div>

  );

};

export default Step2;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**components -> Step1.js**

import React from 'react';

const Step1 = ({ formData, setFormData }) => {

  return (

    <div>

      <h2 className="text-xl font-semibold mb-4">Personal Info</h2>

      <div className="mb-4">

        <label className="block text-gray-700">Name</label>

        <input

          type="text"

          value={formData.name}

          onChange={e => setFormData({ ...formData, name: e.target.value })}

          className="w-full mt-1 p-2 border rounded"

          required

        />

      </div>

      <div>

        <label className="block text-gray-700">Email</label>

        <input

          type="email"

          value={formData.email}

          onChange={e => setFormData({ ...formData, email: e.target.value })}

          className="w-full mt-1 p-2 border rounded"

          required

        />

      </div>

    </div>

  );

};

export default Step1;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**src -> context**

**context -> UserContext.js**

// src/context/UserContext.js

import React, { createContext, useState, useEffect } from 'react';

export const UserContext = createContext();

export const UserProvider = ({ children }) => {

  // Try to load from localStorage if it exists

  const [userData, setUserData] = useState(() => {

    const saved = localStorage.getItem('userData');

    return saved ? JSON.parse(saved) : null;

  });

  // Whenever userData changes, write it to localStorage

  useEffect(() => {

    if (userData) {

      localStorage.setItem('userData', JSON.stringify(userData));

    }

  }, [userData]);

  return (

    <UserContext.Provider value={{ userData, setUserData }}>

      {children}

    </UserContext.Provider>

  );

};

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**src -> App.js**

// src/App.js

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

import { Routes, Route, useNavigate } from 'react-router-dom';

import OnboardingWizard from './components/OnboardingWizard';

import Dashboard from './components/Dashboard';

import { UserContext } from './context/UserContext';

function App() {

  const { userData } = useContext(UserContext);

  const navigate = useNavigate();

  useEffect(() => {

    if (userData) {

      navigate('/dashboard');

    }

  }, [userData, navigate]);

  return (

    <div className="min-h-screen bg-gray-100">

      <Routes>

        <Route path="/" element={<OnboardingWizard />} />

        <Route path="/dashboard" element={<Dashboard />} />

      </Routes>

    </div>

  );

}

export default App;

// src/App.js

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

import { Routes, Route, useNavigate } from 'react-router-dom';

import OnboardingWizard from './components/OnboardingWizard';

import Dashboard from './components/Dashboard';

import { UserContext } from './context/UserContext';

function App() {

  const { userData } = useContext(UserContext);

  const navigate = useNavigate();

  useEffect(() => {

    if (userData) {

      navigate('/dashboard');

    }

  }, [userData, navigate]);

  return (

    <div className="min-h-screen bg-gray-100">

      <Routes>

        <Route path="/" element={<OnboardingWizard />} />

        <Route path="/dashboard" element={<Dashboard />} />

      </Routes>

    </div>

  );

}

export default App;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**src -> App.css**

.App {

  text-align: center;

}

.App-logo {

  height: 40vmin;

  pointer-events: none;

}

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

  .App-logo {

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

  }

}

.App-header {

  background-color: #282c34;

  min-height: 100vh;

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

  font-size: calc(10px + 2vmin);

  color: white;

}

.App-link {

  color: #61dafb;

}

@keyframes App-logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**src -> index.js**

// src/index.js

import React from 'react';

import { createRoot } from 'react-dom/client'; // ← note: react-dom/client

import './index.css';

import App from './App';

import { BrowserRouter } from 'react-router-dom';

import { UserProvider } from './context/UserContext';

// Find the <div id="root"> in public/index.html

const container = document.getElementById('root');

// Create a root and render your App into it

const root = createRoot(container);

root.render(

  <React.StrictMode>

    <BrowserRouter>

      <UserProvider>

        <App />

      </UserProvider>

    </BrowserRouter>

  </React.StrictMode>

);

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**src -> index.css**

@tailwind base;

@tailwind components;

@tailwind utilities;