COLLEGE CODE: 9528

COLLEGE NAME: SCAD College of Engineering and Technology

DEPARTMENT: CSE

STUDENT NM ID: d99d3571e0c45cea55047bbb81583279

ROLL NO: 95282331104125

DATE: 26/09/2005

Completed the project named as phase ___ TECHNOLOGY PROJECT

NAME: SINGLE PAGE APPLICATION

SUBMITED BY,

NAME: D. PRIYA

MOBILE NO: 78458 95165

1. Project Setup

Before starting the core development, set up the project environment properly:

Initialize Project:

```
npx create-react-app spa-project
cd spa-project
```

Install Required Packages:

```
npm install react-router-dom axios redux react-redux tailwindcss
```

Set up Folder Structure:

```
src/
├── components/
├── pages/
├── services/
├── store/
├── assets/
└── App.js
```

Configure Tailwind CSS (or chosen styling library).

2. Core Features Implementation

These are the main features required for MVP:

Routing

Use React Router to enable client-side navigation:

```
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import HomePage from "./pages/HomePage";
import AboutPage from "./pages/AboutPage";

function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/about" element={<AboutPage />} />
      </Routes>
    </Router>
  );
}
export default App;
```

Components

Create reusable components such as:

- Header
- Footer
- Navbar
- ProductCard
- DataList

API Integration

Example for fetching product data:

```
import axios from "axios";

export const fetchProducts = async () => {
  try {
    const res = await axios.get("/api/products");
    return res.data;
  } catch (error) {
    console.error("Error fetching products", error);
  }
};
```

3. Data Storage (Local State / Database)

Local State (Frontend)

Using React state or Redux:

```
import { useState } from "react";

const [products, setProducts] = useState([]);
```

Database (Backend)

Using MongoDB with Express.js:
API endpoints to Create, Read, Update, Delete (CRUD) data.
Example:

```
app.get("/api/products", async (req, res) => {
  const products = await Product.find();
  res.json(products);
});
```

4. Testing Core Features

Before finalizing MVP, test:

- Navigation works without page reload.
- API calls return expected results.
- State updates correctly.
- Responsive design on mobile and desktop.
- Error handling for failed requests.

Testing Tools:

- React Developer Tools
- Postman (for API testing)
- Browser Console Logs
- Manual User Testing

5. Version Control (GitHub)

Track progress using Git and GitHub:

Initialize Git Repository:

```
git init
```

Commit Changes:

```
git add .
git commit -m "Initial commit"
```

Push to GitHub:

```
git remote add origin <your-repository-url>
git push -u origin master
```

Tip: Maintain meaningful commits and branches for different features.