

## OBJECTIVES & CONCEPTS

### 1. Define SPA and Its Benefits

#### **SPA (Single Page Application):**

A web application that loads a single HTML page and dynamically updates the content without refreshing the whole page.

#### **Benefits of SPA:**

- Faster user experience (less server round-trips)
- Smooth navigation
- Backend only sends data, not entire pages
- Better caching and offline support

### 2. Define React and Identify Its Working

#### **React:**

A JavaScript library created by Facebook for building **user interfaces**—especially SPAs.

#### **How it works:**

- Uses **components** (reusable UI blocks)
- Uses a **virtual DOM** to detect changes and update the real DOM efficiently

### 3. Difference Between SPA and MPA

Aspect	SPA	MPA (Multi Page Application)
Page reload	No (updates dynamically)	Yes (each interaction reloads page)
Speed	Faster, after initial load	Slower due to full reloads
Technologies	React, Angular, Vue	JSP, PHP, traditional server-rendered
URL management	Done using client-side routing	Server-based routing
SEO	More difficult	Easier (static content)

### 4. Pros & Cons of Single-Page Application

#### **Pros:**

- Fast and responsive
- Smooth user experience
- Less server load

#### **Cons:**

- Poor SEO (unless using SSR)
- Initial load may be slower
- Can be complex to manage state & routing

## 5. Explain About React

- Component-based UI library
- Developed and maintained by **Meta (Facebook)**
- Follows **declarative programming**
- Uses **JSX** (JavaScript + XML)

## 6. Define Virtual DOM

The **Virtual DOM** is a lightweight in-memory copy of the actual DOM.

React compares (diffs) the new virtual DOM with the previous one, and **efficiently updates** only the parts of the actual DOM that changed.

## 7. Features of React

- Declarative UI
- Component-based architecture
- JSX Syntax
- Virtual DOM
- Unidirectional data flow
- React Hooks
- Strong community and ecosystem

## 1. Hands-On Lab: React Setup

### Prerequisites:

- Install [Node.js and npm](#)
- Install **Visual Studio Code**

### STEP-BY-STEP SETUP

#### 1. Install Create-React-App Tool

npm install -g create-react-app

#### 2. Create React App

npx create-react-app myfirstreact

#### 3. Navigate to the App Directory

cd myfirstreact

#### 4. Open in Visual Studio Code

code .

## 5. Edit App.js

- Go to src/App.js
- Delete the existing code
- Paste the following:

```
import React from 'react';
function App() {
```

```
return (
  <div>
    <h1>Welcome to the first session of React</h1>
  </div>
);
}

export default App;
```

```
src > JS App.js > ...
1 import React from 'react';
2
3 function App() {
4   return (
5     <div>
6       <h1>Welcome to the first session of React</h1>
7     </div>
8   );
9 }
10
11 export default App;
12
```

## 7. Open in Browser

Go to : <http://localhost:3000>



# Welcome to the first session of React

## 2.Hands-On Lab: Build “StudentApp” with Class Components

### Create a React App

Open terminal in **VS Code** and run:

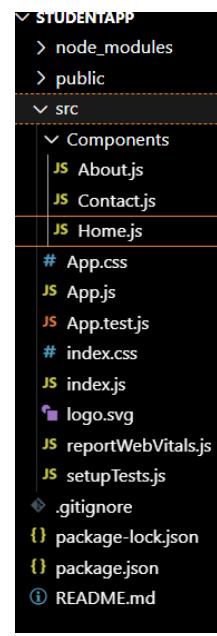
```
npx create-react-app StudentApp
```

```
cd StudentApp
```

## Create "Components" Folder

Inside the src/ folder, create a new folder:

```
src/
  └── Components/
    ├── Home.js
    ├── About.js
    └── Contact.js
```



## Home Component (Class Component)

src/Components/Home.js

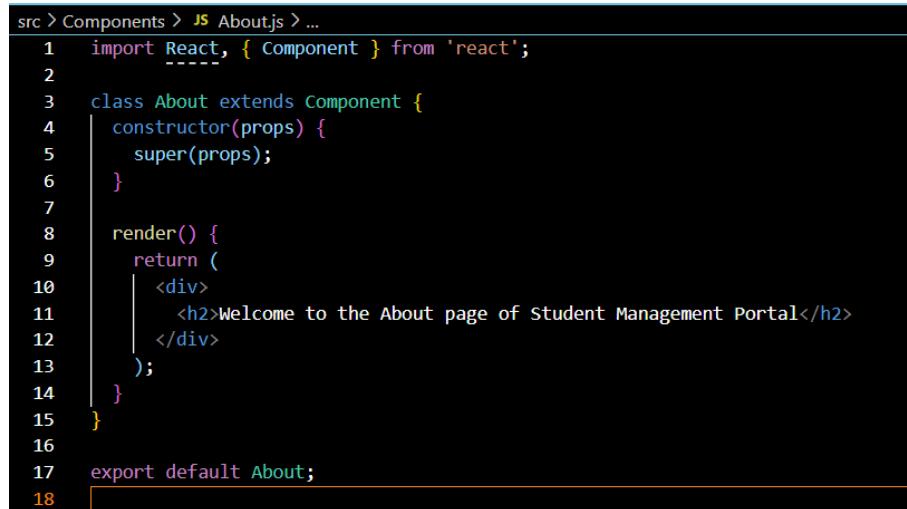
```
import React, { Component } from 'react';
class Home extends Component {
  constructor(props) {
    super(props); // calling parent constructor
  }
  render() {
    return (
      <div>
        <h2>Welcome to the Home page of Student Management Portal</h2>
      </div>
    );
  }
}
export default Home;
```

```
src > Components > JS Home.js > ...
1  import React, { Component } from 'react';
2
3  class Home extends Component {
4    constructor(props) {
5      super(props); // calling parent constructor
6    }
7
8    render() {
9      return (
10        <div>
11          <h2>Welcome to the Home page of Student Management Portal</h2>
12        </div>
13      );
14    }
15  }
16
17  export default Home;
```

## About Component (Class Component)

src/Components/About.js

```
import React, { Component } from 'react';
class About extends Component {
  constructor(props) {
    super(props);
  }
  render() {
    return (
      <div>
        <h2>Welcome to the About page of Student Management Portal</h2>
      </div>
    );
  }
  export default About;
```



```
src > Components > JS About.js > ...
1 import React, { Component } from 'react';
2
3 class About extends Component {
4   constructor(props) {
5     super(props);
6   }
7
8   render() {
9     return (
10       <div>
11         <h2>Welcome to the About page of Student Management Portal</h2>
12       </div>
13     );
14   }
15 }
16
17 export default About;
18
```

## Contact Component (Class Component)

src/Components/Contact.js

```
import React, { Component } from 'react';
class Contact extends Component {
  constructor(props) {
    super(props);
  }
  render() {
```

```

return (
  <div>
    <h2>Welcome to the Contact page of Student Management Portal</h2>
  </div>
);

}

export default Contact;

```

```

src > Components > JS Contact.js > ...
1 import React, { Component } from 'react'
2
3 class Contact extends Component {
4   constructor(props) {
5     super(props);
6   }
7
8   render() {
9     return (
10       <div>
11         <h2>Welcome to the Contact page
12       </div>
13     );
14   }
15
16
17 export default Contact;
18

```

### Edit App.js to Call All Components

src/App.js

```

import React from 'react';
import './App.css';

import Home from './Components/Home';
import About from './Components/About';
import Contact from './Components/Contact';

function App() {
  return (
    <div className="App">
      <h1>Student Management Portal</h1>
      <Home />
      <About />
      <Contact />
    </div>
  );
}

export default App;

```

```

src > JS App.js > ...
1 import React from 'react';
2 import './App.css';
3
4 import Home from './Components/Home';
5 import About from './Components/About';
6 import Contact from './Components/Contact';
7
8 function App() {
9   return (
10     <div className="App">
11       <h1>Student Management Portal</h1>
12       <Home />
13       <About />
14       <Contact />
15     </div>
16   );
17
18
19 export default App;
20

```

## Student Management Portal

Welcome to the Home page of Student Management Portal

### Run the React App

In the terminal : npm start

Welcome to the About page of Student Management Portal

Welcome to the Contact page of Student Management Portal

### 3.Hands-On Lab: Build “ScoreCalculatorApp” with Functional Component

#### 1. Create a React App

```
npx create-react-app scorecalculatorapp
```

```
cd scorecalculatorapp
```

#### 2. Create "Components" Folder

Inside the src/ folder, create a new folder:

```
src/  
└── Components/  
    └── CalculateScore.js
```

#### 3. Create the CalculateScore Functional Component

##### src/Components/CalculateScore.js

```
import React from 'react';  
  
import './Stylesheets/mystyle.css';  
  
function CalculateScore({ name, school, total, goal }) {  
  const average = total / goal;  
  
  return (  
    <div className="score-card">  
      <h2>Score Calculator</h2>  
      <p><strong>Name:</strong> {name}</p>  
      <p><strong>School:</strong> {school}</p>  
      <p><strong>Total:</strong> {total} Marks</p>
```

```

<p><strong>Number of Subjects:</strong> {goal}</p>
<p><strong>Average Score:</strong> {average.toFixed(2)}</p>
</div>
);
}

export default CalculateScore;

```

#### 4. Add Styling

Create a folder Stylesheets/  
and a file named mystyle.css:

**src/Stylesheets/mystyle.css**

```

.score-card {
  background-color: #f2f2f2;
  padding: 20px;
  margin: 40px auto;
  width: 50%;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
  font-family: Arial, sans-serif;
}

.score-card h2 {
  color: #333;
}

.score-card p {
  font-size: 16px;
}

```

```

1 import React from 'react';
2 import './Stylesheets/mystyle.css'; // Make sure your path is correct
3
4 function calcScore(total, goal) {
5   const average = total / goal;
6   return ` ${average.toFixed(2)}%`;
7 }
8
9 export const CalculateScore = ({ Name, School, total, goal }) => (
10   <div className="formatstyle">
11     <h1><font color="Brown">Student Details:</font></h1>
12     <div className="Name">
13       <b><span>Name: </span></b>
14       <span>{Name}</span>
15     </div>
16     <div className="School">
17       <b><span>School: </span></b>
18       <span>{School}</span>
19     </div>
20     <div className="Total">
21       <b><span>Total: </span></b>
22       <span>{total}</span>
23       <span> Marks</span>
24     </div>
25     <div className="Score">
26       <b>Score: </b>
27       <span>{calcScore(total, goal)}</span>
28     </div>
29   </div>
30 );

```

```

rc > Stylesheets > # mystyle.css > ...
1 .Name {
2   font-weight: 300;
3   color: #blue;
4 }
5
6 .School {
7   color: #crimson;
8 }
9
10 .Total {
11   color: #darkmagenta;
12 }
13
14 .formatstyle {
15   text-align: center;
16   font-size: large;
17 }
18
19 .Score {
20   color: #forestgreen;
21 }
22

```

## 5. Edit App.js to Use Component

### src/App.js

```
import React from 'react';
import './App.css';
import CalculateScore from './Components/CalculateScore';
function App() {
  return (
    <div className="App">
      <CalculateScore name="Sparshak Ghosh" school="XYZ High School" total="472" goal="5" />
    </div>
  );
}
export default App;
```

## 6. Run the React App:      npm start

## 7. View in Browser

Open browser and type: <http://localhost:3000>

Score Calculator  
**Name:** Sparshak Ghosh  
**School:** XYZ High School  
**Total:** 472 Marks  
**Number of Subjects:** 5  
**Average Score:** 94.40

## 4. Hands-On Lab: Understanding Component Lifecycle in React (Class-Based)

### Step 1: Create React App

```
npx create-react-app blogapp
```

```
cd blogapp
```

### Step 2: Create Post.js File

Inside src/, create a file called Post.js and add:

```
import React from 'react';

class Post extends React.Component {
  render() {
    return (
      <div>
        <h3>{this.props.title}</h3>
        <p>{this.props.body}</p>
      </div>
    );
  }
}

export default Post;
```

### Step 3: Create Posts.js Component

Create a new file Posts.js inside src/ folder and define a **class component**:

```
import React, { Component } from 'react';
import Post from './Post';

class Posts extends Component {
  constructor(props) {
    super(props);
    this.state = {
      posts: [],
      error: null,
    };
  }
}
```

#### Step 4: Create loadPosts() Method

```

loadPosts = () => {
  fetch('https://jsonplaceholder.typicode.com/posts')
    .then(response => {
      if (!response.ok) {
        throw new Error("Network response was not ok");
      }
      return response.json();
    })
    .then(data => this.setState({ posts: data }))
    .catch(error => this.setState({ error }));
};

// Fetch posts from JSON API and update state
loadPosts() {
  fetch('https://jsonplaceholder.typicode.com/posts')
    .then(res => res.json())
    .then(data => {
      const postList = data.map(p => new Post(p.id, p.title, p.body));
      this.setState({ posts: postList });
    });
}

```

#### Step 5: Use componentDidMount()

```

componentDidMount() {
  this.loadPosts();
}

// React lifecycle method - called after component is mounted
componentDidMount() {
  this.loadPosts();
}

```

#### Step 6: Implement componentDidCatch()

```

componentDidCatch(error, info) {
  alert("An error occurred: " + error.message);
}

// React lifecycle method - error boundary
componentDidCatch(error, info) {
  alert("An error occurred: " + error);
}

```

### Step 7: Implement render() Method

```

render() {
  const { posts, error } = this.state;
  if (error)
    return <p>Error loading posts!</p>;
  return (
    <div>
      <h2>Blog Posts</h2>
      {posts.slice(0, 10).map(post => (
        <Post key={post.id} title={post.title} body={post.body} />
      ))}
    </div>
  );
}

export default Posts;

```

```
// Render method to display all posts
render() {
  return (
    <div>
      <h1>Blog Posts</h1>
      {this.state.posts.map(post => (
        <div key={post.id}>
          <h3>{post.title}</h3>
          <p>{post.body}</p>
        </div>
      )));
    </div>
  );
}
```

### Step 8: Update App.js

Edit src/App.js to include Posts:

```

import React from 'react';
import './App.css';
import Posts from './Posts';

function App() {
  return (
    <div className="App">
      <h1>Welcome to BlogApp</h1>
      <Posts />
    </div>
  );
}

export default App;

```

```
c > JS App.js > ...
1 import React from 'react';
2 import './App.css';
3 import Posts from './Posts';
4
5 function App() {
6   return (
7     <div className="App">
8       <h1>Welcome to BlogApp</h1>
9       <Posts />
10      </div>
11    );
12  }
13
14 export default App;
15
16
```

## Step 9: Run the App

npm start

The screenshot shows a browser window with the title "Welcome to BlogApp" and a subtitle "Blog Posts". Below the subtitle is a large block of Latin placeholder text. The text is organized into several sections with bolded headings and some italicized words. The browser interface includes a top bar with "All Bookmarks" and a vertical scrollbar on the right.

**sunt aut facere repellat provident occaecati excepturi optio reprehenderit**

quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quis totam nostrum rerum est autem sunt rem eveniet architecto

**qui est esse**

um tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

**ea molestias quasi exercitationem repellat qui ipsa sit aut**

et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut

**eum et est occaecati**

am et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hic commodi nesciunt rem tenetur doloremque ipsam iure quis sunt voluptatem rerum illo velit

**nesciunt quas odio**

repudiandaen veniam quaerat sunt sed alias aut fugiat sit autem sed est voluptatem omnis possimus esse voluptatibus quis est aut tenetur dolor neque

**dolorem eum magni eos aperiam quia**

aspernatur corporis harum nihil quis provident sequi mollitia nobis aliquid molestiae perspiciatis et ea nemo ab reprehenderit accusantium quis voluptate dolores velit et doloremque molestiae

**magnam facilis autem**

## 5.Hands-On Lab

### Setup the React App

1. **Unzip** the given React app zip file.
2. Open **Command Prompt/Terminal** and navigate to the app folder.
3. Run: `npm install;` this restores all node packages.
4. Open the folder in **VS Code**:

`code .`

### Step 1: Create CohortDetails.module.css

`css`

`CopyEdit`

```
.container {  
  padding: 20px;  
}  
.title {  
  font-size: 28px;  
}
```

```

font-weight: bold;
margin-bottom: 20px;
}

.cardContainer {
  display: flex;
  gap: 20px;
  flex-wrap: wrap;
}

.card {
  width: 300px;
  border: 1px solid #ccc;
  padding: 15px;
  border-radius: 10px;
  box-shadow: 0 0 4px rgba(0, 0, 0, 0.1);
}

dt {
  font-weight: bold;
}

h3 {
  margin-bottom: 10px;
}

```

```

src > # CohortDetails.module.css > ...
1 .container {
2   padding: 20px;
3 }
4
5 .title {
6   font-size: 28px;
7   font-weight: bold;
8   margin-bottom: 20px;
9 }
10
11 .cardContainer {
12   display: flex;
13   gap: 20px;
14   flex-wrap: wrap;
15 }
16
17 .card {
18   width: 300px;
19   border: 1px solid #ccc;
20   padding: 15px;
21   border-radius: 10px;
22   box-shadow: 0 0 4px rgba(0, 0, 0, 0.1);
23 }
24
25 dt {
26   font-weight: bold;
27 }
28
29 h3 {
30   margin-bottom: 10px;
31 }
32

```

## Step 2: Create CohortDetails.js

```

import React from 'react';

import styles from './CohortDetails.module.css';

const CohortDetails = ({ cohort }) => {

  const titleStyle = {
    color:
      cohort.status === 'Scheduled' ? 'blue' :

```

```

cohort.status === 'Ongoing' ? 'green' : 'black'

};

return (
  <div className={styles.card}>
    <h3 style={titleStyle}>{cohort.name}</h3>
    <dl>
      <dt>Started On</dt>
      <dd>{cohort.startDate}</dd>
      <dt>Current Status</dt>
      <dd>{cohort.status}</dd>
      <dt>Coach</dt>
      <dd>{cohort.coach}</dd>
      <dt>Trainer</dt>
      <dd>{cohort.trainer}</dd>
    </dl>
  </div>
);

export default CohortDetails;

```

```

c > JS CohortDetails.js > ...
1 import React from 'react';
2 import styles from './CohortDetails.module.css';
3
4 const CohortDetails = ({ cohort }) => {
5   const titleStyle = {
6     color:
7       cohort.status === 'Scheduled' ? 'blue' :
8       cohort.status === 'Ongoing' ? 'green' : 'black'
9   };
10
11   return (
12     <div className={styles.card}>
13       <h3 style={titleStyle}>{cohort.name}</h3>
14       <dl>
15         <dt>Started On</dt>
16         <dd>{cohort.startDate}</dd>
17         <dt>Current Status</dt>
18         <dd>{cohort.status}</dd>
19         <dt>Coach</dt>
20         <dd>{cohort.coach}</dd>
21         <dt>Trainer</dt>
22         <dd>{cohort.trainer}</dd>
23       </dl>
24     </div>
25   );
26 }
27
28 export default CohortDetails;
29

```

### Step 3: Update App.js

```

import React from 'react';

import CohortDetails from './CohortDetails';

import styles from './CohortDetails.module.css';

const cohorts = [
  {
    name: 'INTADMDF10 - .NET FSD',
    startDate: '22-Feb-2022',
    status: 'Scheduled',
  }
];

```

```

coach: 'Sparshak',
trainer: 'Jojo Jose'
},
{
  name: 'ADM21JF014 - Java FSD',
  startDate: '10-Sep-2021',
  status: 'Ongoing',
  coach: 'Apoorp',
  trainer: 'Elisa Smith'
},
{
  name: 'CDBJF21025 - Java FSD',
  startDate: '24-Dec-2021',
  status: 'Ongoing',
  coach: 'Sparshak',
  trainer: 'John Doe'
}
];
function App() {
  return (
    <div className={styles.container}>
      <div className={styles.title}>Cohorts Details</div>
      <div className={styles.cardContainer}>
        {cohorts.map((cohort, index) => (
          <CohortDetails key={index} cohort={cohort} />
        )))
      </div>
    </div>
  );
}

export default App;

```

### Cohorts Details

#### INTADMDF10 - .NET FSD

**Started On**  
 22-Feb-2022  
**Current Status**  
 Scheduled  
**Coach** Sparshak  
**Trainer** Jojo Jose

#### ADM21JF014 - Java FSD

**Started On**  
 10-Sep-2021  
**Current Status**  
 Ongoing  
**Coach** Apoorp  
**Trainer** Elisa Smith

#### CDBJF21025 - Java FSD

**Started On**  
 24-Dec-2021  
**Current Status**  
 Ongoing  
**Coach** Sparshak  
**Trainer** John Doe