SVKM's NMIMS University
Mukesh Patel School of Technology Management & Engineering
COURSE: Advanced Web Technologies

Lab - Week 1_B

PART A
(PART A: TO BE REFERRED BY STUDENTS)

Lab Objectives

1. Understand the process of installing Node.js and npm.
2. Set up a React.js project using create-react-app.
3. Run a React.js application and verify the setup.
4. Test React.js with a simple component.

Pre-requisites

1. Basic understanding of HTML, CSS, and JavaScript.
2. A code editor installed (e.g., Visual Studio Code).
3. A modern web browser (e.g., Google Chrome).

1. Node.js Installation

Steps:

1. Open your browser and visit the Node.js official website.
2. Download the **LTS version** for better stability.
3. Install Node.js using the downloaded installer:
    - During installation, ensure the **"Add to PATH"** option is checked.
    - Accept the default installation path and complete the setup.
4. Verify the installation:
    - Open a terminal or command prompt.
    - Type node -v and press Enter. It should display the Node.js version.
    - Type npm -v and press Enter. It should display the npm version.

2. Creating a React.js Application Steps:

1. Open the terminal or command prompt.

Navigate to the folder where you want to create your React app:
cd path/to/your/folder

2. Run the following command to create a new React app:
   npx create-react-app my-app
3. Replace my-app with your desired project name.

4. Wait for the setup to complete. This might take a few minutes as it downloads dependencies.

   Navigate to the project folder:
   cd my-app

5. Start the React development server:
   npm start
6. The default React app should open in your browser at http://localhost:3000.

3. Testing the React Application

Steps:

1. Open the src folder in the my-app project directory.
2. Locate the App.js file and open it in your code editor.

Replace the existing code with the following:

```
import React from 'react';

function App() {
  return (
    <div style={{ textAlign: 'center', marginTop: '50px' }}>
      <h1>Welcome to React!</h1>
      <p>React is working perfectly!</p>
    </div>
  );
}

export default App;
```

3. Save the file. The browser will automatically refresh and display the new content.

4. Adding a Simple React Component (30 minutes)

Steps:

1. Create a new file named HelloComponent.js inside the src folder.

Add the following code:

```
import React from 'react';

function HelloComponent() {
  return (
    <div>
      <h2>Hello from the new Component!</h2>
```

```
      </div>
    );
}

export default HelloComponent;
```

2. Open App.js and modify it to include the new component:
   import React from 'react';

   import HelloComponent from './HelloComponent';

```
function App() {
  return (
    <div style={{ textAlign: 'center', marginTop: '50px' }}>
      <h1>Welcome to React!</h1>
      <p>React is working perfectly!</p>
      <HelloComponent />
    </div>
  );
}

export default App;
```

3. Save all files. The browser will display:
   - The main message: **Welcome to React!**
   - A message from the new component: **Hello from the new Component!**

5. Cleanup and Recap

1. Discuss the structure of a React project:
   - **public/**: Contains static files like index.html.
   - **src/**: Contains all JavaScript and React components.
2. Stop the React server using Ctrl + C in the terminal.
3. Summarize key points:
   - React uses components to build the UI.
   - React automatically refreshes the UI when the code changes.
   - App.js is the root component where other components are included.

SVKM's NMIMS University
Mukesh Patel School of Technology Management & Engineering
COURSE: Advanced Web Technologies

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the portal at the end of the practical. The filename should be **AWT_batch_rollno_experimentno Example: AWT_A1_A001_P1**

Roll No.:	**Name:**
Prog/Yr/Sem:	**Batch:**
Date of Experiment:	**Date of Submission:**

Lab Deliverables

1. A React app with a modified App.js and a new component (HelloComponent).
2. Screenshots of the terminal showing:
    - Successful installation of Node.js and npm.
    - Running the React server.
3. A browser screenshot showing the React app output.

Include Learnings and Conclusion in detail.