R. C. PATEL
INSTITUTE OF TECHNOLOGY
An Autonomous Institute

Department of Artificial Intelligence and Machine Learning

Laboratory Manual

Subject: - Programming Lab-II (Web Development) **Semester: - IV**
Class: - S.Y. B. Tech **Experiment No. : 09**

PRN:	**Roll No.:**
Name:	**Batch:**
Date of Performance: ___/___/2025	**Date of Submission:** ___/___/2025

TITLE: Installation and Configuration of React.

PREREQUISITE: Basic knowledge of software installation.

Theory / Algorithm / Conceptual Description:

React is a popular JavaScript library for building user interfaces, and it's widely used for developing interactive and dynamic web applications. To get started with React, you need to set up your development environment and configure your project correctly. Here, we will explore the installation and configuration of React, providing you with the foundational knowledge to kickstart your React development journey.

Setting Up Your Environment

1. Download and Install Node.js & npm

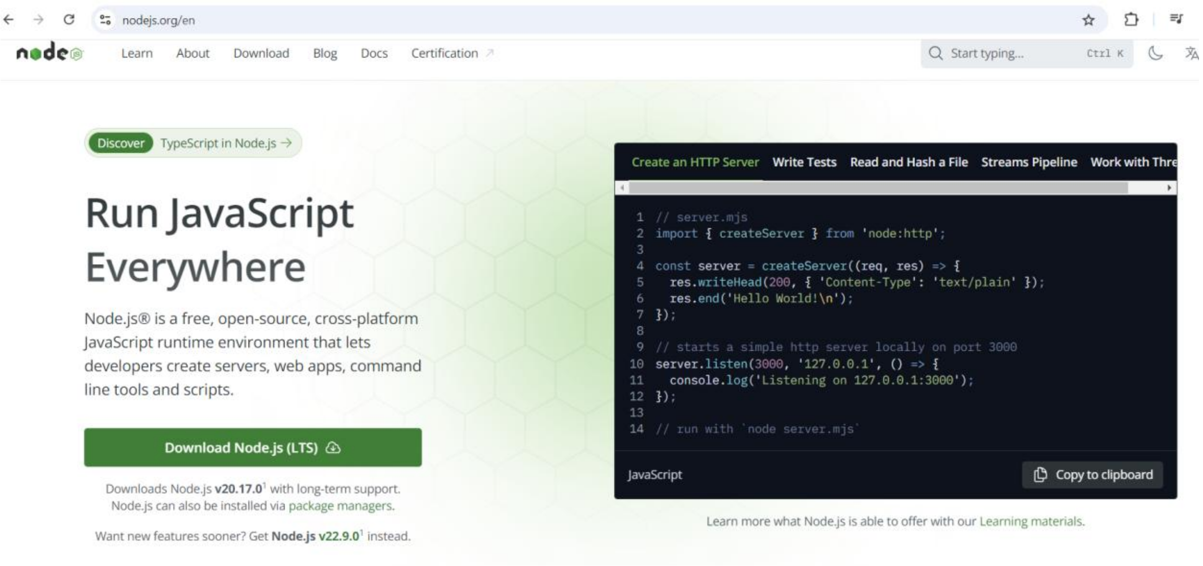
React development relies on Node.js, a JavaScript runtime, and npm (Node Package Manager) for package management.

Node.js should be installed first because React.js is a JavaScript library, and Node.js is a JavaScript runtime environment that allows to run JavaScript on the server side. So when we will writing React, we include JavaScript functions in our React project, and then Node.js helps run this JavaScript code on the web page.

Node.js has various versions. The recommended version is the latest stable version, as it contains major and significant changes. These changes includes bug fixes and security updates, compatibility with your project dependencies, and so on.

To install Node, navigate to the Node.js website. On their webpage, the option to download either the recommended version or the current version, as seen in the image below. After downloading the appropriate version, install it on computer.

Link: https://nodejs.org/en



Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

```javascript
// server.mjs
import { createServer } from 'node:http';

const server = createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Hello World!\n');
});

// starts a simple http server locally on port 3000
server.listen(3000, '127.0.0.1', () => {
  console.log('Listening on 127.0.0.1:3000');
});

// run with `node server.mjs`
```

Downloads Node.js **v20.17.0**[1] with long-term support.
Node.js can also be installed via package managers.

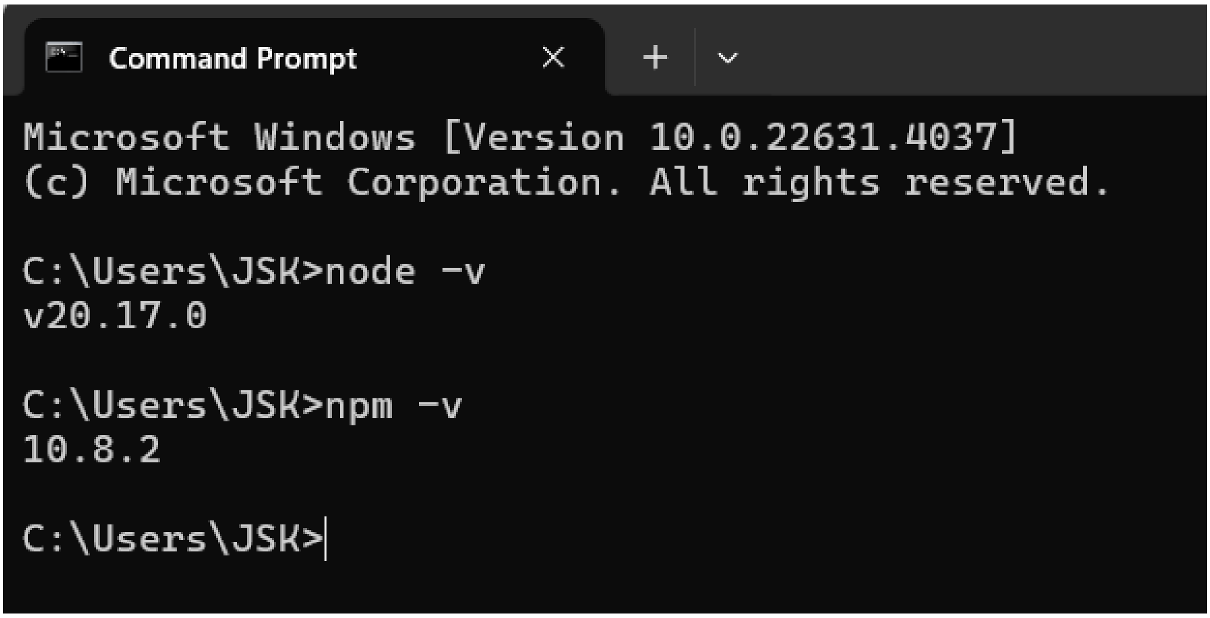
Want new features sooner? Get **Node.js v22.9.0**[1] instead.

Learn more what Node.js is able to offer with our Learning materials.

To verify that Node.js and npm are properly installed, open terminal or command prompt and run the following commands:

node -v

npm –v



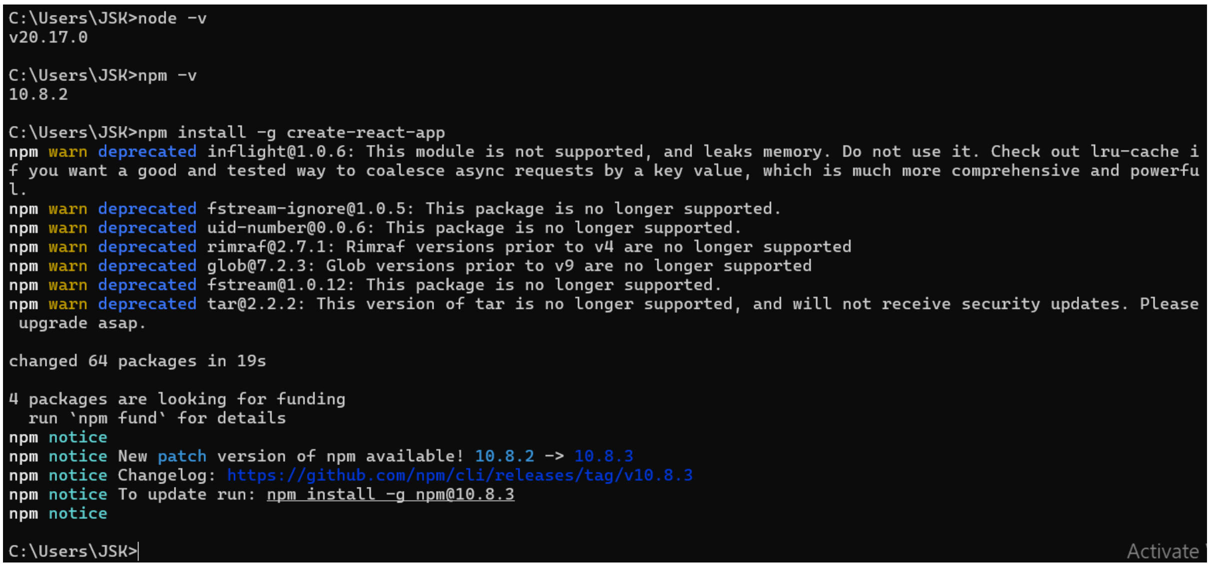
```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JSK>node -v
v20.17.0

C:\Users\JSK>npm -v
10.8.2

C:\Users\JSK>
```

If your Node version is displayed like the above, it means you have successfully installed Node.js on your computer.

2. Install Create-React-App Tool

The next step is to install a tool called create-react-app using NPM. This tool is used to create react applications easily from our system. You can install this at the system level or temporarily at a folder level. We will install it globally by using the following command.

npm install -g create-react-app

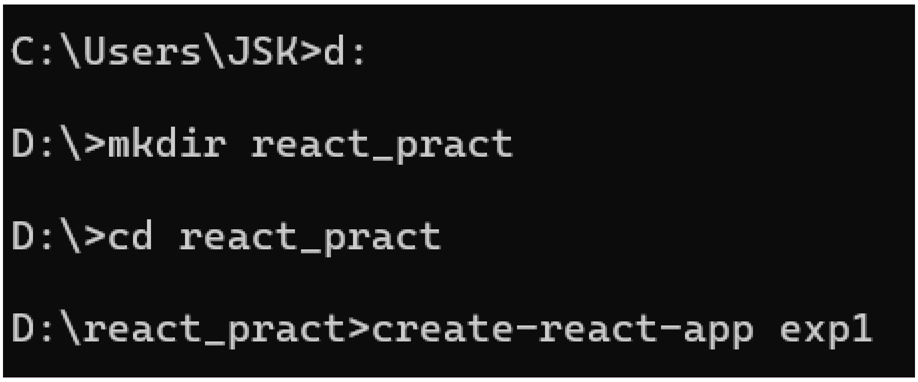


```
C:\Users\JSK>node -v
v20.17.0

C:\Users\JSK>npm -v
10.8.2

C:\Users\JSK>npm install -g create-react-app
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache i
f you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerfu
l.
npm warn deprecated fstream-ignore@1.0.5: This package is no longer supported.
npm warn deprecated uid-number@0.0.6: This package is no longer supported.
npm warn deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated fstream@1.0.12: This package is no longer supported.
npm warn deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please
 upgrade asap.

changed 64 packages in 19s

4 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New patch version of npm available! 10.8.2 -> 10.8.3
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.3
npm notice To update run: npm install -g npm@10.8.3
npm notice

C:\Users\JSK>
```

3. Creating a new react project

After create-react-app is installed, we can create our first react application. Let's say I want to create the project or application in D:\react_pract. I will create this folder and let our command prompt point to it by using the change directory command. Let's create a new Project now using the command.

create-react-app exp1



```
C:\Users\JSK>d:

D:\>mkdir react_pract

D:\>cd react_pract

D:\react_pract>create-react-app exp1
```

Remember not to create the project with an upper case character in it. If you encounter any policy restriction error, solve it as follows.

To resolve this issue and be able to run the create-react-app command, you will need to adjust your PowerShell execution policy to allow script execution. Here's how you can do it:

- Open PowerShell as an administrator. To do this, search for "PowerShell" in the Start menu, right-click on "Windows PowerShell," and select "Run as administrator."
- Check your current execution policy by running the following command:
- Get-ExecutionPolicy
- If the execution policy is set to "Restricted," you need to change it. To allow script execution, you can set the execution policy to "RemoteSigned" or "Unrestricted." For example, to set it to "RemoteSigned," run the following command:

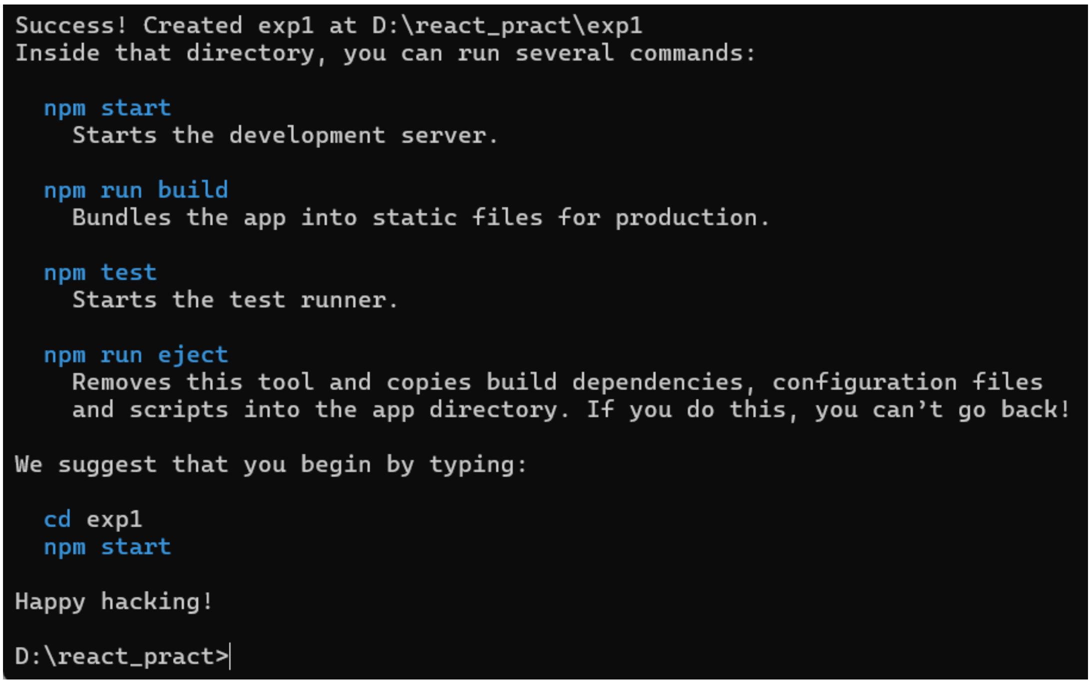
  Set-ExecutionPolicy RemoteSigned

4. Running the React Application

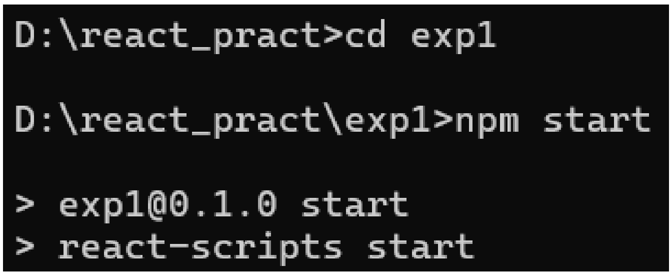
Let's do CD to the Project we have created and run it locally on our system using npm start. Launch the browser and visit http://localhost:3000. We can then see our first React Application response in the browser.

cd exp1

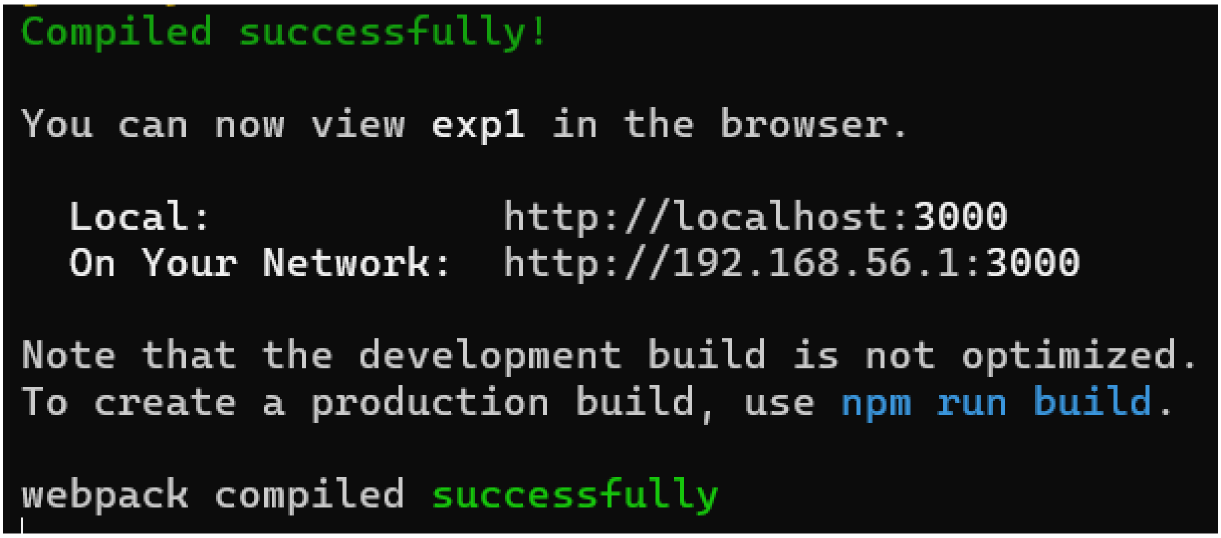
npm start



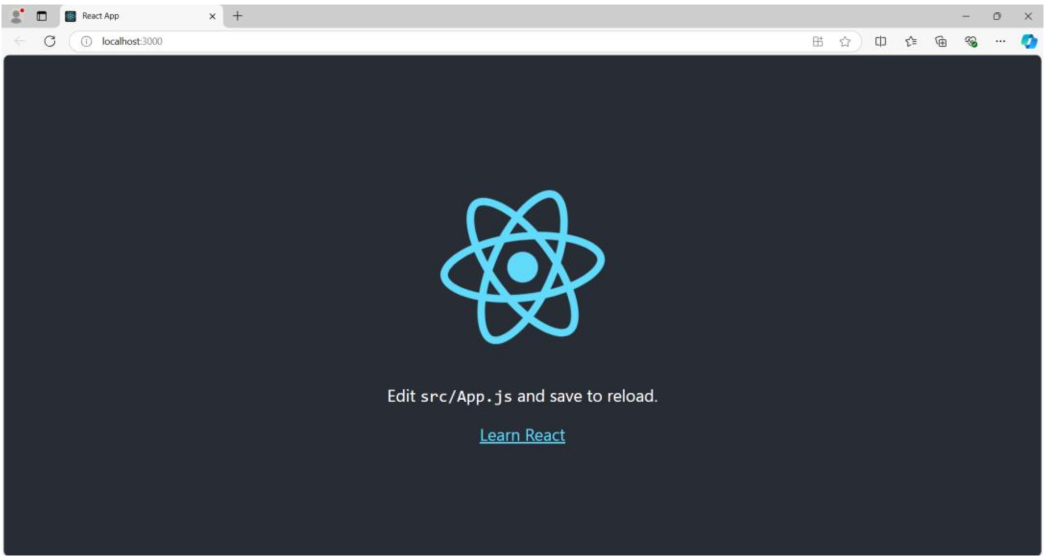
```
Success! Created exp1 at D:\react_pract\exp1
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can’t go back!

We suggest that you begin by typing:

  cd exp1
  npm start

Happy hacking!

D:\react_pract>
```



```
D:\react_pract>cd exp1

D:\react_pract\exp1>npm start

> exp1@0.1.0 start
> react-scripts start
```



```
Compiled successfully!

You can now view exp1 in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



Edit src/App.js and save to reload.

Learn React

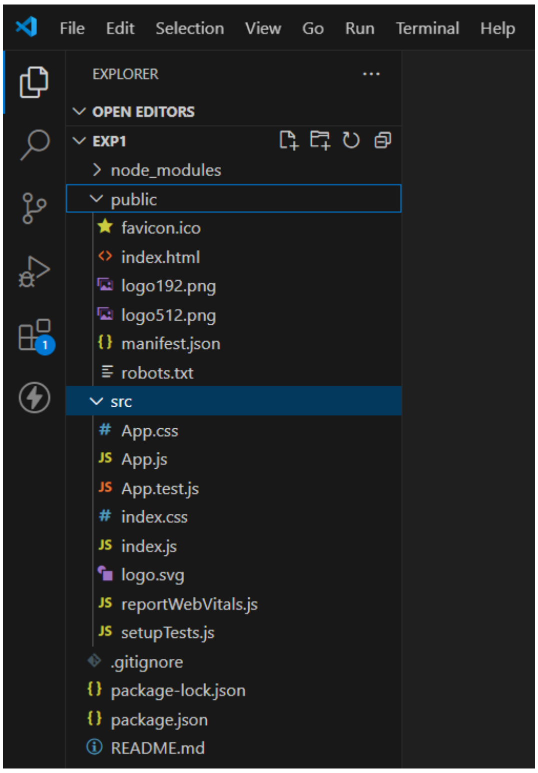
React complete installation in the terminal

We have created a New Project using React and executed the Project. But as a developer, we would be more interested to know about the Project which is created, its structure and we would like to play around with it. So it is time for us to get an Editor. When we think of IDE, we have a variety of choices like Visual Studio Code, React IDE, Sublime Editor, Atom Editor, Webstorm and a few others. We will use the VS Code as our Editor.

5. Install Visual Studio Code

Link: https://code.visualstudio.com/download

Now Start the Visual Studio Code and open the React project in it using *File>Open Folder*. It will be look like this:



```
EXP1
  > node_modules
  v public
      favicon.ico
      index.html
      logo192.png
      logo512.png
      manifest.json
      robots.txt
  v src
      App.css
      App.js
      App.test.js
      index.css
      index.js
      logo.svg
      reportWebVitals.js
      setupTests.js
    .gitignore
    package-lock.json
    package.json
    README.md
```

CONCLUSION: In this experiment, we have installed React onto our system and made our first react project.