

module 1-2

- Q) What is React.js? Explain its features in details.

Ans React.js is an open-source JavaScript library developed by Facebook for building user interfaces, particularly for single-page applications and complex web interfaces.

- It's one of the most popular libraries for frontend development due to its simplicity, flexibility, and high performance.
- React follows a component-based architecture, where UIs are composed of reusable components that manage their own state, making it easier to build complex UIs.

Features:-

- Q) JSX:- JSX stands for JavaScript XML. It is a JavaScript syntax extension. It's an XML or HTML like syntax used by React.js.

- This syntax is processed into JavaScript calls of react framework.
- JSX makes it easier to describe UI components and their structure in a more declarative and intuitive way.

- Q) Components:- React.js is all about components.

- React.js application is made up of multiple components, and each component has its own logic and controls.

- These components can be reusable which help you to maintain the code when working on larger scale projects.

3) one-way Data Binding:- React is designed in such a manner that follows uni-directional data flow or one-way data binding. The benefits of one-way data binding give you better control throughout the application.

If the data flow is in another direction, then it requires additional features.

4) Virtual Dom:- A virtual Dom object is a representation of the original Dom object. It works like a one-way data binding.

when the state of a component changes, React first updates the virtual Dom, then compares it with the previous version to identify the minimal set of Dom operations needed to update the actual Dom.

This minimizes Dom manipulation and improves performance.

5) Simplicity:- React uses JSX file which makes the application simple and the code is as well as understand.

We know that React is a component-based approach which makes the code reusable as you need.

This makes it simple to use and learn.

6) Performance:- As we discussed earlier, React uses Virtual Dom and updates only the modified parts. So, this makes the Dom to run faster.

Dom executes in memory so we can create separate components which makes the Dom fast.

7) conditional statements

JSX allows us to write conditional statements. The data in the browser is displayed according to the conditions provided inside the JSX.

2) How to create React application and explain project structure in details.

Ans How to create React application?

1) install Node.js and NPM! - ensure you have Node.js installed on your system as it comes with NPM

2) install create-react-app! - open your terminal or command prompt and run the following command to install.

- NPM install -g create-react-app

3) Create a new React app! - you can create a new React app by running the following command! -
NPM create-react-app my-app

4) Navigate to the Project directory! - once the app is created, navigate to the Project directory -

- cd my-app

* Project Structure

```
my-app/
    └── node-modules/
    └── public/
        └── favicon.ico
        └── index.html
```

```
|- src/
  |- components/
    |- APP.js
    |- ...
    |- APP.css
    |- APP.test.js
    |- index.js
    |- index.css
    |- ...
  |- .gitignore
  |- package.json
  |- package-lock.json
  |- ...
  |- ...
```

- node_modules :- This directory contains all the dependencies installed by NPM or Yarn. You don't need to modify anything inside this directory.
- public :- This directory contains static assets such as HTML files, images, and the favicon. The 'index.html' file is the main HTML file that serves as the entry point for your React application.
 - it contains a <div id="root"> where React will render your components.
- src :- This directory contains the source code of your react application.
 - components :- This directory contains react components. APP.js is the main component

that serves as the root of your application.

- app.css : CSS styles specific to the 'App' component.
- app.test.js : Test file for the 'App' component.
- index.css : Contains global CSS styles that are applied to the entire application.
- index.js : The entry point of the application, where React is initialized and the main 'App' component is rendered.
- .gitignore : This file specifies which files and directories should be ignored by git.
- package.json : Contains project configuration, dependencies, and scripts defined for running various tasks.

5 marks

- 1) describe about creating react application with dependencies as environment with set of commands:

Ans

Step 1:- install Node.js and npm :-

if you haven't already, you need to install node.js which includes npm.

Step 2:- Create a new react application :-

Open your terminal or command prompt and run the following command to create a new react application using create react app:-

- `npm create-react-app my-app`

Step 3:- After the project is created, navigate to the project directory using the 'cd' command:

- `cd my-app`

Step 4:- install additional dependencies:-

If you need to install additional dependencies, such as react routes for routing or axios for making HTTP requests, you can do so using npm.

Ct

Npm install react-router-dom axios

Step 5: Start the development server!

Once you have installed the necessary dependencies, you can start the development server to see your react application in action.

From NPM Script

⑥ Build for production!

When you're ready to deploy your React application to production, you can build it using the following command:

- npm run build

- That's it! You've now created a React application with dependencies and set up your development environment.

2) Explain `Package.json` file with its attributes and purpose.

Ans

`Package.json` is a metadata file in a `Node.js` project that describes the project's dependencies, scripts, configuration, and other details.

It typically contains information about the project such as its name, version, author, and license.

The `package.json` file is typically located in the root directory of a `Node.js` project and is automatically generated when you run `npm init` command to initialize a new project.

* Attributes and Purpose:

① name:- specifies the name of the package. It should be unique within the npm registry. For a project, it typically matches the project's directory name.

② version:- specifies the version of the package. e.g ('1.0.0').

③ description:- provides a brief description of the package. It helps users understand the purpose of functionality of the project.

These are ~~these dependencies, ~~script + script~~ command and ~~script + script~~ scripts in our js~~

4) main: specifies the entry point of the package. For most Node.js projects, this is typically 'index.js' or 'app.js'.

5) scripts: defines various scripts to be run with npm. Common scripts include 'start' for starting the application, 'test' for running tests, and 'build' for building the project for production.

- users can also define custom scripts for specific tasks.

6) dependencies: lists the dependencies required for the project to run in production.

- these are packages that are necessary for the application to function properly.

7) keywords: provides an array of keywords that describe the package. These keywords help users discover the package when searching on npm.

8) author: This attribute specifies the author of the package. It can be a string or an object.

- 3) what is npm, and how to use in React.js
list out different commands with purpose.

Ans

Npm, which stands for Node Package manager, is a package manager for Node.js packages or modules.

It is primarily used for installing, sharing, and managing dependencies in JavaScript projects.

In the context of React.js, npm is often used to manage project dependencies and scripts.

Npm commands:-

1) NPM init! - initializes a new 'package.json' file for your project.

2) NPM install! - installs all dependencies listed in the 'package.json' file.

It reads the 'dependencies' and 'devDependencies' sections and installs all the required packages locally in the 'node_modules' directory.

3) NPM install <Package-name>! - installs a specific package and adds it to the 'dependencies' section of the 'package.json' file.

Eg:- NPM install react.

a) npm start → runs the script specified in the script field of the scripts section in package.json.

- In many React.js projects, this command is used to start the development server.

b) npm run build → This command is used to build the production-ready version of react.js application.

c) npm update → updates all the packages listed in the dependencies section of the package.json file to the latest version according to the version ranges specified in the package.json file.

- npm update updates the package.json and package-lock.json files accordingly.

d) npm test → This command runs tests for your React.js application.

- This script typically runs unit tests, integration tests, or end-to-end tests depending on your testing setup.

e) npm audit → checks for known vulnerabilities in the dependencies of your project and provides recommendations for resolving them.

f) npm publish → This command is used to publish a package to the npm registry.

v) describe history of react JS and application scope of it.

Ans

- React is the most powerful Javascript library, which is used for building user interfaces.
- It was developed by Facebook and is often used for building single-page applications and mobile applications.
- React was originally released first in 2013, and since then, it has become one of the most popular Javascript libraries for building user interfaces.
- It was developed by Jordan Walke, a software engineer at Facebook, and was initially released as an open-source project on GitHub.
- React quickly gained popularity due to its component-based architecture, virtual DOM, and efficient rendering.

* application Scope!

1) user interfaces! - React.js is primarily used for building dynamic and interactive user interfaces for web applications.

2) single page applications (SPAs)! - React.js is well-suited for developing SPAs where content is dynamically loaded and updated without requiring full page reloads.

- 3) Component-Based Architecture! - React.js promotes a modular approach to UI development, making it ideal for large-scale applications where components can be reused and maintained easily.
- 4) Cross-Platform Development! - With frameworks like React Native, React Native extends its scope to building mobile applications for iOS and Android platforms.
- 5) Real-time Data Applications! - React.js is often used in conjunction with other technologies like Redux or GraphQL to build real-time data applications such as dashboards or collaborative tools.