**REACT JS**

1. What is React Js?

* The React.js framework is an open-source JavaScript framework and library developed by Facebook. It’s used for building interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript.
* In React, you develop your applications by creating reusable components that you can think of as independent Lego blocks. These components are individual pieces of a final interface, which, when assembled, form the application’s entire user interface.
* React’s primary role in an application is to handle the view layer of that application just like the V in a model-view-controller (MVC) pattern by providing the best and most efficient rendering execution. Rather than dealing with the whole user interface as a single unit, React.js encourages developers to separate these complex UIs into individual reusable components that form the building blocks of the whole UI.
* In doing so, the ReactJS framework combines the speed and efficiency of JavaScript with a more efficient method of manipulating the DOM to render web pages faster and create highly dynamic and responsive web applications.

1. What is NPM in React Js?

* In React.js, NPM stands for Node Package Manager. NPM is a command-line tool that allows developers to manage and install third-party packages or libraries in their projects. It serves as a central repository for reusable JavaScript code, making it easy for developers to discover, share, and reuse code.
* When working with React.js, NPM is commonly used to install and manage dependencies required by a React project. Dependencies are external packages or libraries that your project relies on to function properly. By using NPM, you can easily add, update, or remove these dependencies as needed.
* To initialize a new React project, you can use the `create-react-app` command, which sets up a basic project structure and configuration files. Once the project is created, you can use NPM commands to install additional packages. The dependencies are listed in a file called `package.json`, which is automatically generated and updated by NPM.
* NPM also provides features such as version management, dependency resolution, script execution, and more. It allows you to easily manage the packages required by your React.js project, making development more efficient and organized.

1. What is Role of Node Js in react Js?

* NodeJS is a framework of JavaScript which is mainly used for working with the backend of our application or building the backend using JavaScript, whereas ReactJS is a JavaScript front-end library. It is mainly used for building the user interface or the frontend of our application.

1. What is CLI command In React Js?

* In React.js, the CLI command refers to the Command-Line Interface commands that are used to manage and interact with React.js projects. The React.js CLI command is called "create-react-app" and is used to create a new React.js application with a preconfigured setup.









1. What is Components in React Js?

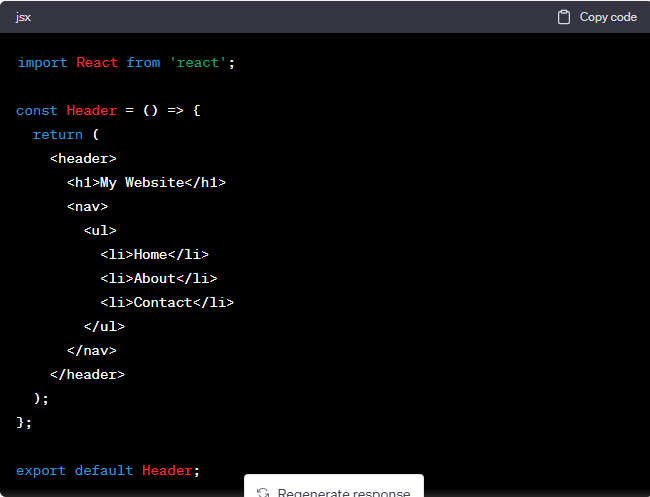
* Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.
* Components come in two types, Class components and Function components, in this tutorial we will concentrate on Function components.

1. What is Header and Content Components in React Js?

* In React.js, header and content components refer to two distinct parts of a user interface that can be created and rendered using React components.

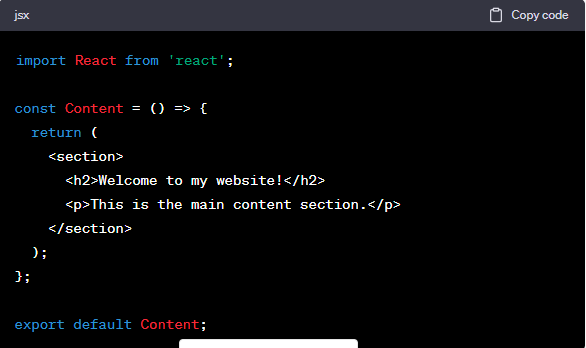
1. Header Component: The header component typically represents the top section of a web page or a section within a page.

Here's an example of a basic header component in React:



1. Content Component: The content component represents the main section of a web page or a specific area within a page where the primary content is displayed.

Here's an example of a basic content component in React:



* These are just basic examples, and the actual implementation of header and content components can vary depending on the specific requirements of your application. In a larger application, you may have multiple header and content components that are composed together to create complex user interfaces.

1. How to install React Js on Windows, Linux Operating System? How to Install NPM and How to check version of NPM?

* To install the full React toolchain on WSL, we recommend using create-react-app: Open a terminal(Windows Command Prompt or PowerShell). Create a new project folder: mkdir ReactProjects and enter that directory: cd ReactProjects .
* Npx is the package runner used by npm to execute packages in place of a global install.

1. How to check version of React Js?

* To check which React version is your project using you need to open the package. json.
* Take a look under the dependencies section. It should list all of the dependencies of your project and one of those should be React.

1. How to change in components of React Js?

* To make changes in components in React.js, you need to follow these general steps:
* 1. Locate the component:
* Identify the specific component that you want to modify. Components in React.js are typically defined as separate JavaScript files, often with a `.jsx` or `.js` extension. Locate the file containing the component you want to change.
* 2. Open the component file:
* Use a text editor or an integrated development environment (IDE) to open the component file. This allows you to view and modify the code of the component.
* 3. Make the desired changes:
* Within the component file, you can modify the JSX syntax and JavaScript code to implement the changes you want. You can update the component's structure, add or remove elements, change styling, modify data handling, or incorporate new functionality. Be mindful of the component's logic and ensure that any changes are coherent with the overall purpose and behavior of the component.

4. Save the changes:

* After making the desired modifications, save the component file.

5. Test the changes:

* Run your React application and navigate to the relevant part of your application where the component is used. Verify that the changes you made have the desired effect. You can test your React application by running the development server using `npm start` or a similar command depending on your setup.
* By following these steps, you can change the components in React.js and observe the updated behavior and appearance of your application. It's important to note that when modifying components, it's best to have a solid understanding of React.js concepts and its component-based architecture to ensure efficient and maintainable code.