**Frontend Assignment**

**React Js**

**MODULE: 1**

**HTML-CSS Login Registration page**

**ANS.**  HTML-CSS Login Registration page in below link:

HTML :<https://github.com/Chigs18/Assignment/blob/main/singnIn-Up.html>

CSS: <https://github.com/Chigs18/Assignment/blob/main/css/singnin.css>

**MODULE: 2**

**(JavaScript Essentials)**

**Calculate subtotal price of quantity in JavaScript?**

**ANS.**  The answer is in link:

<https://github.com/Chigs18/Assignment/blob/main/SubTotal.html>

**What is JavaScript Output method?**

**ANS.**  JavaScript can "display" data in different ways: Writing into an HTML element, using innerHTML. Writing into the HTML output using document.write(). Writing into an alert box, using window.alert(). Writing into the browser console, using console.log().

**How to used JavaScript Output method?**

**ANS.**

1. Writing into an HTML element, using innerHTML.
2. Writing into the HTML output using document.write() .
3. Writing into an alert box, using window.alert().
4. Writing into the browser console, using console.log().

**How to used JavaScript Events to do all examples?**

**ANS.** In JavaScript there are many Events as following:

**Mouse Event:**

|  |  |  |
| --- | --- | --- |
| **Event Performed** | **Event Handler** | **Description** |
| click | onclick | When mouse click on an element |
| mouseover | onmouseover | When the cursor of the mouse comes over the element |
| mouseout | onmouseout | When the cursor of the mouse leaves an element |
| mousedown | onmousedown | When the mouse button is pressed over the element |
| mouseup | onmouseup | When the mouse button is released over the element |
| mousemove | onmousemove | When the mouse movement takes place. |

1. **Click event :** When mouse click on an element

Example:

**<form>**

**<input** type="button" onclick="clickevent()" value="Who's this?"**/>**

**</form>**

**<script** language="Javascript" type="text/Javascript"**>**

    function clickevent()

    {

        document.write("This is click event");

    }

**</script>**

**Keyboards Events:**

|  |  |  |
| --- | --- | --- |
| **Event Performed** | **Event Handler** | **Description** |
| Keydown & Keyup | onkeydown & onkeyup | When the user press and then release the key |

Example:

**<input** type="text" id="input1" onkeydown="keydownevent()"**/>**

**<script>**

    function keydownevent()

    {

        document.getElementById("input1");

        alert("Pressed a key");

    }

**</script>**

**Form Events:**

|  |  |  |
| --- | --- | --- |
| **Event Performed** | **Event Handler** | **Description** |
| focus | onfocus | When the user focuses on an element |
| submit | onsubmit | When the user submits the form |
| blur | onblur | When the focus is away from a form element |
| change | onchange | When the user modifies or changes the value of a form element |

Example:

**<input** type="text" id="input1" onfocus="focusevent()"**/>**

**<script>**

   function focusevent()

    {  document.getElementById("input1").style.background=" aqua";   }

**</script>**

**MODULE: 3**

**(JavaScript Essentials)**

**What is React Js?**

**ANS.** React is a free and open-source front-end JavaScript library for building user interfaces based on components. React 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.

**What is NPM in React Js?**

**ANS.** NPM is short for node package manager, an online directory that contains the various already registered open-source packages. NPM modules consume the various functions as a third-party package when installed into an app using the NPM command npm install.

**What is Role of Node Js in react Js?**

**ANS.** 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.

Node. js is widely used for the back-end of applications, like using Express. js to build the back-end of classic web applications. Also, it is used for server-side programming and non-blocking, event-driven servers like typical websites and backend API services.

**What is CLI command In React Js ?**

**ANS.**

* Creating a new application. Create React App provides multiple ways to create React application.
* Selecting a template. Create React App creates React application using default template.
* Installing a dependency.
* Running the application.

**What is Components in React Js?**

**ANS.** Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.

Every React component have their own structure, methods as well as APIs. They can be reusable as per your need. For better understanding, consider the entire UI as a tree. Here, the root is the starting component, and each of the other pieces becomes branches, which are further divided into sub-branches.

**What is Header and Content Components in React Js?**

**ANS.** A header component defines a reusable custom header that can be sent in an API request or returned in an API response.

A Content 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.

**How to install React Js on Windows, Linux Operating System?**

**ANS.** To install the full React toolchain on WSL, we recommend using create-react-app:

1. Open a terminal(Windows Command Prompt or PowerShell).
2. Create a new project folder: mkdir ReactProjects and enter that directory: cd ReactProjects.
3. Install React using create-react-app, a tool that installs all of the dependencies to build and run a full React.js application:

npx create-react-app my-app

1. This will first ask for your permission to temporarily install create-react-app and it's associated packages. Once completed, change directories into your new app ("my-app" or whatever you've chosen to call it): cd my-app.
2. Start your new React app:

npm start

**How to install NPM and How to check version of NPM?**

**ANS. To install a package, npm uses the following algorithm:**

1. load the existing node\_modules tree from disk.
2. clone the tree.
3. fetch the package.json and assorted metadata and add it to the clone.
4. walk the clone and add any missing dependencies.
5. dependencies will be added as close to the top as is possible.

To see the most current version of a package in the npm repository, use the npm view npm get version of package-name version command.

**How to check version of React Js?**

**ANS.** 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.

**How to change in components of React Js?**

**ANS.** To change a value in the state object, use the this.setState() method. When a value in the state object changes, the component will re-render, meaning that the output will change according to the new value(s).

**How to Create a List View in React Js?**

**ANS.** To List view in react check below link:

<https://github.com/Chigs18/Assignment/tree/main/react/src>

**Create Increment decrement state change by button click?**

**ANS.** Show this program below link:

<https://github.com/Chigs18/Assignment/tree/main/react/src>

**MODULE: 4**

**(List and Hooks)**

**Explain Life cycle in Class Component and functional component with Hooks**

**ANS.**

Each component in React has a lifecycle which you can monitor and manipulate during its three main phases.

The three phases are: **Mounting**, **Updating**, and **Unmounting**.

Mounting

Mounting means putting elements into the DOM.

React has four built-in methods that gets called, in this order, when mounting a component:

1. constructor()
2. getDerivedStateFromProps()
3. render()
4. componentDidMount()

The render() method is required and will always be called, the others are optional and will be called if you define them.

## React component lifecycle with hooks

You can take advantage of the *useEffect* hook to achieve the same results as with the *componentDidMount*, *componentDidUpdate*and *componentWillUnmount*methods. *useEffect*accepts two parameters. The first one is a callback which runs **after render**, much like in *componentDidMount*. The second parameter is the effect dependency array. If you want to run it on mount and unmount only, pass an empty array *[]*.

**MODULE: 5 (Styling & Advance React)**

**Create Shopping site home page with Styled- component.**

**ANS.**