1. Contents

HTTPS://T.ME/PROGRAMMING_AND_ALGORITHMS_EN
@RAMTIN_BA
RAMTINBA145822@GMAIL.COM

2. React and JSX Fundamentals

1. Create a react app using Vite:

My node version on the Ubuntu WSL was outdated so I ran this command to install the latest long term support version of it:

```
$ nvm install -lts
```

```
$ node -v
```

```
v20.14.0
```

In order to create a Vite project this command should be used and the Vite should be followed:

```
$ npm create vite
```

In order not having to install npm packages every time for each project and using the computer hard disk unnecessary und inefficient, pnpm instead will be used. Pnpm uses hard links and symlinks to save one version of a module only ever once on a disk. for installing it, this command have been used:

```
$ wget -qO- https://get.pnpm.io/install.sh | sh -
```

After that, instead of `npm install`, `pnpm install` will be used.

For running the program using pnpm, the following command should be used:

```
$ pnpm run dev
```

For running the program using npm, the following command should be used:

```
$ npm run dev
```

Project can also be run by using the Vite command:

```
$ npx vite
```

HTTPS://T.ME/PROGRAMMING_AND_ALGORITHMS_EN
@RAMTIN_BA
RAMTINBA145822@GMAIL.COM

2. If-else statement in JSX using ternary Operator

```
import React from 'react';

function JSX_Conditional_Rendering_Using_Ternary_Operator() {
    let status = false
    return (
        <div>
        <h1>JSX_Conditional_Rendering_Using_Ternary_Operator:</h1>
            {status ?
            /* If it is true:*/
            <button>Logout</button>
            :
            /* If it is false:*/
            <button>Login</button>}
        </div>
    );
};

export default
JSX_Conditional_Rendering_Using_Ternary_Operator;
```

HTTPS://T.ME/PROGRAMMING_AND_ALGORITHMS_EN
@RAMTIN_BA
RAMTINBA145822@GMAIL.COM

3. Anonymous Function

```
import { useState, React } from 'react'

function App() {
  let points =75;
  return (
    <div>
      {(
        () => {
          /* Functions Body: */
          if(points>=80 && points<100){
            return <h1>A+</h1>
          }
          else if(points>=70 && points<80){
            return <h1>A-</h1>
          }
          else if(points>=60 && points<70){
            return <h1>B</h1>
          }
          else {
            return <h1>Failed</h1>
          }
        }
        /* Calling the function in this section:*/
      ) ()}
    </div>
  );
};
export default App;
```

4. JSX For-Loop

Using map function and then calling an anonymous function inside the map-function:

- ❖ Don’t forgot to add ‘return’ to the map function!

```
import React from "react";
function JSX_Loop() {
  const cars = ["Honda", "Ford", "Toyota"];
  return (
      <ul>
        {cars.map( (item, index) => { return (<li key={index.toString()}> {item} </li>) } ) }
      </ul>
  );
}
export default JSX_Loop;
```

5. JSX Conditional Rendering Using IF-Else:

In this code, more efficient and cleaner way, of implementing If-Else by using functions is illustrated:

```
import React from 'react';
function loginButton(isLoggedIn) {
    if (isLoggedIn){
        return<button>Logout</button>
    }
    else{
        return<button>Login</button>
    }
}
function JSX_Conditional_Rendering_Using_If_Else(){
    return (
        <div>
            {loginButton(true)}
        </div>
    );
};
export default JSX_Conditional_Rendering_Using_If_Else;
```

HTTPS://T.ME/PROGRAMMING_AND_ALGORITHMS_EN
@RAMTIN_BA
RAMTINBA145822@GMAIL.COM

6. JSX Conditional Rendering Using Switch Statement:

```
import React from 'react';
function JSX_Conditional_Rendering_Using_switch(){
    const status = true;

    switch(status){
        case true: return <button>Log out</button>;
        case false: return <button>Log in</button>;
        default: return null;

    }

};
export default JSX_Conditional_Rendering_Using_switch;
```

7. JSX Conditional Rendering Using && Operator:

If condition is true, it will execute the code after the '&&' operator, otherwise it won't execute something else.

```
import React from 'react';

function JSX_Conditional_Rendering_Using_And_And_Operator() {
    let status = true
    return (
        <div>
        <h1>JSX_Conditional_Rendering_Using_And_And_Operator</h1>
        {status && <button>Logout</button>}
        </div>
    );
};

export default JSX_Conditional_Rendering_Using_And_And_Operator;
```

HTTPS://T.ME/PROGRAMMING_AND_ALGORITHMS_EN
@RAMTIN_BA
RAMTINBA145822@GMAIL.COM

8. Passing Properties to Child Component:

It has a unidirectional flow, so you cannot pass components from child to parent.

Passing a String

Like html we will use Attributes:

Parent Component:

```
import { useState, React } from 'react'
import JSX_Passing_Properties_String_to_This_Child_Component from
'./components/JSX_Passing_Properties_String_to_This_Child_Component';

function App() {
  return (
    <div>
      …
    <JSX_Passing_Properties_String_to_This_Child_Component
message='This is a String from parent component'/>
    </div>
  );
};
```

Child Component:

```
import React from 'react';
function JSX_Passing_Properties_String_to_This_Child_Component(props) {
    return (
        <div>
            <h3>JSX_Passing_Properties_String_to_This_Child_Component:</h3>
            <h4>message from parent component:</h4>
            <p>{props.message}</p>
        </div>
    );
};
export default JSX_Passing_Properties_String_to_This_Child_Component;
```

Passing an Object:

Parent Component:

```
import { useState, React } from 'react'
import JSX_Passing_Properties_Object_to_This_Child_Component from
'./components/JSX_Passing_Properties_Object_to_This_Child_Component';
function App() {
  const carObject = {
    brand: "Volvo",
    countryOfOrigin: "Sweden",
    productionDate: 1927
  }
  return (
    <div>
     .. ...
    <JSX_Passing_Properties_Object_to_This_Child_Component car={carObject}/>
    </div>
  );
};
export default App;
```

HTTPS://T.ME/PROGRAMMING_AND_ALGORITHMS_EN
@RAMTIN_BA
RAMTINBA145822@GMAIL.COM

Child Component:

```
import React from 'react';
function JSX_Passing_Properties_Object_to_This_Child_Component(props) {
    return (
        <div>
            <h3>JSX_Passing_Properties_Object_to_This_Child_Component</h3>
            <h4>message from parent component:</h4>
            <ul>
                <li>Name of the brand: {props.car["brand"]} </li>
                <li>Origin country of the brand: {props.car["countryOfOrigin"]} </li>
                <li>Date of origination: {props.car["productionDate"]} </li>
            </ul>
        </div>
    );
};
export default JSX_Passing_Properties_Object_to_This_Child_Component;
```

Passing a Function:

Parent Component:

```
import { useState, React } from 'react'
import JSX_Passing_Properties_Function_to_This_Child_Component from
'./components/JSX_Passing_ Function _Object_to_This_Child_Component';

function buttonOnClick(){
  alert("You have clicked the button")
}

function App() {
  return (
    <div>
     .. ...
     <JSX_Passing_Properties_Function_to_This_Child_Component
func={buttonOnClick}/>
    </div>
  );
};
export default App;
```

HTTPS://T.ME/PROGRAMMING_AND_ALGORITHMS_EN
@RAMTIN_BA
RAMTINBA145822@GMAIL.COM

Child Component:

```
import React from 'react';

function JSX_Passing_Properties_Function_to_This_Child_Component(props) {
    return (
        <div>
            <h3>JSX_Passing_Properties_Function_to_This_Child_Component</h3>
            <h4>function from parent component:</h4>
            <button onClick={props.func} >Submit</button>
        </div>
    );
};

export default JSX_Passing_Properties_Function_to_This_Child_Component;
```

HTTPS://T.ME/PROGRAMMING_AND_ALGORITHMS_EN
@RAMTIN_BA
RAMTINBA145822@GMAIL.COM

9. Managing Click Event:

Wrong Way:

If you implement a function in this way, as illustrated in the code below, the browser will constantly keep running this function whenever the user refreshes the page instead of running it only when the button is clicked.

App Component:

```
import { useState, React } from 'react'
import ButtonComponent from './components/Managing_Click_Event';

function App() {
  return (
    <div>
     .. ...
       <ButtonComponent />

    </div>
  );
};
export default App;
```

Button function in 'Managing_Click_Event' Component:

```
import React from "react";

function ButtonComponent(){
    <button onClick={alert('button is clicked') } > Submit </button>
}

export default ButtonComponent;
```

HTTPS://T.ME/PROGRAMMING_AND_ALGORITHMS_EN
@RAMTIN_BA
RAMTINBA145822@GMAIL.COM

Correct Way By Using Arrow Function:

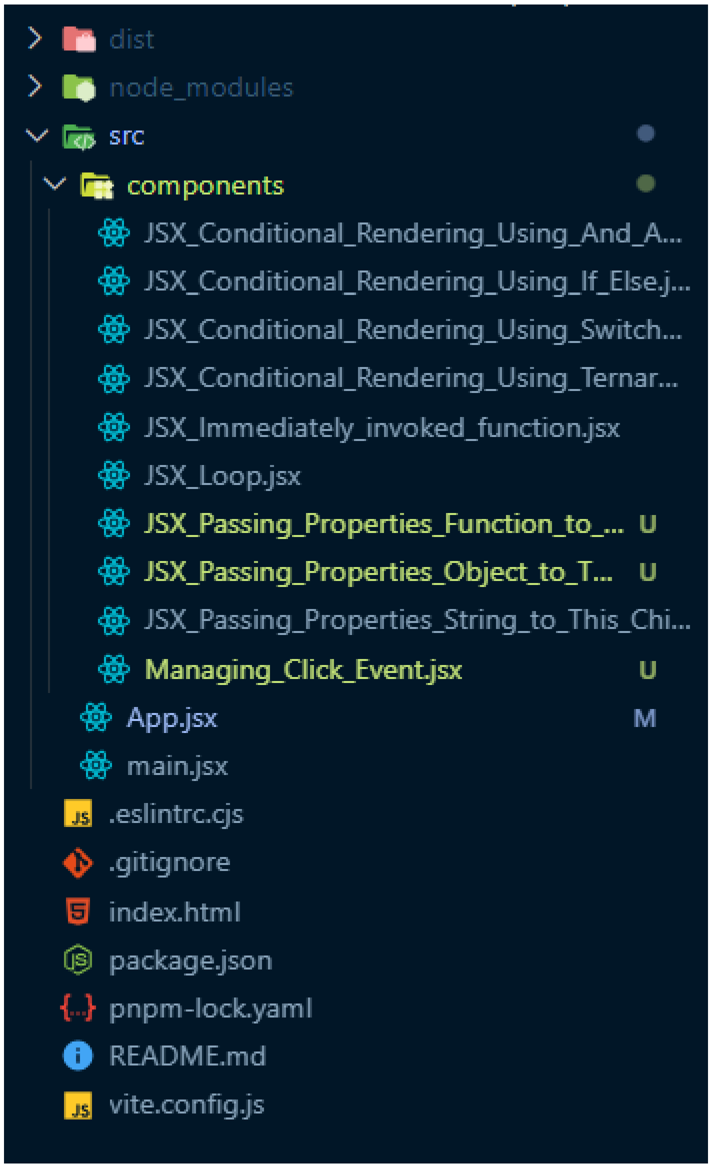
Button function in 'Managing_Click_Event' Component:

```
import React from "react";

function ButtonComponent (){
    return(
    <button onClick={()=>{alert('button is clicked')} } > Submit </button>
    )
}

export default ButtonComponent;
```

Correct Way By Using Regular Function:

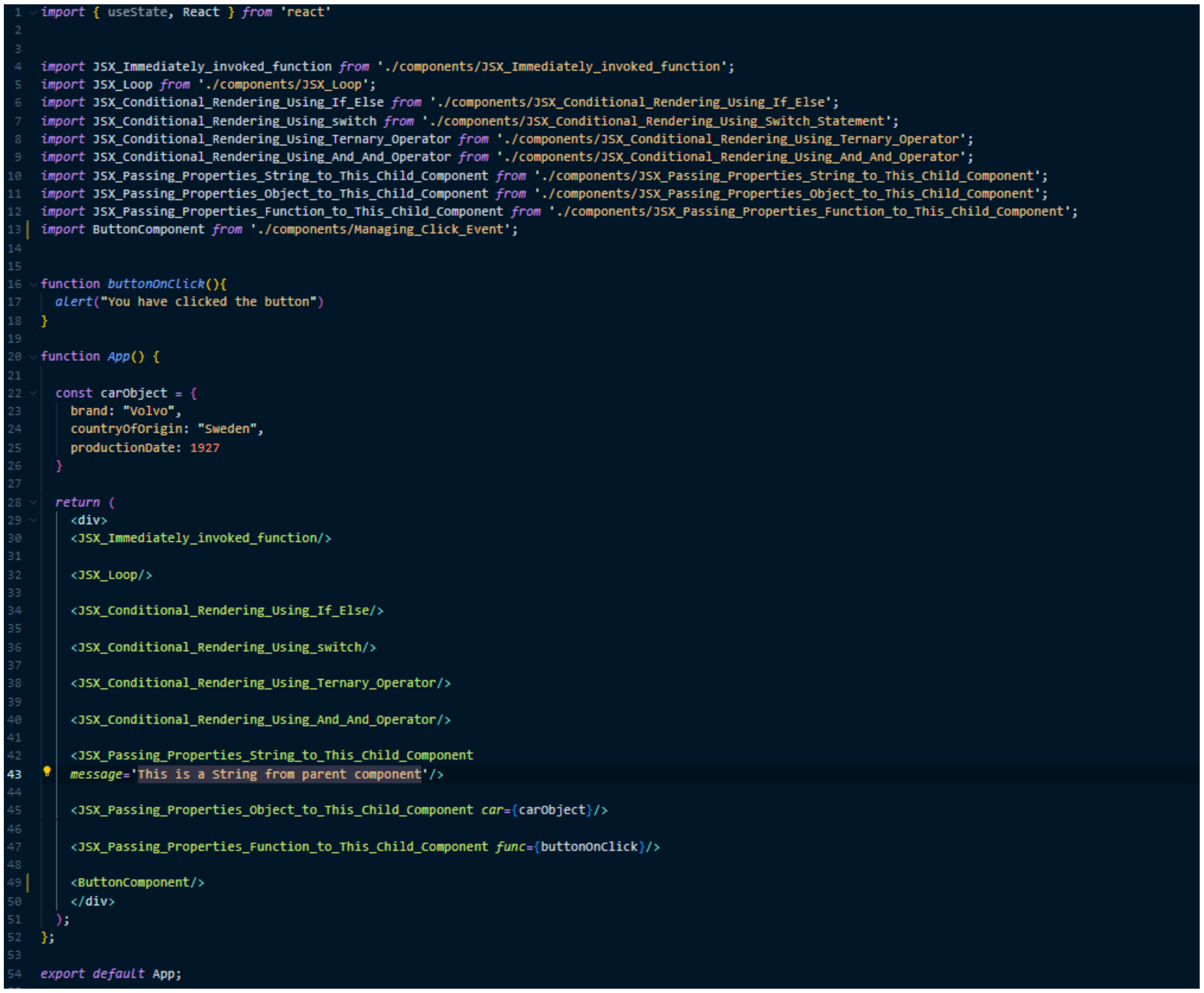
```
import React from "react";


function onClickFunction(){
    alert("you have clicked the button")
}

function FormComponent(){
    return(
    <div>
        <h3>Managing_Click_Events</h3>
        <h4>Button component for managing click events</h4>
        <button onClick={onClickFunction}> Submit </button>
    </div>
    )
}

export default FormComponent;
```

10.File structure of this chapter:



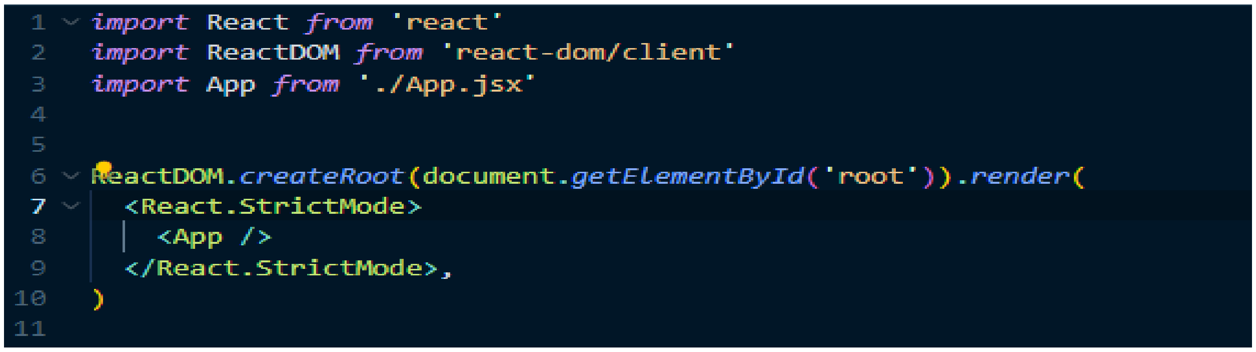
```
> dist
> node_modules
v src                                                   ●
  v components                                          ●
      JSX_Conditional_Rendering_Using_And_A...
      JSX_Conditional_Rendering_Using_If_Else.j...
      JSX_Conditional_Rendering_Using_Switch...
      JSX_Conditional_Rendering_Using_Ternar...
      JSX_Immediately_invoked_function.jsx
      JSX_Loop.jsx
      JSX_Passing_Properties_Function_to_...          U
      JSX_Passing_Properties_Object_to_T...           U
      JSX_Passing_Properties_String_to_This_Chi...
      Managing_Click_Event.jsx                        U
    App.jsx                                           M
    main.jsx
  .eslintrc.cjs
  .gitignore
  index.html
  package.json
  pnpm-lock.yaml
  README.md
  vite.config.js
```

11.App.jsx of this chapter:



```jsx
import { useState, React } from 'react'


import JSX_Immediately_invoked_function from './components/JSX_Immediately_invoked_function';
import JSX_Loop from './components/JSX_Loop';
import JSX_Conditional_Rendering_Using_If_Else from './components/JSX_Conditional_Rendering_Using_If_Else';
import JSX_Conditional_Rendering_Using_switch from './components/JSX_Conditional_Rendering_Using_Switch_Statement';
import JSX_Conditional_Rendering_Using_Ternary_Operator from './components/JSX_Conditional_Rendering_Using_Ternary_Operator';
import JSX_Conditional_Rendering_Using_And_And_Operator from './components/JSX_Conditional_Rendering_Using_And_And_Operator';
import JSX_Passing_Properties_String_to_This_Child_Component from './components/JSX_Passing_Properties_String_to_This_Child_Component';
import JSX_Passing_Properties_Object_to_This_Child_Component from './components/JSX_Passing_Properties_Object_to_This_Child_Component';
import JSX_Passing_Properties_Function_to_This_Child_Component from './components/JSX_Passing_Properties_Function_to_This_Child_Component';
import ButtonComponent from './components/Managing_Click_Event';


function buttonOnClick(){
  alert("You have clicked the button")
}

function App() {

  const carObject = {
    brand: "Volvo",
    countryOfOrigin: "Sweden",
    productionDate: 1927
  }

  return (
    <div>
    <JSX_Immediately_invoked_function/>

    <JSX_Loop/>

    <JSX_Conditional_Rendering_Using_If_Else/>

    <JSX_Conditional_Rendering_Using_switch/>

    <JSX_Conditional_Rendering_Using_Ternary_Operator/>

    <JSX_Conditional_Rendering_Using_And_And_Operator/>

    <JSX_Passing_Properties_String_to_This_Child_Component
    message='This is a String from parent component'/>

    <JSX_Passing_Properties_Object_to_This_Child_Component car={carObject}/>

    <JSX_Passing_Properties_Function_to_This_Child_Component func={buttonOnClick}/>

    <ButtonComponent/>
    </div>
  );
};

export default App;
```

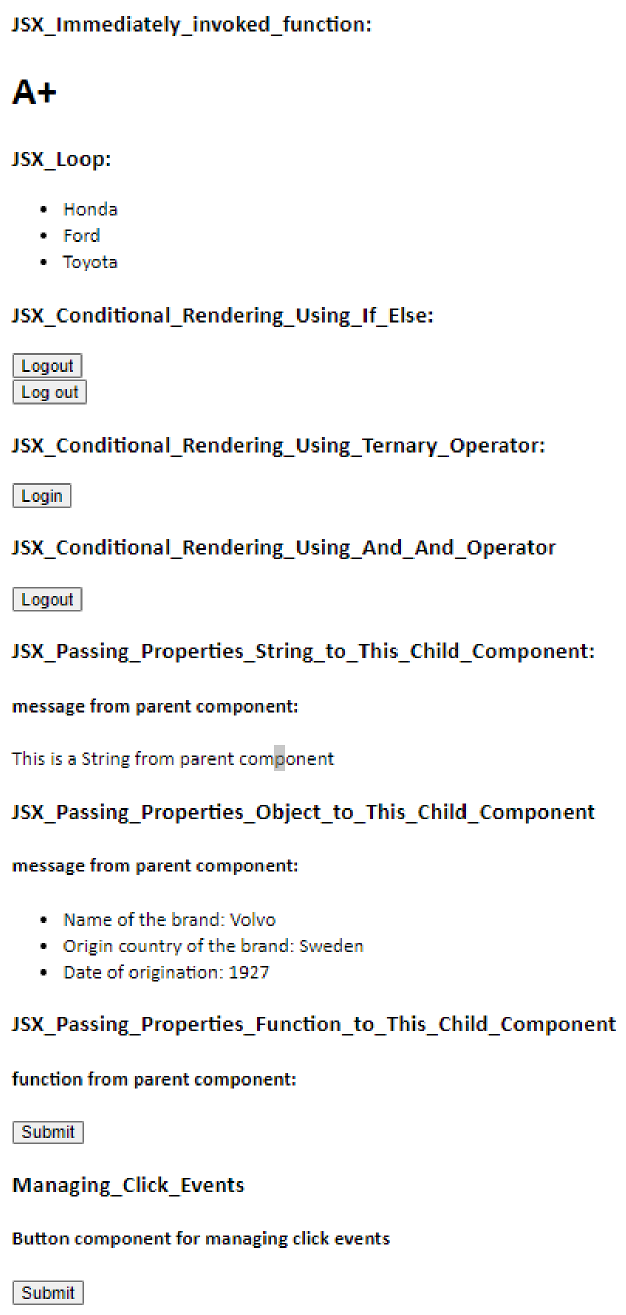
12.Main.jsx of this chapter:



```jsx
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.jsx'


ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
)

```

HTTPS://T.ME/PROGRAMMING_AND_ALGORITHMS_EN
@RAMTIN_BA
RAMTINBA145822@GMAIL.COM

13.Output Result of this chapter:



JSX_Immediately_invoked_function:

A+

JSX_Loop:

- Honda
- Ford
- Toyota

JSX_Conditional_Rendering_Using_If_Else:

Logout
Log out

JSX_Conditional_Rendering_Using_Ternary_Operator:

Login

JSX_Conditional_Rendering_Using_And_And_Operator

Logout

JSX_Passing_Properties_String_to_This_Child_Component:

message from parent component:

This is a String from parent component

JSX_Passing_Properties_Object_to_This_Child_Component

message from parent component:

- Name of the brand: Volvo
- Origin country of the brand: Sweden
- Date of origination: 1927

JSX_Passing_Properties_Function_to_This_Child_Component

function from parent component:

Submit

Managing_Click_Events

Button component for managing click events

Submit

3. React Hook and State Manager:

HTTPS://T.ME/PROGRAMMING_AND_ALGORITHMS_EN
@RAMTIN_BA
RAMTINBA145822@GMAIL.COM