React JS.

- **React JS.**
- React is a free and open-source front-end JavaScript library for building user interfaces based on components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.
- **Is React JS Library or Framework?**
- React is not a Framework. React is a JavaScript library for building user interfaces.
- It is also known as ReactJS and React.js so don't get confused if you read different notation in different places.
- React knows only one thing that is to create an awesome User interface.
- **React History.**
- React was first designed by Jorden Walke, a software engineer at Facebook.
- It was first deployed for Facebook News feed around 2011.
- In 2013 React was open sourced at JS conference.
- **React made by Jordan .**
- **About React.**
- **Component based approach.**
- A component is one of the core building blocks of React.
- In other words, we can say that every application you will develop in React will be made up of pieces called components. Components make the task of building UIs much easier.
- **Component based approach.**
- **Uses a Declarative Approach.**
- Declarative programming is a programming paradigm that expresses the logic of a computation without describing its control flow.
- Declarative programming is like asking your friend to drow a landscape. You don't care how they draw it, that's up to them.
- **Dom updates are handled gracefully.**
- Reusable code.
- React is designed for speed, speed of implementing the application simplicity and scalability.

- **Import React and importDom.**
- We need to import React in every component file because React provides the necessary functions and objects required to create and manage components.
- **Methods of render.**
- Rendering is React's process of describing a user interface based on the application's current state and props.
- Kya dikhana hai kaha dikhana hai.
- Render takes only one html and JSX elements.
- **Hello world in ReactJS.**
- import React from "react";
  import ReactDom from "react-dom";

  ReactDom.render(<h>Helloworld Shanelhai.</h>,document.getElementById('root'));
- **JSX in ReactJS.**
- **JavaScript extension:** JSX stands for JavaScript XML. It is a syntax extension to JavaScript that makes it easier to write HTML in React. JSX allows you to write HTML-like markup inside a JavaScript file. This makes it easier to create and maintain complex user interfaces.
- **Here are some of the benefits of using JSX:**
- JSX makes it easier to write HTML in React.
- JSX makes it easier to create and maintain complex user interfaces.
- JSX makes your code more organized and readable.
- **We have three solutions show the data.**
  1. ReactDom.render(<h>Hello world Shanelhai.</h>,document.getElementById('root'));
- **Bables converts the code.**
  2. ReactDom.render(("h", {

  children: "Hello world Shanelhai."

  }), document.getElementById('root'));

  3. var h1 = document.createElement("h1");
  h1.innerHTML = "Shanelhai Saifi Web Developer";
  document.getElementById("root").appendChild(h1);

- **How to Render Multiple Elements inside ReactDOM.render() in ReactJS.**
- In React v16 it’s possible for render() to return an array of elements.

-
  ```
  import React from "react";
  import ReactDom from "react-dom";
  ReactDom.render(
    <div>
      <h1>Hello world Shanelhai.</h1>

      <p>I am learing react</p>

     </div>,

     document.getElementById('root'));
  ```
- **React Fragment.**
- React Fragment is a feature in React that allows you to return multiple elements from a React component without adding extra nodes to the DOM.

-
  ```
  import React from "react";
  import ReactDom from "react-dom";
  ReactDom.render(
    <React.Fragment>
      <h1>Hello world Shanelhai.</h1>
      <p>I am learing react</p>
    </React.Fragment>,
    document.getElementById('root'));
  ```

- **Expressions in JSX in ReactJS.**
- JSX expressions are a powerful tool that can be used to create dynamic and interactive user interfaces in ReactJS. By understanding how to use JSX expressions, developers can write more efficient and maintainable code.

```
import React from 'react';
import ReactDOM  from 'react-dom';
const flname = "Shanelhai Saifi";
// var a = 5;
// var b =  2;
// var c = a + b;
ReactDOM.render(
  <>
   <h1>My name is {flname}.</h1>
   <p>My luck number is {5 + 2}.</p>
   <p>Find the random number{Math.random()}.</p>
  </>,
  document.getElementById('root'));
  // only use Expression not statements.
```

- **Template Literals in JSX in ReactJS.**
- Template literals are a feature of JavaScript that allow you to create multiline strings and embed expressions within them. They are enclosed in backticks (`) instead of double or single quotes.

```
import React from 'react';
import ReactDOM  from 'react-dom';
const fname = "Shanelhai";
const lname = "Saifi";
ReactDOM.render(
  <>
   <h1>My name is {fname +" "+lname}.</h1>
   <p>My luck number is {5 + 2}.</p>
   <p>Find the random number{Math.random()}.</p>
  </>,
  document.getElementById('root'));
  // only use Expression not statements.
```

- **JSX Attributes in ReactJS.**
- JSX attributes are used to pass data from a component to its rendered HTML element. They are written in camelCase, and can be either strings or expressions. For example, the following JSX code would render a <div> element with a class Name of "my-class" and a style attribute of { color: 'red' }:
- Here are some additional things to keep in mind when using JSX attributes:
- JSX attributes are case-sensitive.
- JSX attributes can be either strings or expressions.
- JSX attributes can be used to pass event handlers to rendered HTML elements.
- JSX attributes can also be used to pass custom data to a component.
- Custom attributes are prefixed with data-.

```
import React from 'react';
import ReactDOM from 'react-dom';
const name = "Shanelhai";
//image gellery create.
const imag1 = "https://picsum.photos/200/300";
const imag2 = "https://picsum.photos/250/300";
const imag3 = "https://picsum.photos/300/300";
const imag4 = "https://picsum.photos/400/300";
const links =
"https://www.youtube.com/watch?v=OloAP8p1k1w&list=PLwGdqU
ZWnOp3aROg4wypcRhZqJG3ajZWJ&index=14";

ReactDOM.render(
  <>
    <h1 contentEditable="true">Hello, My Name is {name}.</h1>
    <img scr={imag1} alt="randomImages"/>
    <img scr={imag2} alt="randomImages"/>
    <img scr={imag3} alt="randomImages"/>
    <a href={links} target="_thapa">
    <img scr={imag4} alt="randomImages"/>
    </a>
  </>,
  document.getElementById('root')
);
```

- **CSS Styling in React JS.**
- To style the React elements using the CSS file, we first import the CSS file and then assign the classes contained in the CSS file to the className prop of React elements. Syntax: The syntax to assign the classes to the className prop is mentioned below. Example: The content of App. js and App.
- **React JS code.**

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css'; // linked in path of CSS.
const name = "Shanelhai";
ReactDOM.render(
  <>
    <h1 className="heading">Hello, My Name is {name}.</h1>
    <div className="img_div">
      <img
src="https://i.pinimg.com/736x/cf/c9/ec/cfc9ec5a5eeee208caf285f561
2f22e1.jpg"/>
      <img
src="https://i.pinimg.com/736x/cf/c9/ec/cfc9ec5a5eeee208caf285f561
2f22e1.jpg"/>
    </div>
  </>,
  document.getElementById("root"));
```

- **CSS Code.**

```
.heading{
  color: black;
  text-align: center;
  text-transform: capitalize;
  font-weight: bold;
  text-shadow: 0px 2px 4px red;
  margin: 50px 0px;                    }
.img_div{
  display: flex;
  justify-content: center;             }
.img_div{
  width: 400px;
  height: 300px;
  margin: auto;                        }
```

- **ADD GOOGLE FONT IN REACT JS.**
- Go to the Google Fonts website.
- Choose the font you want to use.
- Click the : Select this style** button.**
- Click the : Embed** tab.**
- Copy the : CSS** code.**
- Open your React project.
- Create a new file called : fonts.css**.**
- Paste the CSS code into the : fonts.css** file.**
- Import the : fonts.css** file into your React component.**
- Use the : font-family** property to apply the Google Font to your component.**
- **HTML in head tag.**

```
<link rel="preconnect" href="https://fonts.googleapis.com" />
   <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
   <link
href="https://fonts.googleapis.com/css2?family=Josefin+Sans:ital,wght@0,100..700;1,100..700&family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap"
     rel="stylesheet"
  />
```

- **CSS Code.**

```
.heading{
  color: black;
  text-align: center;
  text-transform: capitalize;
  font-weight: bold;
  text-shadow: 0px 2px 4px red;
  margin: 50px 0px;
  font-family: "Josefin Sans", sans-serif;
}
```

- **Inline CSS Style in ReactJS.**
- To apply inline styles in ReactJS, you can use the style attribute. This attribute accepts a JavaScript object with camelCased properties. Each key-value pair in the object represents a CSS property and its value.

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css'; // linked in path of CSS.
const name = "Shanelhai";
//text-align: center; that is called kabab wala case.
const heading = {color: 'black',
textTransform : 'capitalize',
textShadow: '0px 2px 4px red',
textAlign: 'center'
}
ReactDOM.render(
  <>
    <h1 style={heading}>Hello, My Name is {name}.</h1>
    <p>Ertuğrul Ghazi from a Small Principality, He is the founder of the sovereign Ottoman Empire. Ottoman Empire 1299-1923 it ruled for 624 years between the years and it is the 4th longest-lasting empire in world history. Especially between years 1300 and 1700 It can be perceived like the USA today.</p>
  </>,
  document.getElementById("root")
);
```

- **React Components.**
- 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.

-
  ```
  import React from 'react';
  import ReactDOM  from 'react-dom';
  import Master from './Master';
  ReactDOM.render(
       <Master/>,
   document.getElementById('root'));
  ```
- **How to create Master components.**
  ```
  import React from 'react';
  import Heading from './Heading';
  import Paragraph from './Paragraph';
  import List from './List';
  function Master (){
     return(
        <>
        <Heading/>
        <Paragraph/>
        <List/>
        <Paragraph/>
        </>
     );
  }
  export default Master;
  ```
- **Create components.**

-
  ```
  import React from "react";
  function List(){
     return  (<ol>
     <li>Home</li>
     <li>About</li>
     <li>Contact</li>
     <li>Vedio</li>
   </ol>);
  }
  export default List;
  ```

- **Import Export Modules.**
- Make a new JS file to put the components in.
- Export your function component from that file (using either default or named exports).
- Import it in the file where you'll use the component (using the corresponding technique for importing default or named exports).
- **Index.js**

```
import React from 'react';
import ReactDOM  from 'react-dom';
//import.
import youtuber, {Shaan,myname} from './Master';

ReactDOM.render(
   <>
      <ol>
         <li>Shanelhai</li>
         <li>{youtuber}</li>
         <li>{Shaan}</li>
         <li>{myname()}</li>
      </ol>
   </>,
   document.getElementById('root'));
```

- const youtuber = "thapa technical";

```
//export the two and more.
const Shaan = 'Shanelhai Saifi';
//use of function.
function myname(){
   var myname = 'React JS';
   return myname;
}
export default youtuber;
export {Shaan,myname};
```

- **Props in React JS 1.**
- Props in React JS, or properties, are arguments passed to React components. These props are passed to components via HTML attributes. Props enable the component to access customized data, values, and pieces of information.
- Props are read-only, meaning that you cannot change their value inside the component. If you try to change the value of a prop, you will get an error.

```
import React from 'react';
import ReactDOM from 'react-dom';
import Card from './Card';
ReactDOM.render(
   <>
      <Card
imgsrc="https://i.pinimg.com/564x/1a/2e/92/1a2e92b39ffc0ab962634
ba92bc14c66.jpg" title="A Netflix Orignal
Series"sname="Dark"link="https://www.netflix.com/in/title/80990668
?source=35"/> repet
   </>,
   document.getElementById('root'));
```

- **Card.jsx**

```
import React from 'react';
function Card(props){
   return (
      <>
      <div className='cards'>
      <div className='card'>
         <img src={props.imgsrc}alt='myPic'className='card_img'/>
         <div className='card_info'>
            <span className='card_category'>{props.title}</span>
            <h3 className='card_title'>{props.sname}</h3>
            <a href={props.link}target="_blank">
               <button>Watch Now</button>
            </a>
         </div>
      </div>
      </div>
     </>                  );      }
export default Card;
```

- **Arrays in React JS 2 Sdata.jsx.**
- Props in React JS and Arrays work in same like JavaScript.

```
const Sdata = [    {
Id=1,
imgsrc:"https://i.pinimg.com/564x/1a/2e/92/1a2e92b39ffc0ab962634ba92bc14c66.jpg",
   title:"A Netflix Orignal Series",
   sname:"Dark",
   link:"https://www.netflix.com/in/title/80990668?source=35",
},  {
imgsrc:"https://i.pinimg.com/236x/9f/45/88/9f4588f879aa9781cbb77e93c6cef20a.jpg",
   title:"A Netflix Orignal Series",
   sname:"Extra Curriclar",
   link:"https://www.netflix.com/in/title/80990668?source=35",
},          ];
export default Sdata;
```

- **Link in React JS main files Index.js.**

```
import React from 'react';
import ReactDOM from 'react-dom';
import Card from './Card';
import './index.css';
import Sdata from './Sdata';
ReactDOM.render(
   <>
   <h1  className='heading_style'>List of top 5 Netflix Series in 2020.</h1>
      <Card  key={ val.id}
         imgsrc={Sdata[0].imgsrc}
         title={Sdata[0].title}
         sname={Sdata[0].sname}
         link={Sdata[0].link}/>
      <Card
         imgsrc={Sdata[1].imgsrc}
         title={Sdata[1].title}
         sname={Sdata[1].sname}
         link={Sdata[1].link}
   </>,
   document.getElementById('root'));
```

- **Array Map & Fat Arrow function in React JS 3.**
- **Array Map:** The Array.map() method creates a new array by calling a function for each element in an array. The function can be used to modify the elements of the array, or to create a new array with different values.
- **Arrow Function:** Arrow functions are a concise way to write function expressions. They are useful for simple actions, especially one-liners. To create an arrow function, you use the arrow (=>) symbol.
- **Index.js**

```
import React from 'react';
import ReactDOM from 'react-dom';
import Card from './Card';
import './index.css';
import Sdata from './Sdata';

ReactDOM.render(
    <>
    <h1 className='heading_style'>List of top 5 Netflix Series in 2020.</h1>
    {Sdata.map((val)=>{
        return (
        <Card
            imgsrc={val.imgsrc}
            title={val.title}
            sname={val.sname}
            link={val.link}
        />
    );
    })}
    </>,
    document.getElementById('root'));
```

- **If else React JS.**
- The if-else statement is a programming construct that allows us to control the flow of a program based on a condition. If the condition is true, the code inside the if block will be executed. If the condition is false, the code inside the else block will be executed.
- **App.jxs**

```
import React from 'react';
import Netflix from './Netflix';
import Amozon from './Amazon';
const favSeries = "amazon";
const fav = ()=>{
   if(favSeries === "netflix"){
   return <Netflix/>;
}else{
   return <Amozon/>;
}
};
const App = ()=>(
   <>
   <h1 className='heading_style'>List of top 5 Netflix Series in 2020.</h1>
   <fav />
   </>
);
export default App;
```

- **Ternary Operator in React JS.**
- The ternary operator in React JS is a conditional operator that allows you to write more concise and readable code. It is a shortcut for the if-else statement and is the only JavaScript operator that takes three operands.
- syntax.
- condition ? exprIfTrue : exprIfFalse
- Parameters.
- condition
- An expression whose value is used as a condition.
- exprIfTrue.
- An expression which is evaluated if the condition evaluates to a truthy value.
- (one which equals or can be converted to true).
- exprIfFalse.
- An expression which is executed if one condition is falsy.
- (that is, has a value which can be converted to false).
- **Index.js.**

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
ReactDOM.render(
   <App/>,
   document.getElementById('root'));
```

- **App.jsx**

```
import React from 'react';
import Netflix from './Netflix';
import Amozon from './Amazon';
const favSeries = "amazon";
const App = ()=>(
   <>
   <h1 className='heading_style'>List of top 5 Netflix Series in 2020.</h1>
    {/* <favS /> */}
    {favSeries === 'amazon' ? <Netflix/>: <Amozon/> }
   </>            );
export default App;
```

- **Hooks in React JS.**
- Hooks are the new feature introduced in the React 16.8 version.
- It allows you to use state and other React features without writing a class. Hooks are the functions which “hook into” React state and lifecycle features from function components.
- It does not work inside classes.
- Hooks should always be used at the top level of the React functions.
- Nods version 6 or above. NPM version 5.2 or above.
- **App.jsx**

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

const App =() =>{
    const state = useState();
    const [count, setcount] = useState(0);

    const IncNum =()=>{
        setcount(count + 1);
};
     return (
        <>
            <h1> {count} </h1>
            <button onClick={IncNum}>Click Me</button>
        </>
     );
}
export default App;
```

- **Events in React JS.**
- In React JS, events are actions that are triggered by the user or the system. When an event is triggered, it can cause a change in the state of the component, which can then be reflected in the UI.
- Click: This event is triggered when the user clicks on an element.
- Change: This event is triggered when the value of an input element changes.
- Submit: This event is triggered when the user submits a form.
- KeyPress: This event is triggered when the user presses a key on the keyboard.
- MouseMove: This event is triggered when the user moves the mouse over an element.
- MouseOver: This event is triggered when the user moves the mouse over an element.
- MouseOut: This event is triggered when the user moves the mouse out of an element.
- **App.js**

```
import React, { useState } from "react";
import './index.css';
const App = ()=>{
    const purple = '#8e44ed';
    const [bg, setBg] = useState(purple);
    const [names, setName] = useState("Click Me");
  const bgChange = ()=>{
    let newbg = '#34495e';
    setBg(newbg);
    setName("Ouch!! :wow 👲");              };
  const bgBack = ()=>{
    setBg(purple);
    setName("Ayyo! 😣");                       };
  return (
    <>
      <div className="AB"style={{backgroundColor:bg}}>
        <button                                          onClick={bgChange}
onDoubleClick={bgBack}>{names}</button>
      </div>
    </>                    );                      };
export default App;
```

- **React Forms.**
- Forms are one of the most essential components of any interactive web application. They allow users to interact with the application and provide input. In React, forms are typically handled using controlled components.
- **App.js**

```
import React, { useState } from 'react';
import './index.css';

const App =() =>{
    const [names, setName] = useState();
    const [fullName, setFullName] = useState();
    const inputEvent = (event)=>{
       setName(event.target.value);
    };
    const onSubmit = (event)=>{
        event.preventDefault();
        setFullName(names);
    }
   return (<>
    <div>
    <form onSubmit={onSubmit}>
        <h1>Hello {fullName}</h1>
        <input type='text' placeholder='Enter Your Name'onChange={inputEvent} value={names}/>
        <br/>
        <button type='submit'>Click</button>
    </form>
    </div>
   </>
   );
}
export default App;
```

- **Login form submit.**
- To submit a form in React, you can use the onSubmit event handler. This event handler is triggered when the user clicks the submit button on the form.
- **App.js**

```
import React, { useState } from 'react';
import './index.css';
const App =() =>{
    const [names, setName] = useState("");
    const [lastName, setLastName]= useState("");
    const [fullName, setFullName] = useState("");
    const [LastNamenew, setFullNamenew] = useState("");
    const inputEvent = (event)=>{
       setName(event.target.value);
    };
    const onSubmit = (event)=>{
        event.preventDefault();
        setFullName(names);
        setFullNamenew(lastName);
    }
    const inputEventTwo = (event)=>{
        setLastName(event.target.value);
    }
   return (<>
    <div>
    <form onSubmit={onSubmit}>
        <h1>Hello {fullName} {LastNamenew}</h1>
        <input type='text' placeholder='Enter Your Name'onChange={inputEvent} value={names}/>
        <br/>
        <input type='text' placeholder='Enter Your Last Name'onChange={inputEventTwo} value={lastName}/>
        <button type='submit'>Click</button>
    </form>
    </div>
   </>
   );
}
export default App;
```

- **Handling Complex Multiple Input Form States.**

```
import React, { useState } from 'react';
import './index.css';
const App = () => {
    const [fullName, setFullName] = useState({
        fName: "",
        lName: "",                              });
    const inputEvent = (event) => {
        const { value, name } = event.target;
        setFullName((prevValue) => {
            if (name === 'fName') {
                return {
                    fName: value,
                    lName: prevValue.lName,     };
            } else if (name === 'lName') {
                return {
                    fName: prevValue.fName,
                    lName: value,
                };      }                     });                        };
    const onSubmit = (event) => {
        event.preventDefault();
        alert("Form Submitted");                  };
    return (
        <>
            <div>
                <form onSubmit={onSubmit}>
                    <h1>Hello {fullName.fName} {fullName.lName}</h1>
                    <input
                        type='text'
                        placeholder='Enter Your Name'
                        name='fName'
                        onChange={inputEvent}
                        value={fullName.fName}/>
                    <br />
                    <button type='submit'>Click</button>
                </form>
            </div>       </>                   );};
export default App;
```

- **Handling Complex Multiple Input Form States.**

```
import React, { useState } from 'react';
import './index.css';
const App = () => {
    const [fullName, setFullName] = useState({
        fName: "",
        lName: "",
        email: "",
        phone: "",
    });
    const inputEvent = (event) => {
        const { value, name } = event.target;
        setFullName((prevValue) => ({
            ...prevValue,
            [name]: value
        }));
    };
    const onSubmit = (event) => {
        event.preventDefault();
        alert("Form Submitted");
    };
    return (
        <>
            <div>
                <form onSubmit={onSubmit}>
                    <h1>Hello {fullName.fName} {fullName.lName}</h1>
                    <p>{fullName.email}</p>
                    <p>{fullName.phone}</p>
                    <input
                        type='text'
                        placeholder='Enter Your Name'
                        name='fName'
                        onChange={inputEvent}
                        value={fullName.fName}
                    />
                    <br />
                    <input
                        type='text'
                        placeholder='Enter Your Last Name'
                        name='lName'
                        onChange={inputEvent}
                        value={fullName.lName}
                    />
                    <input
                        type='email'
                        placeholder='Enter Your Email Address'
                        name='email'
                        onChange={inputEvent}
                        value={fullName.email}
                        autoComplete='off'
                    />
                    <input
                        type='number'
                        placeholder='Enter Your Phone Number'
                        name='phone'  // Fixed name here
                        onChange={inputEvent}
                        value={fullName.phone}
                    />
                    <button type='submit'>Click</button>
                </form>
            </div>
        </>
    );
};
export default App;
```

- **Spread Operator in ReactJS.**
- The spread operator is a JavaScript expression that allows you to expand arrays and objects into separate elements. It is often used to copy arrays and objects, or to merge multiple arrays or objects into a new one.
- **Array.**
- Concatenates, Destructuring, Adding elements.
- **Objects.**
- Combining two
- **Use of Spread operator.**

```
const fullName = ["Shanelhai","Saifi"];
const biodata = [1,...fullName,"male",21];
console.log(fullName);
console.log(biodata);
```

- **The concatenate of two Array.**

```
var shooterGames = ["Call of Duty","Far Cry","Resident Evil"];
var racingGames  = ["Need For Speed","Gran Turismo","Burnout"];
var Games = [...shooterGames,...racingGames];
console.log(Games);
```

- **The destructuring of an array.**

```
var shooterGames = ["Call of Duty","Far Cry","Resident Evil"];
var [first,...remaining] = shooterGames;
console.log(first);
console.log(remaining);
```

- **The concatenate of two objects.**

```
const fullName = {
        fName: "Shanelhai",
        lName: "Saifi",
        email: "Shanelhai7@gmail.com",
        phone: 8449983380,
};
const biodata ={
    id:1,
    ...fullName,
    age:21,
    gender:"male",
};
console.log(biodata);
```

- **Material UI Icons in React JS.**
- Material UI icons are a collection of pre-designed icons that can be used in React applications. They are based on Google's Material Design guidelines and are designed to be consistent and easy to use.
- **There are two web site use icons.**
- material ui icons: https://mui.com/material-ui/material-icons/
- font awesome: https://fontawesome.com/search?q=add&o=r
- **code.**

```
import React, { useState } from 'react';
import './index.css';
// import AddIcon from '@mui/icons-material/Add';
// import AddBoxIcon from '@mui/icons-material/AddBox';
import CakeIcon from '@mui/icons-material/Cake';

const App = ()=>{
    const [num, setNum] = useState(0);
    const incNum =()=>{
        setNum(num+1)
    }
    const decNum =()=>{
        if(num > 0){
        setNum(num -1);
        }else{
            setNum(0);
        }               };
    return(
        <>
            <div className='continer'>
            <div className='continer2'>
                <h1>{num}</h1>
                <button                                 onClick={incNum}
className='btn_green'><CakeIcon/>
                </button>
                <button onClick={decNum}><CakeIcon/></button>
              //button is in build and CakeIcon is a custom
            </div>
            </div>
        </>            );     }
export default App;
```

- **Material UI Framework.**
- Material UI is a popular React UI framework for building user interfaces in react application. It provides a set of reusable and customizable UI components that follow the principles of Google's Material Design.
- **Install the package in your project directory with:**
- **with npm**
- npm install @material-ui/icons
- **with yarn**
- yarn add @material-ui/icons
- **If you are not already using Material-UI in your project, you can add it with:**
- **with npm**
- npm install @material-ui/core
- **with yarn**
- yarn add @material-ui/core
- **Add the import this links.**

```
import React, { useState } from 'react';
import './index.css';
import AddIcon from '@mui/icons-material/Add';
import Button from '@mui/material/Button';
```

- **Add the after import.**

```
<div className='continer'>
          <div className='continer2'>
             <h1>{num}</h1>
             <Button onClick={incNum}
className='btn_green'><AddIcon/>
             </Button>
             <Button onClick={decNum}><AddIcon/></Button>

          </div>
          </div>
```

- **NPM (Node Package Manager).**
- NPM (Node Package Manager) is a package manager and software registry that allows developers to find, build, and manage code packages. It's the standard package manager for Node.js and is bundled with the Node.js runtime. npm contains over 800,000 packages for various applications, from front-end and robotics to mobile apps.
  1. Npm is the world's largest Software Registry.
  2. The registry contains over 800,000 code packages.
  3. Open-source developers use npm to share software.
  4. Its full form is Node Package Manager.
- **Installing npm.**
  1. Npm is installed with Node.js.
  2. This means that you have to install Node.js to get npm installed on your computer.
  3. Download Node.js from the official Node.js Web Site **https://nodejs.org**
- **Software Package Manager.**
- Npm is a software package manager that comes with Node.js. It allows developers to find, build, and maintain code packages. Npm is a command-line interface that interacts with the repository. It is the world's largest software registry, hosting over a million packages of reusable code for Node.js.
  1. All npm packages are defined in files called package.json.ex.npm init.
  2. The content of package. Json must be written in JOSON.
  3. At least two fields must be present in the definition file: name and version.
  4. **Command Line Client**
     1. Npm includes a CLI (Command Line Client) that can be used to download and install software:
     2. Npm can manage dependencies.

- **BOOTSTRAP 4 IN REACT JS.**
- **How to install Bootstrap in React JS.**

1. Step: open your google chrome.
2. Step: search npm.
3. Step: open build amazing things / npm home.
4. Step : search bootstrap.
5. Step : Install with npm: npm install bootstrap@v5.3.3.
6. Step:import“../node_modules/boostrap/dist/css/bootstrap.min.css”; add this in app.js.
7. Step: Get started with Bootstrap open this all component here.

- **React Bootstrap Autocomplete Extension.**
- Install this Extension: IntelliSense for CSS class names in.
- **Pictures lorem picsum.**
- **How to use bootstrap in React js.**

```
import React from 'react';
import "../node_modules/bootstrap/dist/css/bootstrap.min.css";
const App = ()=>{
 return(
  <>
   <React.Fragment>
    <h1 className='text-center text-capital text-danger my-5'>Welcome to my channel.</h1>
    <div class="container text-center">
    <div class="row">
    <div class="col"><div class="card">
    <img  src="https://picsum.photos/200/301"  class="card-img-top" alt="..." height="200px"/>
    <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
    </div>
    </div></div></div>
   </div>
   </React.Fragment>
  </>          );          };
export default App;
```

- **React Context API.**
- The React Context API is a way to pass data through the component tree without having to pass props down manually at every level. This makes it easier to share data between components.
  1. Step: createContext().
  2. Step: provider.
  3. Step: consumer.
- **App.js.**

```
import React, {createContext} from 'react';
import ComA from './ComA';
const FirstName = createContext();
const LastName = createContext();
const App =()=>{
        return (
          <>
            <FirstName.Provider value={"Shanelhai"}>
            <LastName.Provider value={"Saifi"}>
            <ComA/>
            </LastName.Provider>
            </FirstName.Provider>
          </>
        );              };
export default App;
export{FirstName, LastName};
```

- **ComC.jsx**

```
import React from 'react';
import {FirstName, LastName} from './App';
const ComC =()=>{
        return (
          <>
          <FirstName.Consumer>{(fname)=>{
            return (
                <LastName.Consumer>{(lname)=>{
                  return  <h1>My name is {fname} {lname}</h1>
                }}</LastName.Consumer>
            )
          }}</FirstName.Consumer>
          </>                    );                    };
export default ComC;
```

- **Use Context Hook React.**
- The useContext hook is a way to access the shared data in a functional component without having to use the Consumer component.
- **ComB.jsx**

```
import React, { useContext } from 'react';
import ComC from './ComC';
import {FirstName, LastName} from './App';

const ComB =()=>{
  const fname = useContext(FirstName);
  const lname = useContext(LastName);
        return (
          <>
            <h1>My name is {fname} {lname}</h1>
          </>                    );                   };
export default ComB;
```

- **React UseEffect hooks.**
- By using this Hook, you tell React that your component needs to do something after render. React will remember the function you passed (we’ll refer to it as our “effect”), and call it later after performing the DOM updates. In this effect, we set the document title, but we could also perform data fetching or call some other imperative API.
- **App.js**

```
import React, { useEffect, useState } from 'react';
const App = ()=>{
  const [num, setNum]= useState(0);
  const [nums, setNums]= useState(0);
  useEffect(()=>{
    alert("i am click");
  },[num]);
  return (
    <>
      <button onClick={()=>{setNum(num + 1);}}>Click me{num}</button>
      <br/>
      <button onClick={()=>{setNums(nums + 1);
      }}>Click me{nums}</button>
    </>                );                }
export default App;
```

- **React Axios API.**
- Axios is a popular JavaScript library used for making HTTP requests from a web browser or a Node.js server. It performs various operations, such as retrieving and sending data and handling responses.
- **Install package manger:** npm install axios --force.
- **ComA.jsx**

```
import React, { useEffect, useState } from 'react';
import axios from "axios";
const ComA = ()=>{
   const [num, setNum]=useState();
   const [name, setNames]=useState();
   const [move, setMoves]=useState();
   useEffect(()=>{
      async function getData(){
         const res = await
axios.get(`https://pokeapi.co/api/v2/pokemon/${num}`);
        console.log(res.data);
        setNames(res.data.name);
        setMoves(res.data.moves.length);                  }
      getData();                                          });
      return (
         <>
         <h1>you choose <span style={{color:'red'}}> {num} value
</span></h1>
         <h1>my name is <span style={{color:'red'}}>
{name}</span></h1>
         <h1>i have <span style={{color:'red'}}> {move} moves
</span></h1>
            <select value={num} onChange={(event)=>{
               setNum(event.target.value);
            }}>
               <option value='1'>1</option>
               <option value='25'>25</option>
               <option value='3'>3</option>
               <option value='4'>4</option>
               <option value='5'>5</option>
            </select>
         </>          );                    }
export default  ComA;
```

- **React Router Dom.**
- React Router DOM is a JavaScript library that helps developers create single-page web applications (SPAs) with dynamic routing. It allows developers to create routes that map to different components, and then render those components based on the current URL. This makes it easy to create complex UIs that can change without reloading the page.
- **Install this:** npm i --save react-router-dom.
- **Index.js**

```
import React from 'react';
import ReactDom from 'react-dom';
import App from './App';
import {BrowserRouter} from 'react-router-dom';
ReactDom.render(
  <BrowserRouter>
    <App/>
  </BrowserRouter>,document.getElementById('root'));
```

- **App.js**

```
import React from 'react';
import { Route, Routes } from 'react-router-dom';
import About from './About';
import Contact from './Contect';
import Error from './Error';
const App = ()=>{
         const  Name = ()=>{
          return <h1>Hello I am a Name Page</h1>
         }
      return (
        <>
        <Routes>
          <Route exact path='/' Component={About}/>
          <Route exact path='/contact' Component={Contact}/>
          <Route path='/contact/Name' Component={Name}/>
          <Route Component={Error}/>
        </Routes>
        </>
      );
}
export default App;
```

- **React Navbar / Menu.**
- A navbar, or navigation bar, is an interactive element on a website that provides access to its main sections. It's usually located at the top or side of a website, and can contain links, buttons, or other navigation components.
- **Menu.jsx**

```
import React from 'react';
import { NavLink } from 'react-router-dom';
import './index.css';

const  Menu =()=>{
   return (
      <>
         <NavLink exact activeClassName="active_class"
to='/'>AboutUs</NavLink>
         <NavLink exact activeClassName="active_class"
to='/contact'>ContactUs</NavLink>
      </>
);
};
export default Menu;
```

- **Error.jsx**

```
import React from 'react';

const  Error =()=>{
   return <h1>OOps! Page not found!</h1>
};
export default Error;;
```

- **Render and Component Prop in React Router.**
- **Component:** A React component is a reusable piece of code that defines a section of a user interface.
- **Render:** A component with a render prop takes a function that returns a React element and calls it instead of implementing its own render logic.
- **App.js**

```
import React from 'react';
import { Route, Routes } from 'react-router-dom';
import About from './About';
import Contact from './Contect';
import Error from './Error';
import Menu from './Menu';
import Service from './Service';
import './index.css';

const App = ()=>{
        const  Name = ()=>{
         return <h1>Hello I am a Name Page</h1>
        }
      return (
        <>
        <Menu/>
        <Routes>
          <Route exact path='/' Component={()=><About name="About"/>}/>
          <Route exact path='/Service' render={()=><Service name="Service"/>}/>
          <Route exact path='/contact' Component={()=><Contact name="Contact"/>}/>
          <Route path='/contact/Name' Component={Name}/>
          <Route Component={Error}/>
        </Routes>
        </>
      );
};
export default App;
```

- **React useParams Hooks.**
- The useParams hook is a React Router hook that allows you to access dynamic parameters in the URL. The hook is based on React Router, so you need to import the following from react-router-dom:
- **App.js**

```
import React from 'react';
import { Route, Routes } from 'react-router-dom';
import About from './About';
import Contact from './Contect';
import Error from './Error';
import Menu from './Menu';
import Service from './Service';
import User from './User';
import './index.css';
const App = ()=>{
         const  Name = ()=>{
          return <h1>Hello I am a Name Page</h1>
         }
      return (
        <>
        <Menu/>
        <Routes>
          <Route exact path='/' Component={()=><About
name="About"/>}/>
          <Route exact path='/Service' Component={()=><Service
name="Service"/>}/>
          <Route exact path='/contact' Component={()=><Contact
name="Contact"/>}/>
          <Route path='/contact/Name' Component={Name}/>
          <Route path='/user/:fname/:lname' Component={User}/>
          <Route Component={Error}/>
        </Routes>
        </>                    );                  };
export default App;
```

- **UseParams.**
- useParams returns an object of key/value pairs of URL parameters use it to access match.params of the current <Route>
- **User.jsx**

```
import React from 'react';
import { useParams } from 'react-router-dom';
const  User =()=>{
   const {fname, lname} = useParams();
   return <h1>user {fname} {lname}page</h1>
};
export default User;
```

- **Menu.jsx**

```
import React from 'react';
import { NavLink } from 'react-router-dom';
const  Menu =()=>{
   return (
      <>
      <div className='box'>
          <NavLink exact activeClassName="active_class"
to='/'>AboutUs</NavLink>
          <NavLink exact activeClassName="active_class"
to='/Service'>Service</NavLink>
          <NavLink exact activeClassName="active_class"
to='/user/'>User</NavLink>
          <NavLink exact activeClassName="active_class"
to='/contact'>ContactUs</NavLink>
      </div>
      </>              );            };
export default Menu;
```

- **React use Location Hooks.**
- The useLocation hook is a React Router hook that returns the location object that represents the current URL. The location object contains information about the current URL, such as the pathname, search query, and hash.
- **User.jsx**

```
import React from 'react';
import { useLocation, useParams } from 'react-router-dom';

const  User =()=>{
    const {fname, lname} = useParams();
    const location = useLocation();
    console.log(location);

    return (
        <>
            <h1>user {fname} {lname}page</h1>
            <p>My current location is {location.pathname}</p>
            {location.pathname === `/user/shaan/saifi` ?(
                <button  onClick={()=>  alert(`you  are  awesome`)}>click
me</button>
            ):null}
        </>
    );
};
export default User;
```

- **React History Hooks.**
- React Router provides a few hooks to help with routing. You can use the useHistory hook to push the new route and pass the state object as the second argument. In the receiving component, use the useLocation hook to access the state object and retrieve the passed data.
- **User.jsx**

```
import React from 'react';
import { useLocation, useParams,  useNavigate } from 'react-router-dom';

const  User =()=>{
   const {fname, lname} = useParams();
   const location = useLocation();
   const history = useNavigate();
   console.log(history);

   return (
      <>
         <h1>user {fname} {lname}page</h1>
         <p>My current location is {location.pathname}</p>
         {location.pathname === `/user/shaan/saifi` ?(
            <button onClick={()=> history.push('/')}>click me</button>
         ):null}
      </>
   );
};
export default User;
```

- **React Live Search Filter.**
- **Search.jsx**

```
import React, { useState } from 'react';
import Sresult from './Sresult';
const Search = ()=>{
   const [img, setImg]=useState();
   const inputEvent = (event)=>{
      const data = event.target.value;
      console.log(data);
      setImg(data);
   }
   return (
      <>
      <div className='searchbar'>
         <input type='text' placeholder='search anything' value={img}
onChange={inputEvent}/>
      </div>
       {img === " " ? null : <Sresult name={img}/>}
      </>
   );
}
export default Search;
```

- **Sresult.jsx**

```
import React from 'react';
const Sresult = (props)=>{
   const                                   img                                   =
`https://source.unsplash.com/600*400/?${props.name}`;
   return (
      <>
         <div>
            <img src={img} alt='search'/>
         </div>
      </>         );              }
export default Sresult;
```

- **Create React 404 Error Page Not Found using React Router.**
- The 404 error page displays paths that don't exist on the website. So, instead of specifying the path, use an asterisk (*) to match anything. The NotFound component will render for all the URLs not specified in routes.
- **Error.jsx**

```
import React from 'react';
import { NavLink } from 'react-router-dom';

const  Error =()=>{
    return (
        <>
        <div>
            <h1>OOps! Page not found!</h1>
            <p>404 Page not found!</p>
            <NavLink to="/">Go back</NavLink>
        </div>
    </>
    );
};
export default Error;
```

- **Redirect page.**
- Redirect to another page in React JS refers to navigating to different components in the single page react app using the react-router-dom package.
- **App.js**

```
import { Route, Routes, Navigate } from 'react-router-dom';
<Route path="*" element={<Navigate to="/" />} />
```

- **How to Install and Use Bootstrap 5.**
- **Install this command =** npm i bootstrap@5.0.1
- **Install this command =** npx tailwindcss init
- **App.js**

```
import React from 'react';
import '../node_modules/bootstrap/dist/css/bootstrap.min.css';
const App = ()=>{
    return (
     <>
       <div className='container'>
        <h1 className='text-center'>Shaan</h1>
        <br/>
        <div className='text-center'>
        <div className="card" style={{width: '18rem'}}>
        <img src="https://picsum.photos/seed/picsum/200/200"
class="card-img-top" alt="..."/>
        <div class="card-body">
        <p class="card-text">Some quick example text to build on the
card title and make up the bulk of the card's content.</p>
        <a
href='https://www.youtube.com/watch?v=NvmkaX5PTDE&list=PLw
GdqUZWnOp3aROg4wypcRhZqJG3ajZWJ&index=73'
className='btn btn-outline-success'>Welcome to future</a>
        </div>
        </div>
        </div>
       </div>
     </>
    );
};
export default App;
```

- **How To Host React JS Website Live for Free using GitHub.**
- Web hosting -- also known as website hosting or webhosting -- is the process where a web hosting provider stores and maintains website files and applications on a server to make its customers' websites accessible on the internet.
- **React with SASS.**
- Sass, or Syntactically Awesome Style Sheets, is a preprocessor language that extends CSS, or Cascading Style Sheets, and is compatible with all CSS versions. Sass allows you to write CSS more efficiently and conveniently by reducing repetition, and includes features like variables, nested rules, mixins, functions, imports, and inheritance. Sass code is easier to read and understand, especially in large web projects.
- **Sass allows you to use features that don't exist in CSS, such as:**
- **Variables.**
- For example, you can define variables for colors instead of typing the HEX values repeatedly.
- **Nesting.**
- You can nest your elements inside other elements in CSS, which helps with visual hierarchy.
- **SASS install and use in project.**
  1. Step: open your project.
  2. Step: npm install –save-dev node-sass
- **React JSX (HTML) Autocomplete.**
- It is gives to suggestion.
  1. Step: open setting on VScode.
  2. Step: open setting.json file.
  3. Step: "emmet.includeLanguages":{

     "javascript":"javascriptreact" }
- **Best React JS Extension for Visual Studio Code.**
  1. Step: open your extension.
  2. Step: install ES7+ React/Redux/React-Native/JS snippets.

- **Question searches the unique value in array.**
- let a = ['1', '1', '2', '2', '3', '3', '4', '4'];
- let unique = a.filter((item, i, ar) => ar.indexOf(item) === i);
- console.log(unique);
- **simple code.**
- let a = new Set(['1', '1', '2', '2', '3', '3', '4', '4']);
- const allCatValues = [...new Set(Menu.map((curElem)=> curElem.category)),'all'];
- console.log(allCatValues);
- **Master React by Creating Dynamic Filter Photo Gallery with React & Hooks add the API details in menu card.**
- **Menu.jsx**

```
const Menu = [
    {
        id : 1,
        image : 'https://i.pinimg.com/736x/d6/5c/9d/d65c9d36d584fdffd5c03fcc9278de16.jpg',
        name :  'maggi',
        category : 'breakfast',
        price : '12',
        description : 'MAGGI has changed its tag line from “2 minute noodles” to “2 minutes for education”.',
    },
    {
        id : 2,
        image : 'https://i.pinimg.com/236x/27/88/89/278889bf329e42c5b229a084bad0d41b.jpg' ,
        name : 'Aloo pakora' ,
        category : 'evening',
        price : '20',
        description : 'Aloo pakora is an Indian snack of potato slices coated in batter and deep-fried until crispy.',
    },
]
export default Menu;
```

- **MenuItems.jsx**

```
import React from 'react';
const Menuitems = (props)=>{
    return (
        <>
            <div className='menu-item container-fluid mt-5'>
                <div className='row'>
                    <div className='col-11 mx-auto'>
                        <div className='row my-5'>
                            {
                                props.items.map((elem) => {
                                    const { id, name, image, description, price } =
elem;
                                    return (
                                        <div className='item1 col-12 col-md-6 col-
lg-6 col-xl-4 my-5'>
                                            <div className='row item-inside'>
                                                <div className='col-12 col-md-12 col-
lg-4 img-div'>
                                                    <img src={image} alt='menupic'
className='img-fluid' />
                                                </div>
                                                {/* menu items description */}
                                                <div className='col-12 col-md-12 col-
lg-8'>
                                                    <div className='main-tittle pt-4 pb-

3'>

                                                        <h1>{name}</h1>
                                                        <p>{description}</p>
                                                    </div>
                                                    <div className='menu-price'>
                                                        <div className='price-divide d-
flex justify-content-between'>
                                                            <h2>{price}</h2>
                                                            <a href=">
                                                                <button className='btn btn-
primary'>Order now</button>
                                                            </a>
                                                        </div>
                                                        <p>*Prices may vary on selected
date.</p>
                                                    </div>  </div>  </div>     </div>
                                    )            })              }
            </div>      </div>      </div>      </div>
    </>       );          }
export default Menuitems;
```

- **Gallery.jsx**

```
import React, { useState } from 'react';
import '../node_modules/bootstrap/dist/css/bootstrap.min.css';
import Menu from './menu';
import Menuitems from './Menuitems';
const allCatValues = [...new Set(Menu.map((curElem) =>
curElem.category)), 'all'];
const Gallery = () => {
  const [items, setItems] = useState(Menu);
  const [calItems, setCalItems] = useState(allCatValues);
  const filterMenu = (category) => {
    if (category === 'all') {
      setItems(Menu);
    } else {
      const updatedItems = Menu.filter((curElem) =>
curElem.category === category);
      setItems(updatedItems);
    }        };
  return (
    <>
      <h1 className='mt-5 text-center main-heading'>Order Your
Favourite Dish.</h1>
      <hr />
      <div className='menu-tabs container'>
        <div className='menu-tab d-flex justify-content-around'>
          {calItems.map((curElem, index) => (
            <button className='btn btn-warning' key={index}
onClick={() => filterMenu(curElem)}>
              {curElem}
            </button>
          ))}
        </div>
      </div>
      <Menuitems items={items} />
    </>        );         };
export default Gallery;
```

- **Navbar with using hamburger.**
- **Npm install react-icons --save**
- **Navbar.jsx**

```
 import React, { useState } from 'react';
import './index.css';
import { FaFacebookSquare, FaInstagramSquare, FaYoutube } from "react-icons/fa";
import { GiHamburgerMenu } from "react-icons/gi";

const Navbar = () => {
    const [showMediaIcons, setShowMediaIcons] = useState(false);
    return (
        <>
            <nav className='main-nav'>
            {/* 1 is logo part */}
            <div className='logo'>
                <h1>S<span>hanelhai</span>
                <span> S</span>aifi</h1>
            </div>
            {/* 2 is menu part */}
            <div className={showMediaIcons ? "menu-link mobile-menu-link" : "menu-link"}>
                <ul>
                    <li>
                        <a href='#'>Home</a>
                    </li>
                    <li>
                        <a href='#'>About</a>
                    </li>
                    <li>
                        <a href='#'>Services</a>
                    </li>
                    <li>
                        <a href='#'>Contact</a>
                    </li>
                </ul>
            </div>
            {/* 3 social media link */}
            <div className='social-media'>
        <ul className='social-media-desktop'>
          <li>
            <a
href="https://www.youtube.com/watch?v=8AJ3Kcz5FsM&list=PLwG
dqUZWnOp3aROg4wypcRhZqJG3ajZWJ&index=86"
target='_Shaan'><FaYoutube className='Youtube'/></a>
          </li>
          <li>
            <a
href="https://www.youtube.com/watch?v=8AJ3Kcz5FsM&list=PLwG
dqUZWnOp3aROg4wypcRhZqJG3ajZWJ&index=86"
target='_Shaan'><FaInstagramSquare className='Instagram'/></a>
          </li>
          <li>
            <a
href="https://www.youtube.com/watch?v=8AJ3Kcz5FsM&list=PLwG
dqUZWnOp3aROg4wypcRhZqJG3ajZWJ&index=86"
target='_Shaan'><FaFacebookSquare className='Facebook'/></a>
          </li>
        </ul>
        {/* 4 hamburger-menu */}
        <div className='hamburger-menu'>
        <a href='#' onClick={()=>
setShowMediaIcons(!showMediaIcons)}><GiHamburgerMenu/></a>
        </div>
      </div>
      </nav>
      {/* 5 hero section */}
      <section className='hero-section'>
        <p>Welcome to</p>
        <h1>Shanelhai</h1>
      </section>
    </>
  );
};

export default Navbar;
```

- **Navber index.css**

```
@import
url("https://fonts.googleapis.com/css2?family=Jost:ital,wght@0,100;0,
200;0,300;0,400;0,500;0,600;0,700;1,100&display=swap");
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Jost", sans-serif;
}
html {
  font-size: 62.5%;
}
a {
  text-decoration: none;
}
li {
  list-style: none;
}
/* /* navbar style start  */
.main-nav {
  width: 100%;
  height: 10rem;
  display: grid;
  grid-template-columns: 10rem 1fr 2fr 1fr 10rem;
  box-shadow: rgba(50, 50, 93, 0.25) 0px 50px 100px -20px,
    rgba(0, 0, 0, 0.3) 0px 30px 60px -30px;
}
.logo {
  display: grid;
  /* background-color: #3b5998; */
  grid-column: 2/3;
  justify-content: start;
  align-items: center;
}
```

```
.menu-link {
  grid-column: 3/4;
}
.menu-link ul {
  height: 10rem;
  display: flex;
  justify-content: space-around;
  align-items: center;
}
.social-media {
  grid-column: 4/5;
}
.social-media ul {
  height: 10rem;
  display: grid;
  grid-template-columns: 3fr repeat(3, 1fr);
  align-items: center;
  justify-content: flex-end;
}
.social-media ul li {
  text-align: right;
}
.social-media ul li:first-child {
  grid-column: 2/3;
}
/* ----------- Grid part ends ----------------- */
.logo h2 {
  font-size: 2.5rem;
  font-weight: 400;
  text-transform: uppercase;
  background: -webkit-linear-gradient(#eee, rgb(69, 57, 248));
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
}
.logo h2 span {
  font-size: 3.5rem;
}
```

```css
.menu-link ul li {
  font-size: 1.8rem;
}
.menu-link ul li a {
  text-transform: capitalize;
  color: rgba(0, 0, 3, 0.8);
  transition: 0.5s;
}
.menu-link ul li:hover > a {
  transform-origin: left;
  color: rgba(0, 0, 3, 1);
  text-decoration: underline;
  transition: 0.5s;
}
.social-media ul li {
  font-size: 1.8rem;
}
.social-media .hamburger-menu {
  display: none;
}
.facebook {
  color: #3b5998;
}
.instagram {
  color: #c32aa3;
}
.youtube {
  color: #ff0000;
}
/* hero section   */
.hero-section {
  height: 80vh;
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: column;
}

.hero-section p {
  font-size: 3rem;
  text-transform: capitalize;
}
```

```
.hero-section h1 {
  font-size: 5rem;
  text-transform: uppercase;
  text-align: center;
}
/* responsive css style  */
@media (max-width: 1080px) {
  .main-nav {
    height: 8rem;
    grid-template-columns: 2rem 3fr 3fr 1fr 2rem;
  }
  .logo,
  .menu-link ul,
  .social-media ul {
    height: 8rem;
  }
}
/* responsive css style  */
@media (max-width: 998px) {
  .main-nav {
    height: 7rem;
    grid-template-columns: 2rem 2fr 3fr 2rem 2rem;
  }
  .menu-link {
    display: none;
  }
  .logo,
  .social-media ul {
    height: 7rem;
  }
  .mobile-menu-link {
    grid-column: 2/4;
    position: relative;
    z-index: 99;
  }
  .mobile-menu-link {
    background-color: white;
    height: 20rem;
    display: grid;
    grid-column: 2/5;
    align-items: center;
    padding-left: 3rem;
    transition: all 2s linear;
    transform-origin: top;
    box-shadow: rgba(50, 50, 93, 0.25) 0px 50px 100px -20px,
      rgba(0, 0, 0, 0.3) 0px 30px 60px -30px;
  }
  .mobile-menu-link ul {
    height: 20rem;
    display: flex;
    justify-content: space-around;
    flex-direction: column;
    align-items: start;
  }
  .mobile-menu-link ul li:first-child {
    transition-delay: 0.2s;
  }
  .social-media {
    grid-row: 1/2;
    grid-column: 3/5;
    justify-items: end;
    align-items: center;
    transition: all 2s linear;
  }
  .social-media .social-media-desktop {
    height: 0;
    display: none;
  }
  .social-media {
    height: 7rem;
    display: flex;
    justify-self: end;
    align-items: center;
  }
  .social-media .hamburger-menu {
    display: block;
    font-size: 2.5rem;
  }
}
@media (max-width: 798px) {
  .main-nav {
    height: 6rem;
    grid-template-columns: 1rem 2fr 1fr 1fr 1rem;
  }
  .logo,
  .social-media ul {
    height: 6rem;
  }
  .social-media {
    height: 6rem;
    display: flex;
    justify-self: end;
    align-items: center;
  }
  .social-media .hamburger-menu {
    display: block;
    font-size: 2.5rem;
  }
}
@media (max-width: 520px) {
  .main-nav {
    height: 6rem;
    grid-template-columns: 1rem 4fr 1fr 1fr 1rem;
  }
  .logo,
  .social-media ul {
    height: 6rem;
  }
  .logo h2 {
    font-size: 2rem;
  }
  .social-media {
    height: 6rem;
    display: flex;
    justify-self: end;
    align-items: center;
  }
  .social-media .hamburger-menu {
    display: block;
    font-size: 2.5rem;
  }
  /* hero section  */
  .hero-section h1 {
    font-size: 3.8rem;
  }
}
```