DYNAMIC WEB PAGE DESIGNING USING REACT JS

#### Summer Internship Report Submitted by

**Name: Shubham UmeshKumar Lad Enroll No: 200170107013**

**COMPUTER ENGINEERING DEPARTMENT VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE CHANDKHEDA**



**External Guide: Internal Guide:**

**Prof. Chintan Nagrecha Prof. J M Ramavat**

**Director Associate Professor**

**Infolabz It Services Pvt Ltd VGEC, Chandkheda**

## Gujarat Technological University

**Academic Year 2023-24**

## Vishwakarma Government Engineering College, chandkheda

**Computer Department CERTIFICATE**



#### Date:

This is to certify that the Summer Internship Report entitled name of seminar/miniproject/internship Submitted by EnrollNo: 200170107013

#### Name: Shubham UmeshKumar Lad towards the fulfillment of Subject: Summer Internship (3170001) of Gujarat Technological University is the record of work carried out by him under our supervision and guidance in the Academic Year 2023-24.

Internal Guide Head of Department

Prof. J M Ramavat Prof M. T. Savaliya

Associate Professor Associate Professor

VGEC Chandkheda VGEC Chandkheda

# DECLARATION

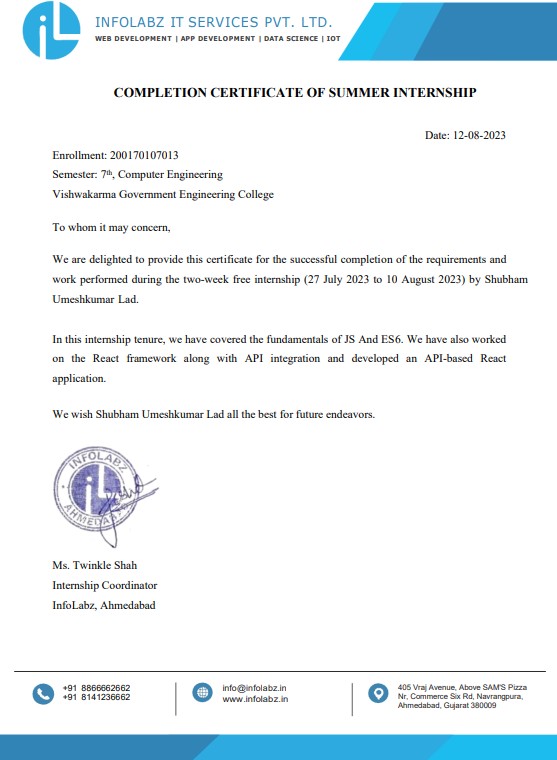
I hereby declare that I have completed my fifteen days summer internship at Infolabz It Services Pvt Ltd from 27 July 2023 to 10 August 2023 under the guidance of Chintan Nagrecha. I have declare that I have worked with full dedication during these fifteen days of training and my learning outcomes fulfill the requirements of training for the 7th Sem of degree of Computer Engineering (B.E.), Vishwakarma Government Engineering College, Ahmedabad.

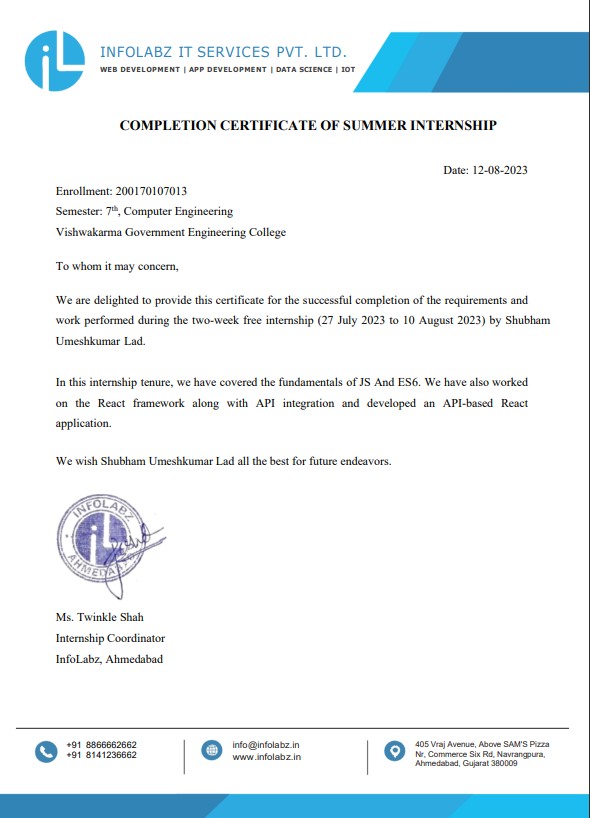
I hereby warrant that the work I have presented does not breach any existing copyright acts.

Date: SHUBHAM LAD

(200170107003)

# CONFIRMATION LETTER



**CERTIFICATE OF COMPLETION**

# ACKNOWLEDGEMENT

I wish to express my sincere gratitude to our External guide **Prof. Chintan Nagracha** for Continuously guiding me at the company and answering all my doubts with patience. I would also like to thank my Internal Guide **Prof. Jalpa Ramavat** for helping me through my internship by giving me the necessary suggestions and advice along with their valuable co-ordination in completing this internship.

I express sincere and heartfelt thanks to **Prof. Dr. Nilay Bhuptani** (Principal, Vishwakarma Government Engineering College, Ahmedabad). I would also like to thank to our HOD **Prof. Mansukh Savaliya**(Head of Department, Vishwakarma Government Engineering College, Ahmedabad).

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the company personals who gave me the permission to use and experience the valuable resources required forth internship.

Also, I appreciate the guidance given by the developer at INFOLABZ, **Prof. Chintan Nagracha** as well as the panels especially for the internship that has advised me and gave guidance at every moment of the internship. Thus, inconclusion to the above said, I once again thank the staff members of INFOLABZ IT SERVICES PVT.LTD. for their valuable support in completion of the project.

Thank you all.

# Abstract

Dynamic web page designing using ReactJS involves building interactive and responsive web interfaces by leveraging React's component-based structure and state management. This approach allows developers to create modular UI elements, manage real-time data updates, and enhance user experiences through seamless navigation and efficient rendering. This paper underscores the importance of dynamic web page designing in modern web development and how ReactJS serves as a robust solution for creating interactive and user-centric web applications. By leveraging React's component-based architecture, state management capabilities, and seamless routing, developers can craft dynamic web pages that elevate user experiences to new heights. As the web continues to evolve, mastering the art of dynamic web page designing using ReactJS remains a valuable skill for developers aiming to create cutting-edge web applications.

# Company Profile

Established in 2016, incorporation with our parent IT company, INFOLABZ IT SERVICES PVT. LTD. has managed to make it's own position in IT Sector. We are involved in Web Development, App Development, Progressive Web Application Development, IOT solutions, Graphics & Designing, Digital Marketing, Domain & Hosting services, SMS services etc.

In the span of seven years, we have managed to deliver all projects on time with utmost accuracy to our clients across the globe. We have dedicated teams of experienced and hard-working developers. Our developers who are always willing to take new challenges and looking forward to learn new things, are heart of this company.

Our objective is to sustain with exponential growth in IT industry. Our mission is to deliver the best with top notch quality every quarter and vision is to develop a product with one of its kind concept which could be used by millions of people.

# DYNAMIC WEB PAGE DESIGNING USING REACT JS

Personal Details

Name : Shubham UmeshKumar Lad Enrollment No. : 200170107013

College Name : Vishwakarma Government Engineering College (AIET)Degree : B.E. (Computer Engineering)

Semester : 7th

Company Details

Company Name : INFOLABZ IT SERVICES PVT. LTD

External Guide : Chintan Nagrecha Training Duration : 27-07-2023 to 10-08-2023

# TABLE OF CONTENT

|  |  |  |
| --- | --- | --- |
| **WEEK 1** | **01AUGUST 2023** | 18 |
|  | - Form to table | 18 |
|  | - JS Arrays | 20 |
|  |  |  |
| **WEEK 1** | **02 AUGUST 2023** | **22** |
|  | - Task Assignment | 22 |
|  | - Covid Data Search | 22 |
|  | - Mutual Fund API | 26 |
|  |  |  |
| **WEEK 2** | **03 AUGUST 2023** | **28** |
|  | - React Environment setup ( Node JS installation ) | 28 |
|  | - First React App | 28 |
|  | - Functional Components | 28 |
|  | - Class Components |  |
|  |  |  |
| **WEEK 2** | **04 AUGUST 2023** | **31** |
|  | - Variable Data Map | 31 |
|  | - Object Map | 34 |
|  |  |  |
| **WEEK 2** | **07 AUGUST 2023** | **37** |

|  |  |  |
| --- | --- | --- |
|  | - React Props | 37 |
|  | - React Bootstrap | 39 |
|  |  |  |
| **WEEK 2** | **08 AUGUST 2023** | **42** |
|  | - React Hooks : Use Effect and Use State | 42 |
|  | - API data fetch with react | 43 |
|  |  |  |
| **WEEK 2** | **09 AUGUST 2023** | **44** |
|  | - ASSIGNMENT TASK | 44 |
|  | - API BASED REAL TIME NEWS WEB APPLICATION | 44 |
|  |  |  |
| **WEEK 2** | **10 AUGUST 2023** | **47** |
|  | - Project & conclusion | 47 |

**DAY-1**

**INTRODUCTION TO JS :**

**JavaScript** is a lightweight, cross-platform, single-threaded, and interpreted compiled programming language. It is also known as the scripting language for webpages. It is well-known for the development of web pages, and many non-browser environments also use it.

JavaScript is a weakly typed language (dynamically typed). JavaScript can be used for Client-Side developments as well as Server-Side developments. JavaScript is both an imperative and declarative type of language. JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.

* Client-side: It supplies objects to control a browser and its Document Object Model (DOM) Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation. Useful libraries for the client side are AngularJS, ReactJS, VueJS, and so many others.
* Server-side: It supplies objects relevant to running JavaScript on a server. For if the serverside extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is node.js.
* Imperative language – In this type of language we are mostly concerned about how it is to be done. It simply controls the flow of computation. The procedural programming approach, object, oriented approach comes under this as async await we are thinking about what is to be done further after the async call.

**INTRODUCTION TO ES6 :**

ES6 or ECMAScript 6 is a scripting language specification which is standardized by ECMAScript International. This specification governs some languages such as JavaScript, ActionScript, and Jscript. ECMAScript is generally used for client-side scripting, and it is also used for writing server applications and services by using Node.js.

ES6 allows you to write the code in such a way that makes your code more modern and readable. By using ES6 features, we write less and do more, so the term 'Write less, do more' suits ES6.ES6 is an acronym of ECMAScript 6 and also known as ECMAScript 2015.

**OBJECTS :**

JavaScript objects are data structures that store key-value pairs, representing real-world entities or concepts in code, allowing for structured and organized data storage and manipulation. JavaScript objects are data structures that store key-value pairs, representing real-world entities or concepts in code, allowing for structured and organized data storage and manipulation. Objects can be created using object literals or constructed using constructor functions. They enable you to model real-world entities, abstract concepts, or complex data structures in your code.

**Example:**

<!DOCTYPE html>

<head>

</head>

<body>

<script> var mydata = {}; document.write(mydata);

document.write("<br>");

var mydata1 ={"gujarat":"ahmedabad"}; document.write(mydata1["gujarat"]+"<br>");

var mydata2={"ahmedabad":100,"surat":150,"vadodara":75}; document.write(mydata2["surat"]+"<br>");

var mydata3={"ahmedabad":100,"surat":[150,200,1],"vadodara":75}; document.write(mydata3["surat"][1]+"<br>");

var mydata4={"ahmedabad":[{"date":"26 july 2023","cases":210},

{"date":"27 july 2023","cases":220},

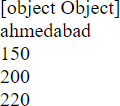
{"date":"28 july 2023","cases":230}], "surat":150,"vadodara":75} document.write(mydata4["ahmedabad"][1]["cases"]+"<br>");

</script>

</body>

</html>

Output **:**



**Example:**

<!DOCTYPE html>

<head>

</head>

<body> <script> var myobject={"gujarat":"ahmedabad","maharashtra":"mumbai","rajasthan":"jaipur"}; for(var i in myobject){ document.write(i+"<br>");

}

//print values for(var i in myobject){ document.write(myobject[i]+"<br>");

}

//print both for(var i in myobject){ document.write("state is :"+i+" city is :"+myobject[i]+"<br>");

}

var cases={"ahmedabad":[1500,1600,2],"rajkot":[500,600,1],"vapi":[100,50,0]}; for(var i in cases){ document.write(i +"new cases :"+cases[i][0]+" recovered cases: "+cases[i][1]+" death cases: "+cases[i][2]+"<br>");

}

// display all data in table

</script>

<table border="1" width="100%">

<tr>

<th>city name</th>

<th>new cases</th>

<th>recoverded cases</th>

<th>death cases</th>

</tr> <script> for(var i in cases){ document.write("<tr><td>"+i+"</td><td>"+cases[i][0]+"</td><td>"+cases[i][1]+"</td><td>"+cases[i][2]+"</ td></tr>");

}

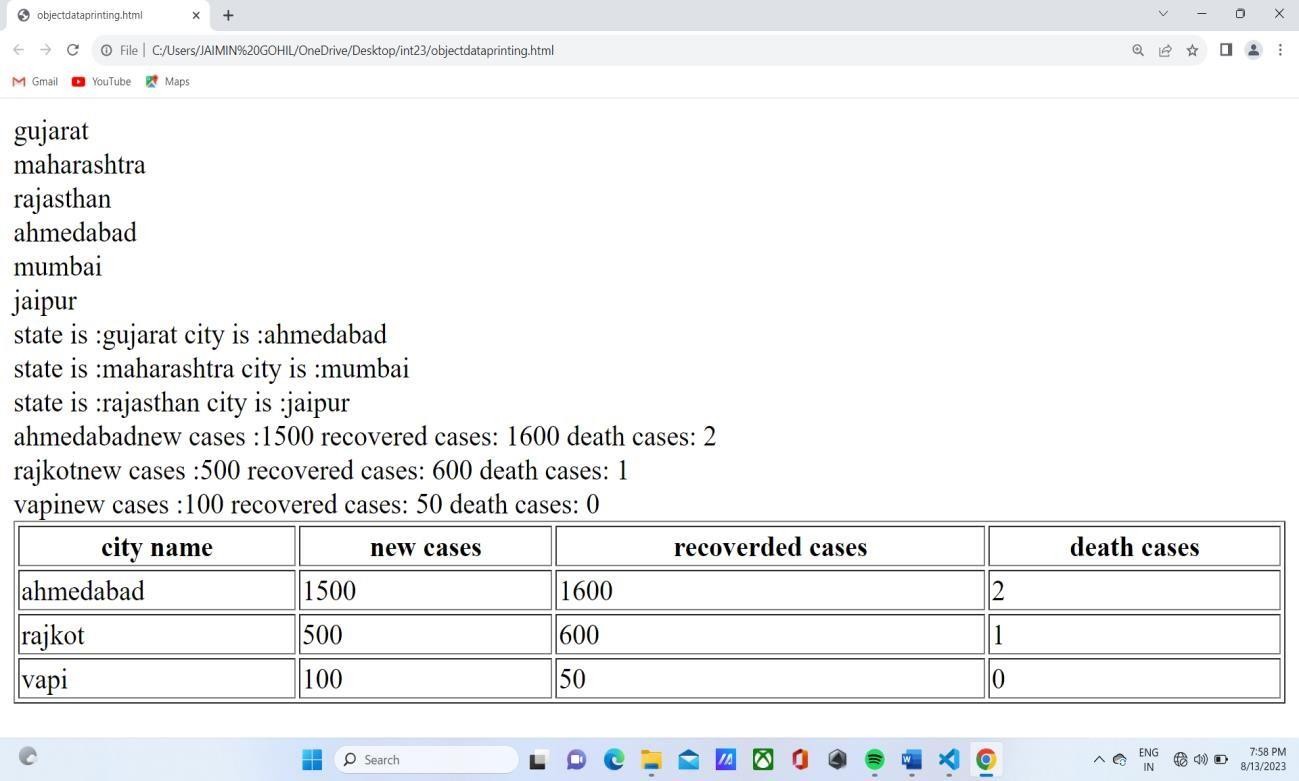
</script>

</table>

</body>

</html>

Output :



**DAY-2**

**JS Functions & Arrow Functions:**

Dynamic web page designing using React.js involves building interactive and responsive user interfaces by composing reusable components. React's virtual DOM efficiently updates and renders changes, enhancing user experience by minimizing page reloads, making it a popular choice for modern web development. Dynamic web page designing with React.js entails creating responsive and interactive user interfaces through modular components. React's virtual DOM optimizes updates, reducing page reloads for smoother user experiences, making it a leading framework for contemporary web development.

Developers can efficiently manage state, handle events, and integrate with APIs to build feature-rich web applications.

**Example:**

<!DOCTYPE html>

<head>

</head>

<body>

<script>

//function with no argument and no return value function one(){ document.write("function one called<br>");

}

one();

//function with argument and no return value function two(a,b){ document.write("addition of "+a+" and "+b+" is "+(a+b)+"<br>");

} two(100,100);

//function with no argument but it returna a value function three(){ return"hello<br>";

} var x=three(); document.write(x);

//function with argument and return value function four(i,j){ return"subtration of "+i+" and "+j+" is "+(i-j);

}

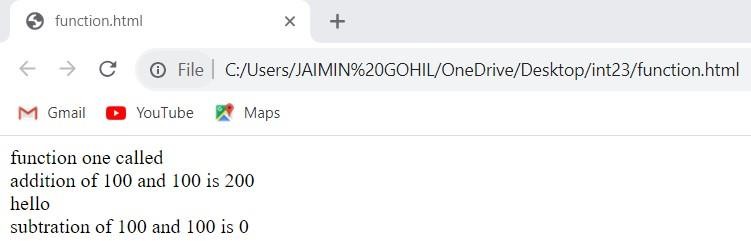
var y=four(100,100); document.write(y);

</script>

</body>

</html>

**Output :**



**Example :**

<!DOCTYPE html>

<head>

</head>

<body>

<script>

//arrowfuncwith no argument and no return value var one=()=>{ document.write("function one called<br>");

}

one();

//arrowfunc with argument but no return value var two=(a,b)=>{ document.write("addition of "+a+" and "+b+" is "+(a+b)+"<br>");

} two(50,10);

//arrowfunc with no argument but it return a value var three=()=>{ return("hello<br>");

}

var x=three(); document.write(x);

//arrowfunc with argument and return value both var four=(i,j)=>{ return("subtration of "+i+" and "+j+" is "+(i-j)+"<br>");

}

var y=four(100,100); document.write(y);

var eo=(no)=>{ if(no%2==0){ document.write("number is even<br>");

}

else{ document.write("number is odd<br>");

}

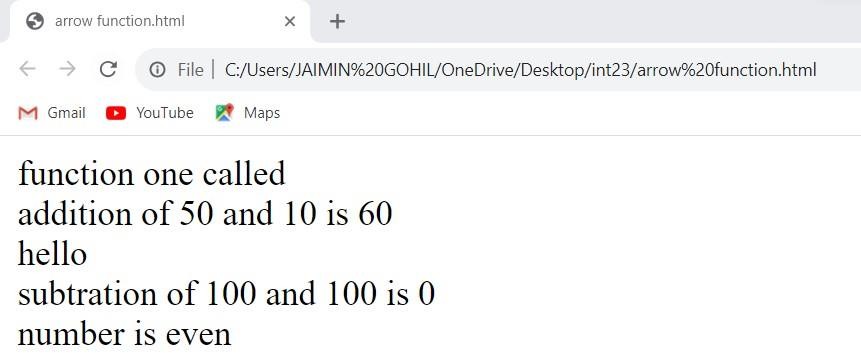
}

eo(400); </script>

</body>

</html>

**Output :**



**Async Function:**

Async functions in JavaScript simplify asynchronous code by allowing developers to write and manage asynchronous operations in a more sequential and readable manner. They use the `async` keyword before the function declaration and allow the use of the `await` keyword inside the function to pause execution until a promise is resolved. This makes working with promises and asynchronous tasks more intuitive and easier to understand.

**Bitcoin API :**

**Example :**

<!DOCTYPE html>

<head>

</head>

<body> <script> async function load(){ let url="https://api.coindesk.com/v1/bpi/currentprice.json"; let myobject=await (await fetch(url)).json(); document.write(myobject["bpi"]["USD"]["rate"]);

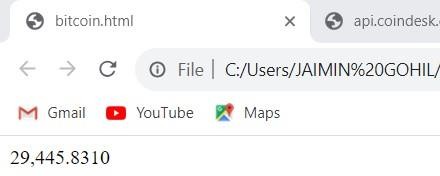
}

load(); </script>

</body>

</html>

**Output :**



# DAY-3

**Covid API – Data fetching and mapping :**

Example **:**

<!DOCTYPE html>

<head>

</head>

<body> <script> async function load(){ let url="https://data.covid19india.org/data.json"; let myobject=await (await fetch(url)).json(); document.write(myobject["cases\_time\_series"][0]["date"]+"<br>"); document.write("total number of cases: "+myobject["cases\_time\_series"].length); document.write("<table border='1' width='100%'>"); for(var i=0;i<myobject["cases\_time\_series"].length;i++){ if(myobject["cases\_time\_series"][i]["dailyconfirmed"]>=100000){ document.write("<tr style='color:red'><td>"+myobject["cases\_time\_series"][i]["date"]+

"</td><td>"+myobject["cases\_time\_series"][i]["dailyconfirmed"]+ "</td><td>"+myobject["cases\_time\_series"][i]["dailydeceased"]+

"</td><td>"+myobject["cases\_time\_series"][i]["dailyrecovered"]+"</td></tr>");

}

else{ document.write("<tr><td>"+myobject["cases\_time\_series"][i]["date"]+ "</td><td>"+myobject["cases\_time\_series"][i]["dailyconfirmed"]+

"</td><td>"+myobject["cases\_time\_series"][i]["dailydeceased"]+ "</td><td>"+myobject["cases\_time\_series"][i]["dailyrecovered"]+"</td></tr>");

}

}

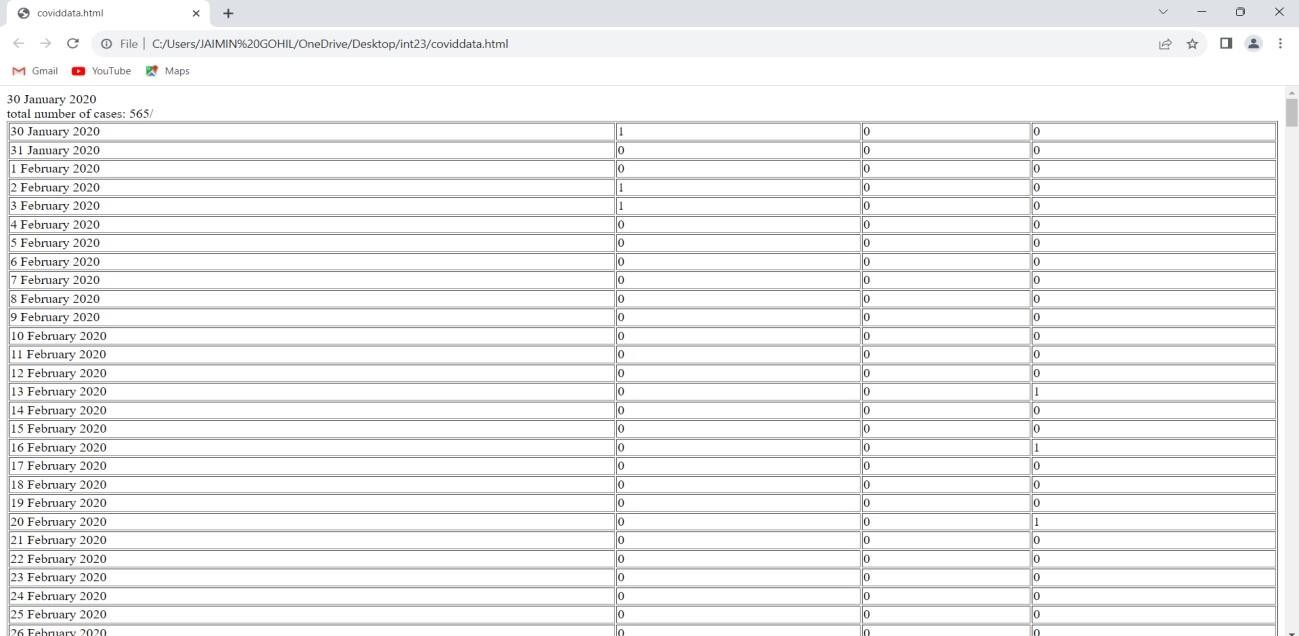
document.write("/<table>");

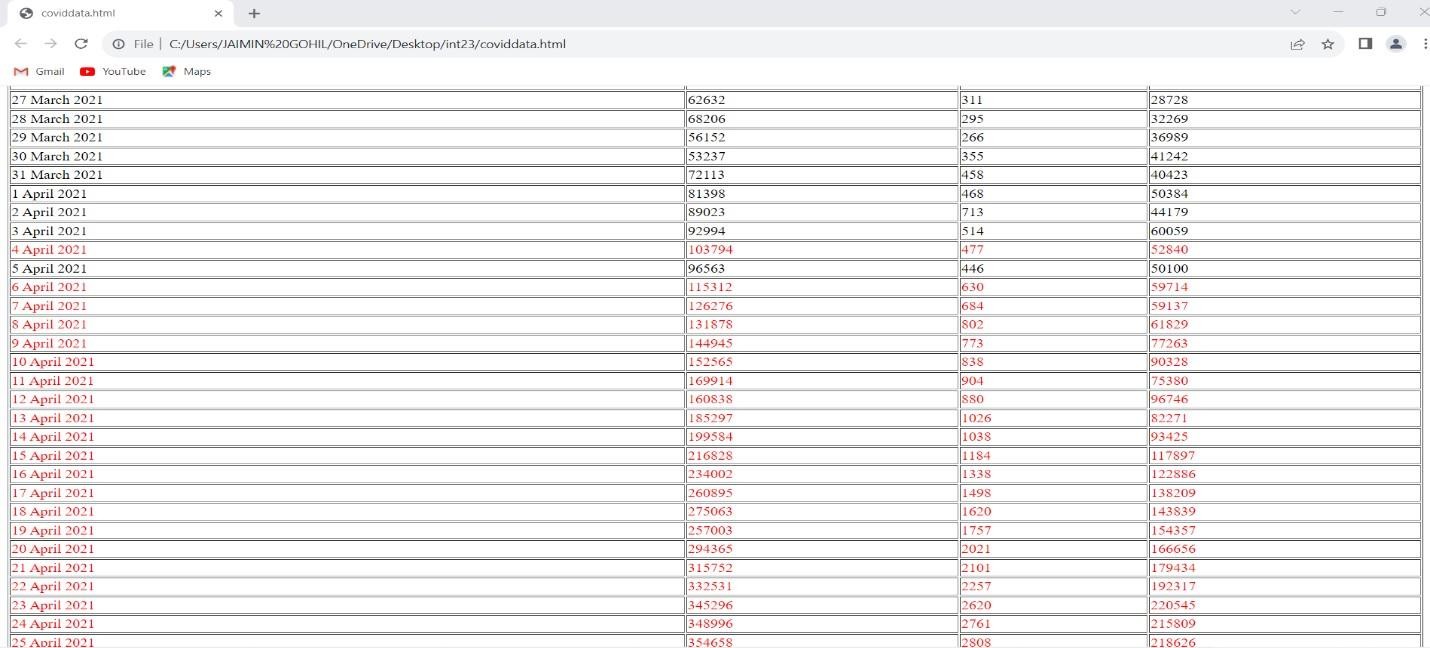
}

load(); </script>

</body> </html>

**Output :**





**USER INPUT 1 : JS Pop Up Boxes**

### Example**:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>formtovariable</title>

</head>

<body>

<script> function check(form){ var name=form.uname.value; var surname=document.getElementById("surname").value; alert("Username is "+name+" and Surname is "+surname) document.getElementById("msg").innerHTML = "Username is "+name+" and Surname is "+surname;

}

</script>

<form action="">

Name :<input type="text" class="uname" name="uname"><br><br> Surname: <input type="text" class="surname" id="surname"><br>

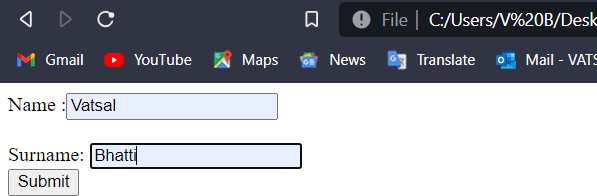
<input type="submit" onclick="check(this.form); return false;" class="submit" name="submit">

</form>

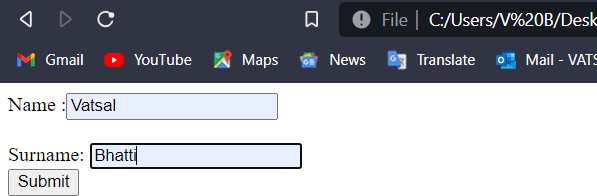
<p id="msg"></p>

</body> </html>

**Output :**



**USER INPUT 2 : Form to Variable**



# DAY-4

**Form to table :**

Form collects name and age inputs. Upon clicking the "Add" button, the addToTable function is triggered. This function extracts the input values, creates new table cells, appends them to a new table row, and adds that row to the table's body. This effectively converts form inputs into a table row.

Example**:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>formtovariable</title>

</head>

<body> <script> function showdata(form) { var name = form.uname.value; var password = form.password.value; var city = form.ucity.value; if (name == "" || password == "" || city == "") { document.getElementById("msg").innerHTML = "Fields can not be blank";

} else { document.getElementById("msg").innerHTML =

"<table border=\"1\" width='500px'><tr><td>Name :</td><td>" +

name +

"</td></tr><tr><td>Password :</td><td>" +

password +

"</td></tr><tr><td>City :</td><td>" +

city +

}

}

"</td></tr></table>";

</script>

<form>

Name: <input type="text" name="uname" id="uname" /><br /><br />

Password: <input type="password" class="password" id="password" /><br /><br /> City:

<select name="ucity" id="ucity" class="ucity">

<option value="Ahmedabad">Ahmedabad</option>

<option value="Gandhinagar">Gandhinagar</option>

<option value="Mehsana">Mehsana</option></select

><br /><br />

<input type="submit" onclick="showdata(this.form); return false;" /><br />

</form>

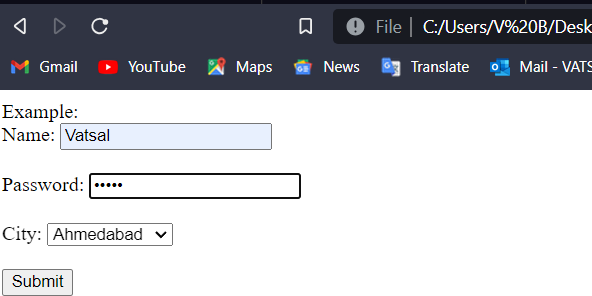
<p id="msg">

</p>

</body>

</html>

**Output :**



**JS array :**

A JavaScript array is a data structure that stores multiple values in a single variable. It's ordered and indexed, allowing access to elements using numerical indices. Arrays can hold various types of data, such as numbers, strings, objects, or even other arrays. They offer methods for adding, removing, and manipulating elements, making them versatile for storing and working with collections of data.

### Example **:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Array</title>

</head>

<body> <script> var emails = ["[a@b.com"](mailto:a@b.com), ["i@j.com"](mailto:i@j.com), ["x@y.com"](mailto:x@y.com)]; var password = ["abc123", "ijk123", "xyz123"]; function login(form) { var useremail = form.uemail.value; var userpassword = form.upassword.value;

if (useremail == "" || userpassword == "") { if (useremail == "" &&userpassword != "") { document.getElementById("msg").innerHTML =

"Email Field can not be blank";

} else if (useremail != "" &&userpassword == "") { document.getElementById("msg").innerHTML =

"Password Field can not be blank";

} else {

document.getElementById("msg").innerHTML = "Both are required";

} } else {

for (var i = 0; i<emails.length; i++) {

if (useremail == emails[i] &&userpassword == password[i]) { document.getElementById("msg").innerHTML = "<font style='color:green'>Login

Successful</font>"; break; } else {

document.getElementById("msg").innerHTML =

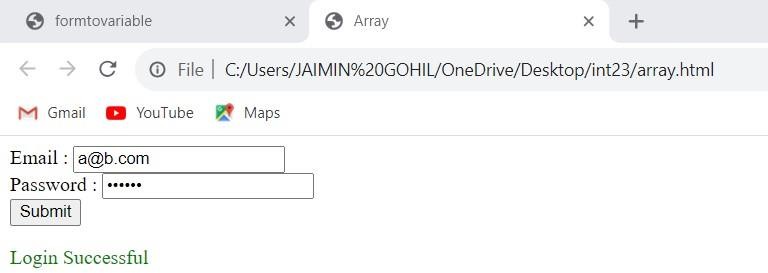
"<font style='color:red'>Invalid username and password</font>";

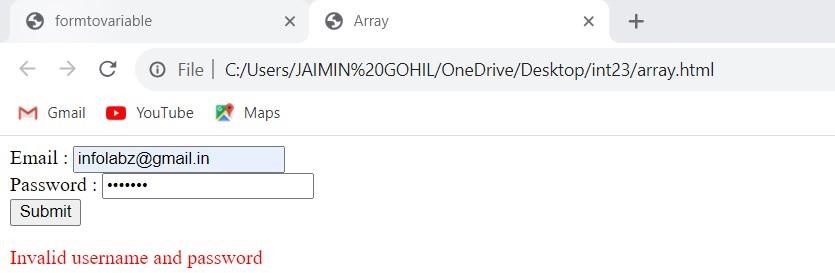
}

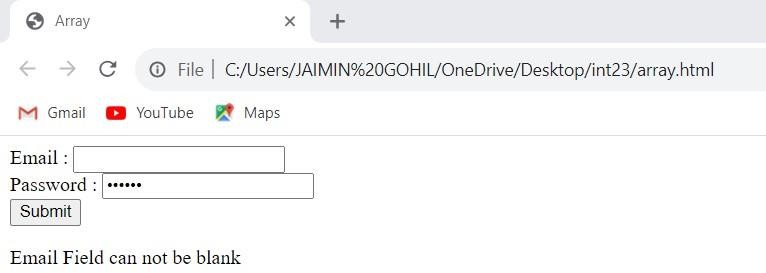
}

} }

</script>

**Output :**





# DAY-5

**Task Assignment :**

1. **Covid Data Search :**

Example **:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<title>Assignment 1 Covid</title>

<style> .date { width: 250px;

}

#submit { border-radius:

8px;

} form { margin-

top: 10px;

} td { width:

100px; text-align: right;

}

</style>

</head>

<body>

<script>

async function validate() {

let url = "https://data.covid19india.org/data.json";

let myobject = await (await fetch(url)).json(); var date

= document.getElementById("date").value; if (date

== "") {

// alert("Please select Date. ")

document.getElementById("msg").innerHTML =

"<font style='color:red'>Please Enter Date. </font>";

}

else { for (var i = 0; i<myobject["cases\_time\_series"].length; i++) { if (date != myobject["cases\_time\_series"][i]["dateymd"]) { document.getElementById("msg").innerHTML =

"<font style='color:red'>No record founded..</font>";

// alert("No record founded..")

} else {

// document.getElementById("tablee").innerHTML =

// "No record founded.."; document.getElementById("msg").innerHTML =

"<table border='2' width='200px'><tr><td>New Cases</td><td>" + myobject["cases\_time\_series"][i]["dailyconfirmed"] +

"</td></tr><tr><td>New Deaths</td><td>" + myobject["cases\_time\_series"][i]["dailydeceased"] +

"</td></tr></table>";

break;

}

}

}

}

</script>

<div class="container-fluid">

<form>

<b>Date</b> :

<input type="date" class="date" id="date" placeholder="Enter Date 30 January 2020" />

<input type="submit" class="btn-success" id="submit" value="Submit" onclick="validate(); return false; " />

<!--<button type="submit" class="submit" id="submit">Submit</button> -->

</form>

<br />

<p id="msg"></p>

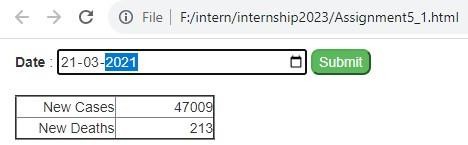
</div>

</body>

</html>

**Output :**





1. **Mutual Fund API :**

### Example **:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Assignment 2</title>

</head>

<body>

<script> async function validate() { let url = "https://api.mfapi.in/mf"; let myobject = await (await fetch(url)).json(); var c = 0;

for (var i = 0; i<myobject.length; i++) { c = c

+ 1;

}

document.getElementById("schemes").innerHTML = "Total number of schemes : " + c;

}

async function show() { let url = "https://api.mfapi.in/mf"; let myobject = await (await fetch(url)).json(); var code = document.getElementById("scheme").value; for (var i = 0; i<myobject.length; i++) { if (code == myobject[i]["schemeCode"]) { document.getElementById("msg").innerHTML = "<b>Scheme Name : <font style='color:green'>" + myobject[i]["schemeName"]+"</font></b> "; break;

}

else if(code==""){ document.getElementById("msg").innerHTML = "<font style='color:red'>Please enter scheme code</font>";

} else{

document.getElementById("msg").innerHTML = "<font style='color:red'>Scheme not found</font>";

} } }

</script>

<p id="schemes"></p>

<form>

Scheme code :<input type="number" class="scheme" id="scheme" /><br />

<button type="submit" class="submit" id="submit" onclick="show(); return false;"> Show fund

</button>

<br />

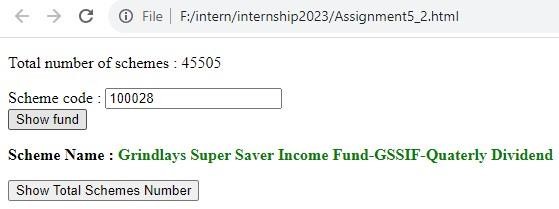
<p id="msg"></p>

<button type="submit" class="submit" id="submit" onclick="validate(); return false;"> Show Total Schemes Number

</button></form>

</body> </html>

**Output :**





# Day-6

* 1. React Environment setup ( Node JS installation )
  2. First React App
  3. Functional Components
  4. Class Components

**React Environment setup (Node JS installation) :**

Setting up a React environment involves installing Node.js and using the Node Package Manager (npm) to manage dependencies. Here's a basic overview of the process:

1. **Install Node.js:**

* Visit the official Node.js website (https://nodejs.org/).
* Run the installer and follow the installation instructions.

1. **Verify Installation:**

- Open a terminal (or command prompt) and run the following commands to verify that Node.js and npm are installed:

node -v npm -v

1. **Create a React App:**

* To create a new React application, use `create-react-app`, a tool that sets up a basic React project structure.
* Run the following command in your terminal:

npx create-react-app my-react-app

* Replace "my-react-app" with the name you want for your project.

1. **Navigate to the Project:**

- Navigate to the project folder using:

cd my-react-app

1. **Start the Development Server:**

- Once in the project folder, start the development server with:

npm start

-This will launch your React application in a web browser. The development server will automatically reload as you make changes to your code.

**App.js :**

import logo from "./logo.svg"; import "./App.css";

function App() { return (

<div>

<h1>Hello</h1>

</div>

);

}

function Msg() { return (

<div>

<h2> New Message</h2>

</div>

);

}

export function Add()

{ var a = 100, b = 900; return ( <div>

<h2>

{" "}

Addition of {a} and {b} is {a + b}

</h2>

</div>

);

}

export default App; export { Msg };

**index.js**

import React from 'react'; import ReactDOM from 'react-dom/client'; import './index.css'; import App, { Add, Msg } from './App'; import reportWebVitals from './reportWebVitals'; import

{ Mycomp, Mycomp1 } from './custom';

const root = ReactDOM.createRoot(document.getElementById('root')); root.render(

<React.StrictMode>

<App />

<Msg />

<Add/>

<Mycomp />

<Mycomp1 />

</React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log)) // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals reportWebVitals();

custom.js export function Mycomp(){ return(

<div>

<h1> First Custom Component</h1>

</div>

);

}

export function Mycomp1(){ return(

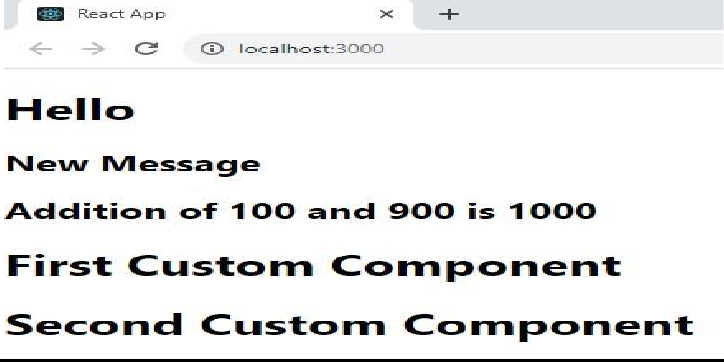
<div>

<h1> Second Custom Component</h1>

</div>

);

}

**Output :**

# Day-7

**Variable Data Map:**

Example**:**

Index.js

import React from 'react'; import ReactDOM from 'react-dom/client'; import './index.css'; import App, { Display, Newcomp } from './App'; import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root')); root.render(

<React.StrictMode>

<App />

<Newcomp />

<Display />

</React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals reportWebVitals();

App.js

import logo from './logo.svg'; import './App.css';

const data = "hi"; function App() { constnewdata = "how are you?"; return ( <div>

<h1>Data variable</h1>

<h1>{newdata}</h1>

<h1>{data}</h1> </div> );

constnewacc = [5,10,15,20,25]; constdataa = "hi";

export function Newcomp(){ return (

<div>

<h1>Hello</h1>

<h1>{dataa}</h1>

<h1>{newacc[3]}</h1>

</div>

);

}

constalldata = [5,10,15,20,25]; export function Display(){ return( <div>

{ alldata.map( (dataaa)=>{ return <h1>{dataaa}</h1>

}

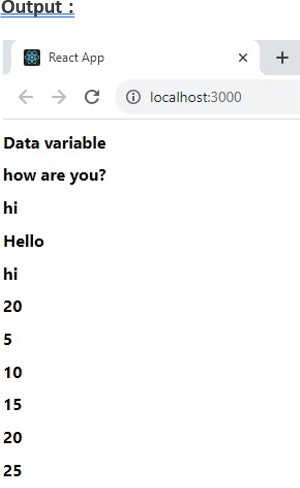
)

}

</div>

);

} export default App;



**Object Map:**

### Example**:**

index.js

import React from 'react'; import ReactDOM from 'react-dom/client'; import './index.css'; import App, { Datashow, Isrodata, New, Tblshow } from './App'; import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root')); root.render(

<React.StrictMode>

<App />

<Datashow />

{/\* <Tblshow /> \*/}

<Isrodata />

<New />

</React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals reportWebVitals(); App.js

import logo from "./logo.svg"; import "./App.css"; import { Details, Multiple } from "./props";

function App() { return (

<div className="App">

<h1>Object Data Map</h1>

</div>

);

}

const data = { KEY1: "VALUE1", KEY2: "VALUE2" }; function Datashow() { return (

<div>

<table border="1">

<tr>

<th>KEY NAME</th>

<th>VALUE NAME</th>

</tr>

{Object.keys(data).map((items) => { return (

<tr><td>

{items}</td><td>{data[items]}</td>

</tr>

);

})}

</table><br>

</div>

);

}

// https://isro.vercel.app/api/spacecrafts constisrodata = { spacecrafts: [

{ id: 1, name: "Aryabhata" },

{ id: 2, name: "Bhaskara-I" },

{ id: 3, name: "Rohini Technology Payload (RTP)" },

{ id: 4, name: "Rohini Satellite RS-1" },

{ id: 5, name: "Rohini Satellite RS-D1" },

{ id: 6, name: "APPLE" },

{ id: 7, name: "Bhaskara-II" },

{ id: 8, name: "INSAT-1A" },

{ id: 9, name: "Rohini Satellite RS-D2" },

{ id: 10, name: "INSAT-1B" },

{ id: 11, name: "SROSS-1" },

{ id: 12, name: "IRS-1A" },

{ id: 13, name: "SROSS-2" },

{ id: 14, name: "INSAT-1C" },

{ id: 15, name: "INSAT-1D" },

{ id: 16, name: "IRS-1B" },

{ id: 17, name: "SROSS-C" },

{ id: 18, name: "INSAT-2A" },

{ id: 19, name: "INSAT-2B" },

{ id: 20, name: "IRS-1E" },

],

};

export function Isrodata() { return (

<div>

<table border="1">

<tr>

<th>ID</th>

<th>Name</th>

</tr>

{isrodata.spacecrafts.map((value) => { return(

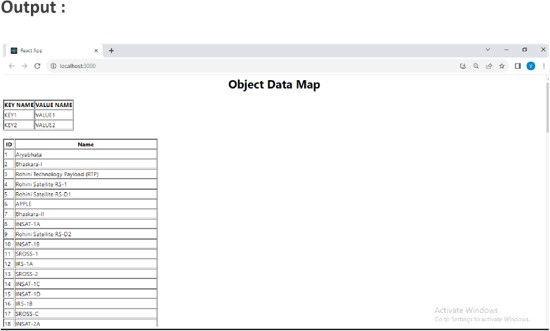
<tr><td>

(value.id}</td> <td> {value.name}</td>

</tr>);

</table>

</div>):

export default App: export { Datashow };

# Day-8

**React Props :**

Example**:**

App.js

export function New(){ return( <div>

<h1> New Components</h1> <Details name="vraj"/>

<Details name="Gadhiya/>

{/\* <Multiple name="" email =["Vraj@gm](mailto:Vraj@gmail.com)a[il.com](mailto:Vraj@gmail.com)"/> \*/}

<Multiple name="Vraj" e[mail="V](mailto:Vraj@gmail.com)raj@gm[ail.com"](mailto:Vraj@gmail.com)/>

<Multiple name="sahil" email=["sahil@gmail.com"](mailto:sahil@gmail.com) /> </div>

);

}

Props.js

export function Details(props){ return(

<div>

<h1> Hello, {props.name} </h1>

</div>

);

}

export function Multiple(props){ return ( <div>

<p> Hello, Name: {props.name} and Email: {props.email} </p>

</div>

);

}

**React Bootstrap:** Example**:** index.js

import React from 'react'; import ReactDOM from 'react-dom/client'; import './index.css'; import App, { Btn, Tablecom } from './App'; import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root')); root.render(

<React.StrictMode>

<App />

<Btn />

<Tablecom />

</React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals reportWebVitals();

App.js

// import logo from './logo.svg'; import './App.css'; import 'bootstrap/dist/css/bootstrap.min.css'; // import Button from 'react- bootstrap/Button'; import { Button, Table } from 'react-bootstrap';

// import { Table } from 'react-bootstrap';

function App() { return (

<div className="App">

<p>Hi</p>

</div>

);

}

export function Btn(){ return(

<div>

<Button variant="primary">Click</Button><br/><br/> </div>

);

} export function Tablecom(){ return( <div>

<Table striped bordered hover>

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Username</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Vraj</td>

<td>Gadhiya</td>

<td>@Vraj</td>

</tr>

<tr>

<td>2</td>

<td>sahil</td>

<td>vagh</td>

<td>@sahil</td>

</tr>

<tr>

<td>3</td>

<td colSpan={2}>vivekjain</td>

<td>@vivek</td>

</tr>

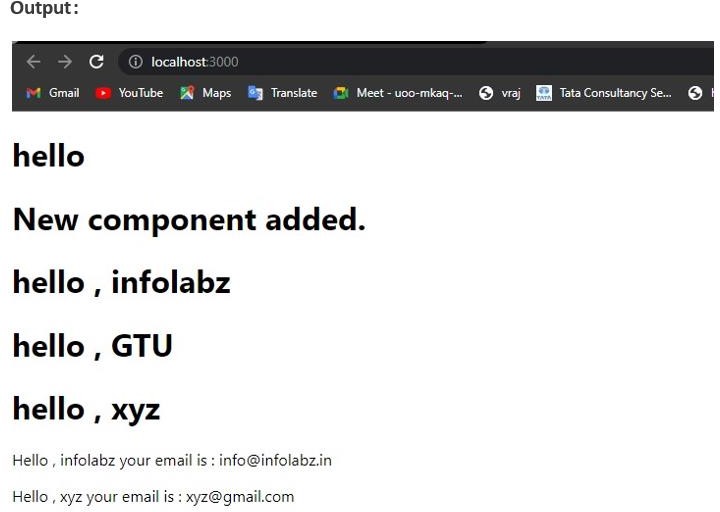
</tbody>

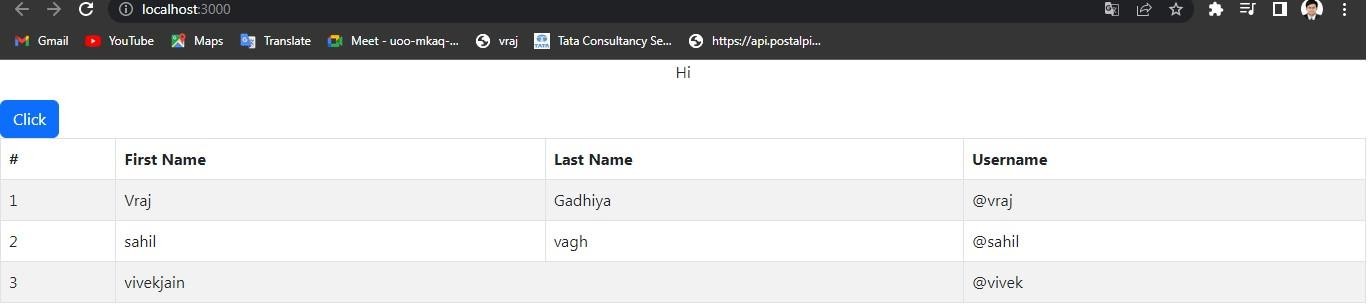
</Table>

</div>

);

} export default App;



**Output:**

# Day-9

* 1. React Hooks: UseEffect and UseState
  2. API data fetch with react

**useState:**

The **useState** hook is used in functional components to add state management to them. It allows you to declare and manage state variables within your component. It takes an initial value as an argument and returns an array with two elements: the current state value and a function to update that value.

In the example above, **count** is the state variable, and **setCount** is the function used to update it. When

**setCount** is called, it triggers a re-render of the component with the updated state value. import React, { useState } from 'react';

function Counter() { const [count, setCount]

= useState(0); return (

<div>

<p>Count: {count}</p>

<button onClick={() =>setCount(count + 1)}>Increment</button>

</div>

);

}

**useEffect:**

The **useEffect** hook is used to perform side effects in functional components. It takes two arguments: a function that contains the side effect code, and an optional array of dependencies. The function inside **useEffect** will run after the component renders and every time the dependencies change. If no dependencies are provided, the function runs after every render.

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

function DataFetching() {

const [data, setData] = useState([]);

useEffect(() => { fetch('https://api.example.com/data')

.then(response =>response.json())

.then(data =>setData(data));

}, []); // Empty dependency array: run only after the initial render

return ( <div> <ul>

{data.map(item => (

<li key={item.id}>{item.name}</li>

))}

</ul>

</div>

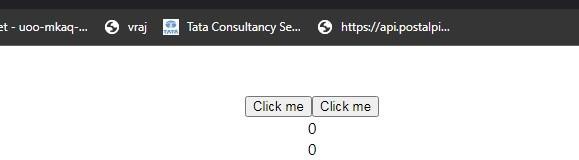
);

}

In this example, the **useEffect** hook is fetching data from an API when the component mounts (initial render) due to the empty dependency array. If you pass in dependencies (e.g., **[someValue]**), the effect will run whenever those dependencies change.

Remember that **useEffect** also returns a cleanup function that can be used to clean up any resources or subscriptions when the component unmounts or when the effect re-runs due to dependencies changing.

These hooks, **useState** and **useEffect**, are essential tools for managing state and handling side effects in React functional components, making them more powerful and concise compared to the older classbased approach.

**Output:**

# Day-10

**ASSIGNMENT TASK:**

**API BASED REAL TIME NEWS WEB APPLICATION:**

App.js

// import logo from "./logo.svg"; import "./App.css"; import { Container, Row, Col, Card } from "react-bootstrap"; import "bootstrap/dist/css/bootstrap.min.css"; import React, { useState, useEffect } from "react";

function App() { const [mydata, setData] = useState([]);

// https://inshortsapi.vercel.app/news?category=sports constapiget = () => { fetch("https://inshortsapi.vercel.app/news?category=sports")

.then((response) =>response.json())

.then((json) => { console.log(json); setData(json.data);

});

};

useEffect(() => { apiget(); const interval = setInterval(()

=> { apiget(); }, 500000); return ()

=>clearInterval(interval);

}, []); return (

<Container fluid>

<Row xs={1} md={3} className="g-4">

{mydata.map( (value) => { return (

<>

<Col className="container-fluid mt-4">

<Card>

<Card.Img variant="top" src={value.imageUrl} height="400px" />

{/\* <Card.Img variant="top" src="holder.js/100px160" /> \*/} <Card.Body>

<Card.Title>{value.title}</Card.Title>

<Card.Text>

{value.content}

</Card.Text>

<footer className="blockquote-footer"> Published on:

{value.date},{value.time}

</footer>

</Card.Body>

</Card>

</Col>

</>

)

})}

</Row>

</Container>

);

}

export default App;

index.js

import React from 'react'; import ReactDOM from 'react-dom/client'; import './index.css'; import App from './App'; import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root')); root.render(

<React.StrictMode>

<App />

</React.StrictMode>

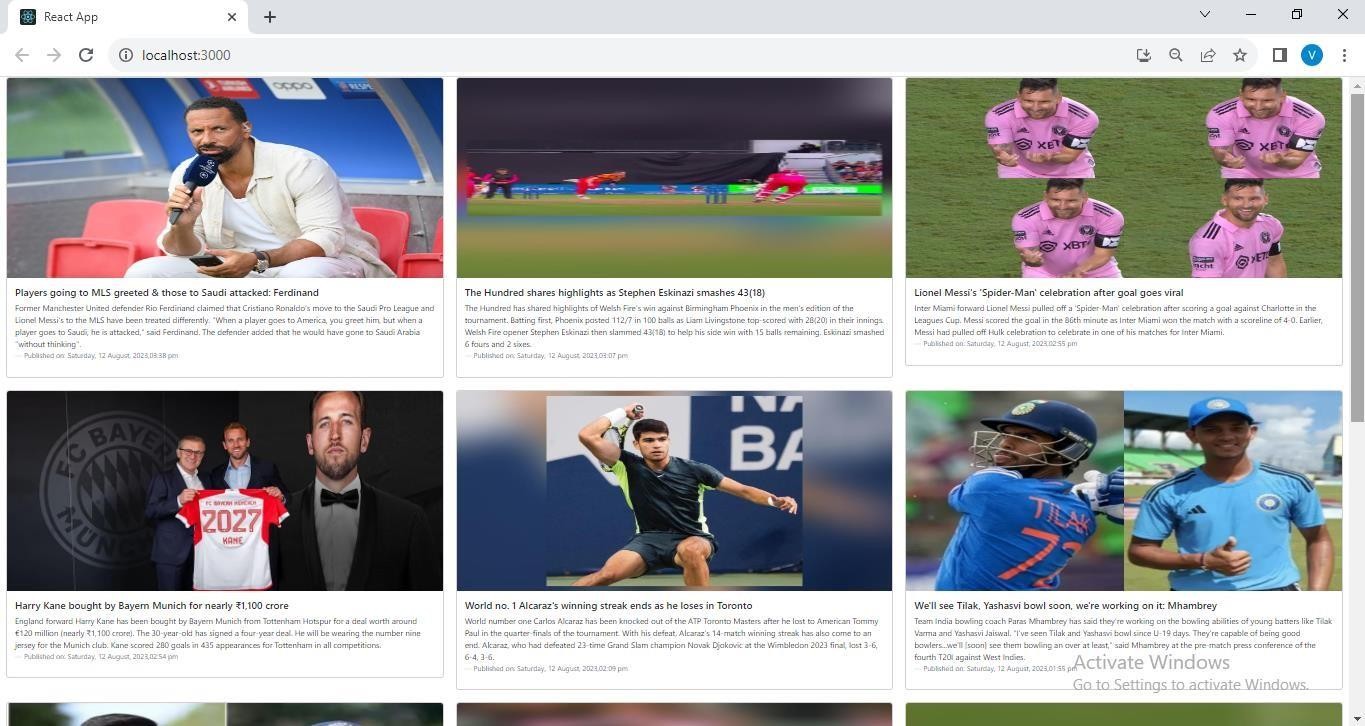
);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals reportWebVitals();

Output :



# Day-11

**Project &conclusion :**

#### index.js

import React from 'react'; import ReactDOM from 'react-dom/client'; import './index.css'; import App from './App'; import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root')); root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals reportWebVitals();

App.css

.App{ background-color:#e3ffe9;

}

.App-logo { height: 40vmin; pointer-events: none;

}

.Back{ background-image: url("News.jpg"); background- position: center; background-repeat: no-repeat; background- size: cover; height: 300px; border-radius: 10px; opacity: .8;

}

footer{ background-color: #daffa8; padding: 10px; margin- top: 110px; position: sticky; text-align: center; font-weight: bolder;

}

@media (prefers-reduced-motion: no-preference) {

.App-logo { animation: App-logo-spin infinite 20s linear;

}

}

.App-header { background-color: #282c34; min-height: 100vh; display: flex; flex-direction: column; align- items: center; justify-content: center; font-size: calc(10px + 2vmin); color: white;

}

.App-link { color: #61dafb;

}

@keyframes App-logo-spin { from { transform: rotate(0deg);

} to { transform: rotate(360deg);

}

}

App.js

import logo from "./logo.svg"; import "./App.css"; import { Container, Row, Col, Card, Table } from "react-bootstrap"; import "bootstrap/dist/css/bootstrap.min.css";

import React, { useState, useEffect } from "react";

function App() { const [mydata, setData] = useState([]);

constapiget = () => { fetch("https://inshorts.me/news/all?offset=0&limit=21")

.then((response) =>response.json())

.then((json) => { console.log(json); setData(json.data.articles);

});

};

useEffect(() => { apiget(); const interval = setInterval(() => { apiget();

}, 500000); return ()

=>clearInterval(interval);

}, []);

return (

<Container fluid>

<Row xs={1} md={5} className="g-4">

{mydata.map((value) => { return (

<>

<Col className="container-fluid mt-4">

<Card>

<a href={value.sourceUrl} target="\_blank">

<Card.Img variant="top" src={value.imageUrl} height="275px" /></a>

<Card.Body>

<Card.Title>{value.title}</Card.Title>

<Card.Text>

{value.content}

<br />

</Card.Text>

<footer className="blockquote-footer"> Author Name: {value.authorName}

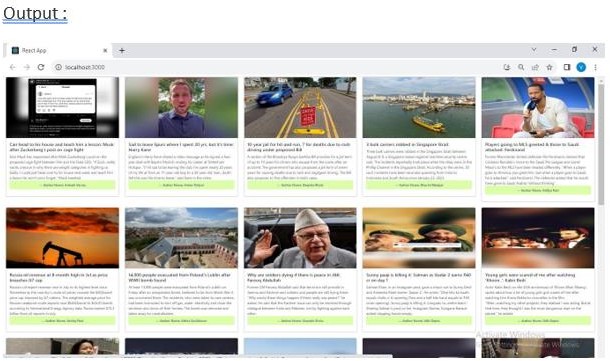
</footer>

</Card.Body>

</Card>

</Col> < );

})} <Row <Container);} export default App;



**References**

* + 1. <https://getbootstrap.com/>
    2. <https://www.w3schools.com/>
    3. <https://www.codewithharry.com/>
    4. [https://practice.geeksforgeeks.org](https://practice.geeksforgeeks.org/)
    5. <https://react.dev/>

## Conclusion

React JS has revolutionized web development, empowering developers to build high- performance, interactive, and scalable web applications. React JS is positioned to influence the direction of web development thanks to its component based architecture, virtual DOM, and broad ecosystem. As the industry continues to evolve, we can expect React JS to play an even more prominent role, especially with the rise of mobile app development using React Native. Whether you’re a seasoned developer or just starting your web development journey, embracing React JS can unlock a world of possibilities and open doors to innovative and dynamic web applications.