

FSWD LAB MANUAL

1. Form Registration

Aim:

To create registration form with validation using HTML, CSS, JavaScript

Code:

```
<html>
<head><title>Registration Form Example</title>
<style>
body{
    display: flex;
}
div{
    background-image: linear-gradient(145deg, #F1F645, #33FFCB, #45BEF6);
    width: 45%;
    justify-content: center;
    display: flex;
    margin: auto;
}

input{
    height: 20px;
    flex: 0 0 200px;
    margin-left: 60px;
}

input[type="tel"]{
    margin-left: 5px;
}
```

```
select{  
margin-left:13px;}
```

```
input[type="submit"]{  
margin-left:120;  
border-radius:5px;  
border:none;  
background-color:white;}
```

```
input[type="reset"]{  
border-radius:5px;  
border:none;  
background-color:white;}
```

```
</style>
```

```
<script>
```

```
function validate(){  
var n=document.forms["myform"]["name"].value;  
var email=document.forms["myform"]["email"].value;  
var i=email.indexOf("@");  
var j=email.lastIndexOf(".");  
if(i<1||j<i)  
{  
alert("Invalid email ID");  
e.focus();  
}  
localStorage.setItem("name", n);
```

```
window.open('welcome.html');
}
</script>
</head>
<body>
<div>
<form id="myform" method="get" onsubmit=validate()>
<center><h3>Register Form</h3></center>

<label>Name:</label>
<input type="text" name="name" required><br><br>

<label>Email:</label>
<input type="email" name="email" required><br><br>

<label>Gender:</label>
<input type="radio" value="male" name="g1">Male

<input type="radio" value="female" name="g1">Female

<input type="radio" value="others" name="g1">Transgender
<br><br>

<label>D.O.B:</label>
<input type="date" name="dob" required><br><br>

<label>Phone number:</label>
```

```
<input type="tel" id="phone" name="phone" placeholder="123-456-7890"
pattern="[0-9]{10}" required><br><br>
```

```
<label>Qualification:</label>
```

```
<select>
```

```
<option>SSLC</option>
```

```
<option>HSC</option>
```

```
<option>UG</option>
```

```
<option>PG</option>
```

```
</select>
```

```
<br><br>
```

```
<label>Address:</label><br><br>
```

```
<textarea rows="5" cols="30"></textarea><br><br>
```

```
<input type="submit">
```

```
<input type="reset">
```

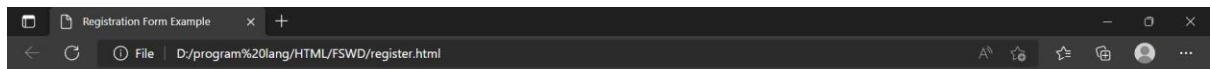
```
</form>
```

```
</div>
```

```
</body>
```

```
</html>
```

Output:



Register Form

Name:

Email:

Gender: ☐ Male ☐ Female ☐ Transgender

D.O.B:

Phone number:

Qualification:

Address:



2.Time Table

Aim:

To create time table using HTML, CSS, JS

Coding:

```
<html><head>
```

```
<style>
```

```
body{
```

```
background-image: linear-gradient(145deg,#45BEF6 , #5E6CDF,#45BEF6);}
```

```
h3{
```

```
line-height:10px;
```

```
color:white;
```

```
}
```

```
#year{
```

```
margin-left:300px;
```

```
font-weight:bold;
```

```
font-size:20px;}
```

```
#sem{
```

```
transform: translate(900px,-35px);
```

```
font-weight:bold;
```

```
font-size:20px;}
```

```
table{
```

```
width:65%;
```

```
margin:auto;
```

```
background-color:white;
text-align:center;}
```

```
th{
height:50px;
}
```

```
table, th, td {
border: 1px solid #802B85;
}
```

```
</style>
```

```
</head>
```

```
<body><br><br>
```

```
<center><h3>PSNA COLLEGE OF ENGINEERING AND
TECHNOLOGY,DINDIGUL-624622</h3>
```

```
<h3>DEPARTMENT OF COMPUTER APPLICATIONS</h3>
```

```
<u><h3>CLASS TIME TABLE</h3></u>
```

```
<u><h3>ACADEMIC YEAR 2021-2022[EVEN]</h3></u></center>
```

```
<br>
```

```
<p id="year">Year:I</p>
```

```
<p id="sem">Sem/Sec:II/A</p>
```

```
<table>
```

```
<tr><th>Day<th>8.45AM<br>9.35AM<th>9.35AM<br>10.25AM
```

```
<th rowspan="10" ><h2>B<br>R<br>E<br>A<br>K</h2>
```


<th>10.40AM
11.30AM<th>11.30AM
12.20PM

<th rowspan="10" ><h2>L
U
N
C
H</h2>

<th>1.30PM
2.20PM<th>2.20PM
3.10PM<th>3.10PM
4.00PM</tr>
>

<tr><th>MON<td>FSWD<td>MAD<td>ELE<td>PT<td colspan="10">FSWD
LAB</tr>

<tr><th>TUE<td>ADBT<td>ELE<td>CCT<td>CS<td>PT<td
colspan="5">MAD LAB</tr>

<tr><th>WED<td>CCT<td>FSWD<td>MAD<td>PT<td colspan="10">ADBT
LAB</tr>

<tr><th>THU<td>MAD<td>PT<td>ELE<td>FSWD<td>Audit
Course-
II<td>ADBT<td>CS</tr>

<tr><th>FRI<td>CS<td>ADBT<td>Audit
Course-II<td>CCT<td
colspan="10">CSE LAB-II</tr>

</table>

</body>

</html>

Output:

timetable.html

D:/program%20lang/HTML/FSWD/timetable.html

PSNA COLLEGE OF ENGINEERING AND TECHNOLOGY,DINDIGUL-624622

DEPARTMENT OF COMPUTER APPLICATIONS

CLASS TIME TABLE

ACADEMIC YEAR 2021-2022[EVEN]

Year:I

Sem/Sec:II/A

Day	8.45AM 9.35AM	9.35AM 10.25AM	B R E A K	10.40AM 11.30AM	11.30AM 12.20PM	L U N C H	1.30PM 2.20PM	2.20PM 3.10PM	3.10PM 4.00PM
MON	FSWD	MAD		ELE	PT		FSWD LAB		
TUE	ADBT	ELE		CCT	CS		PT	MAD LAB	
WED	CCT	FSWD		MAD	PT		ADBT LAB		
THU	MAD	PT		ELE	FSWD		Audit Course-II	ADBT	CS
FRI	CS	ADBT		Audit Course-II	CCT		CSE LAB-II		

27°C Cloudy

10:16 09-08-2022

3.Simple Program Using React JS

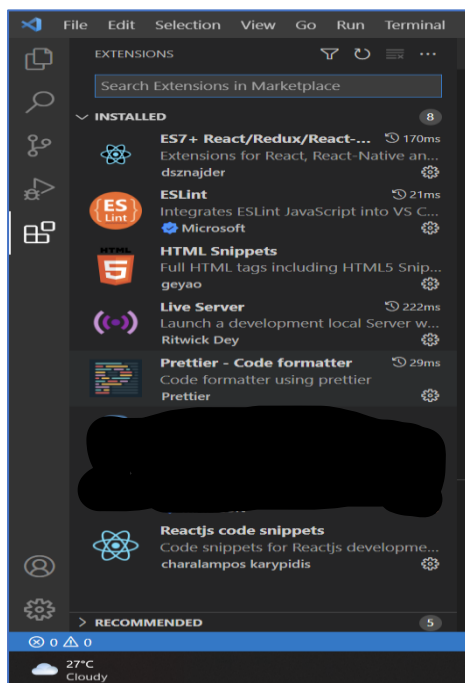
Aim:

To download and install react JS and to perform simple task using reactJS

Steps:

Step1: Download and install VS code

Step2: Select extension tab on left side and download extension like HTML snippets, Live server, ES lint, Reactjs code snippets, Prettier - Code formatter, ES7+ React/Redux/React-Native snippets.



Step3: Download and install nodeJS

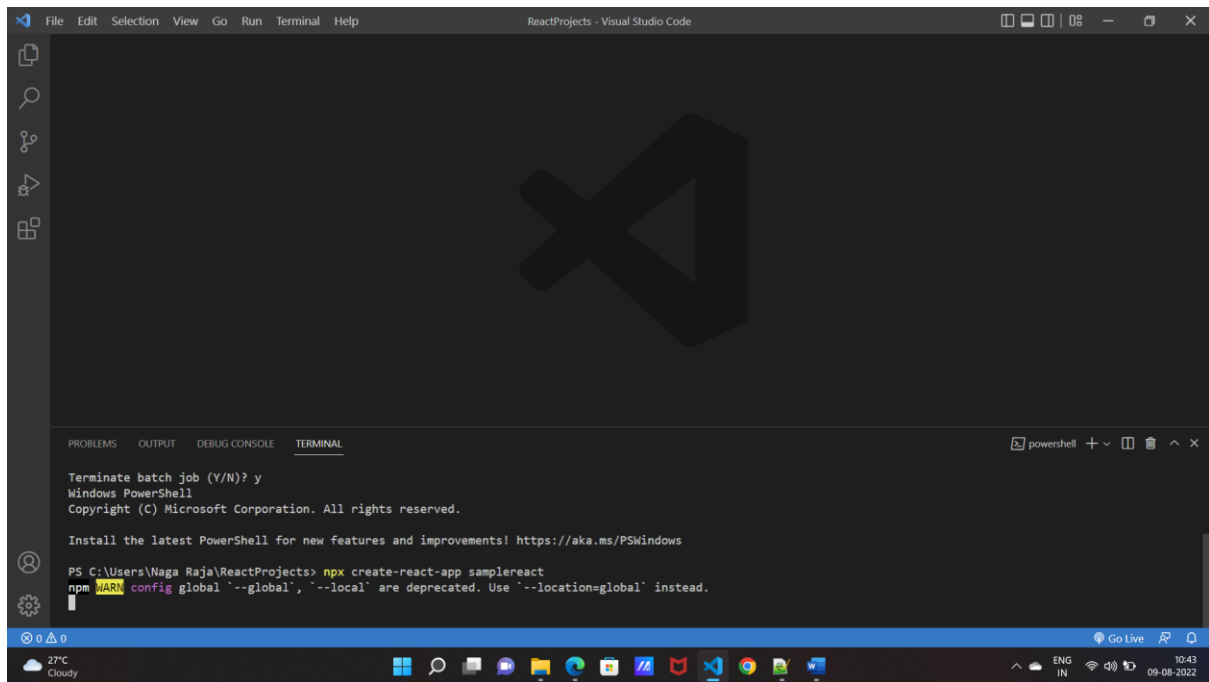
Step4: In VS code Terminal, type 'node'

Step5: Include or update npm and type '`npm install -g npm`' to verify is present or not

Step6: Create a directory by using `mkdir ReactProjects`

Step7: Enter into your folder using `cd ReactProjects`

Step8: To create a ReactJS project using '`npx create-react-app samplereact`'



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal is running a PowerShell session. The user has executed the command `npm create-react-app samplereact`. The output shows a warning about deprecated flags and a prompt to install the latest PowerShell. The user has responded with 'y' to the prompt.

```
PS C:\Users\Naga Raja\ReactProjects> npm create-react-app samplereact
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.

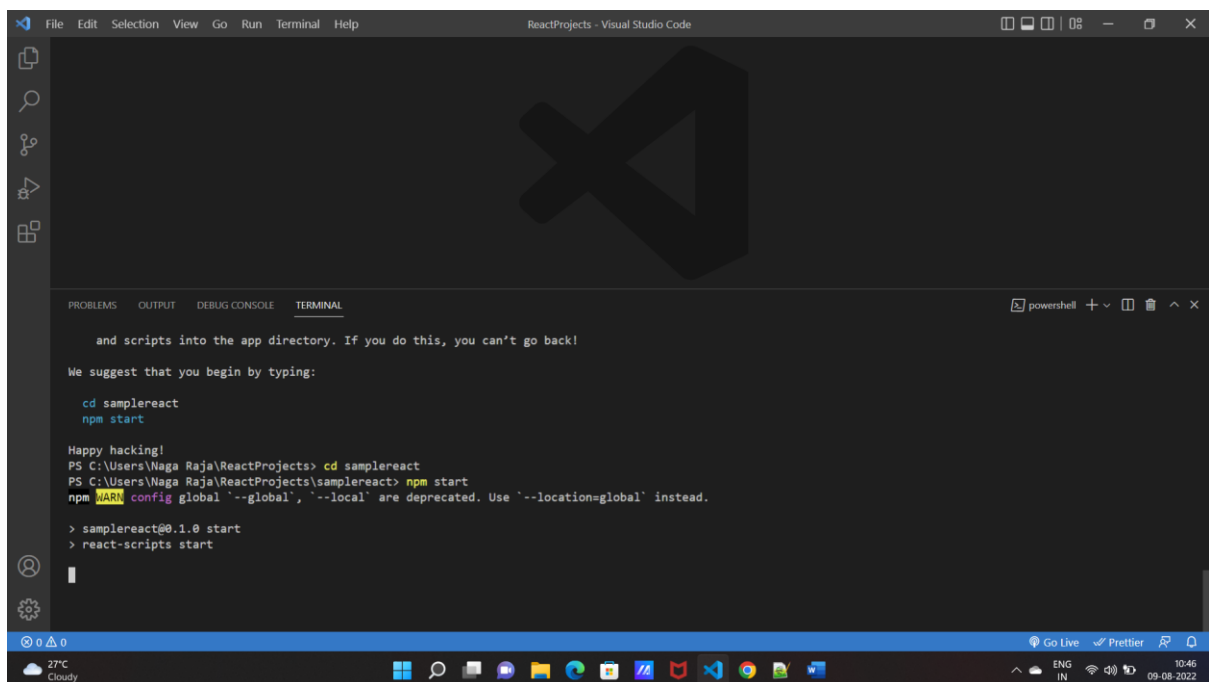
Terminate batch job (Y/N)? y
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Naga Raja\ReactProjects>
```

Step9: Again enter into folder `cd samplereact`

Step10: To start the server using `'npm start'`



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal is running a PowerShell session. The user has executed the command `cd samplereact` and then `npm start`. The output shows a warning about deprecated flags and a prompt to install the latest PowerShell. The user has responded with 'y' to the prompt.

```
and scripts into the app directory. If you do this, you can't go back!

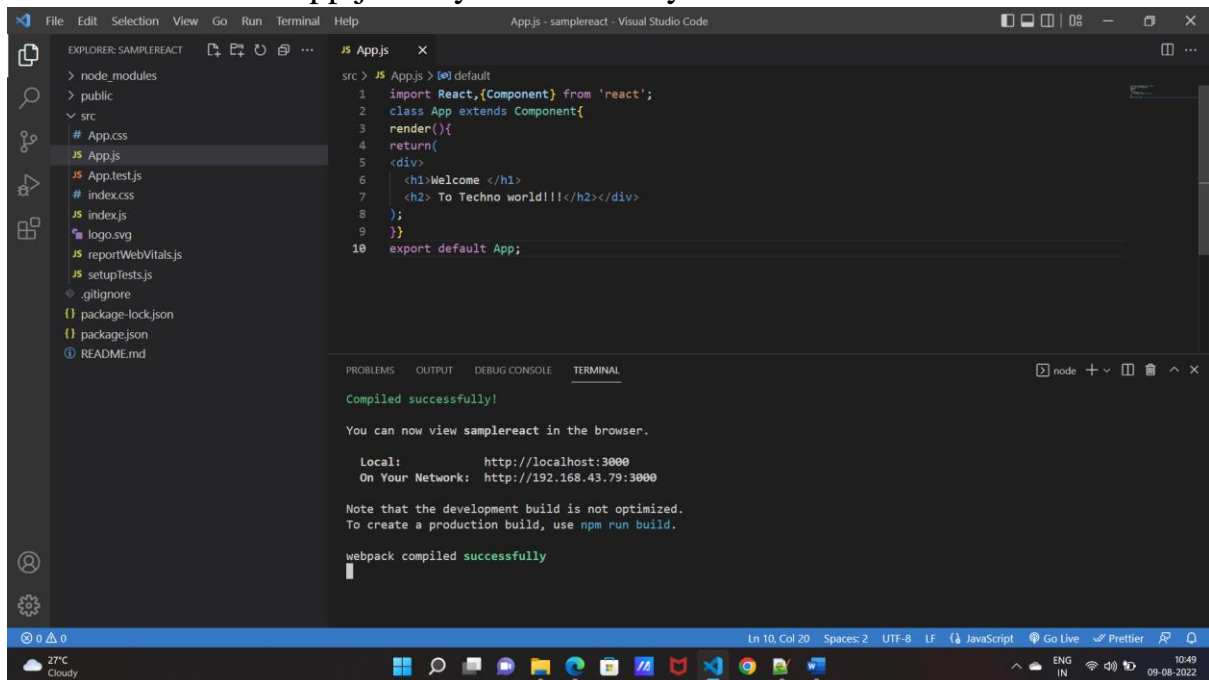
We suggest that you begin by typing:

  cd samplereact
  npm start

Happy hacking!
PS C:\Users\Naga Raja\ReactProjects> cd samplereact
PS C:\Users\Naga Raja\ReactProjects\samplereact> npm start
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.

> samplereact@0.1.0 start
> react-scripts start
```

Step11: In vs code open file menu and open folder ReactProject, click SRC in left side and select app.js and you can write your code here.



The screenshot shows the Visual Studio Code interface with the 'App.js' file open in the editor. The file contains the following code:

```
src > JS App.js > default
1 import React,{Component} from 'react';
2 class App extends Component{
3   render(){
4     return(
5       <div>
6         <h1>Welcome </h1>
7         <h2> To Techno world!!!</h2></div>
8     );
9   }
10  export default App;
```

The terminal at the bottom shows the following output:

```
Compiled successfully!

You can now view samplereact in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.43.79:3000

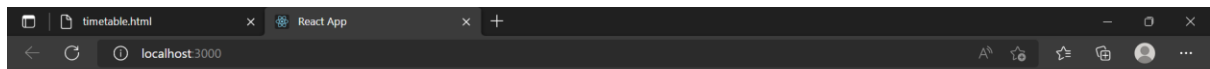
Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Program:

```
import React,{Component} from 'react';
class App extends Component{
  render(){
    return(
      <div>
        <h1>Welcome </h1>
        <h2> To techno world!!!</h2>
      </div>
    );
  }
}
export default App;
```

Output:



Welcome

To Techno world!!!



4.Login Page Using ReactJS

Aim:

To create a login page using ReactJS.

Coding:

---Copy the following code in App.js file ---

```
import './App.css';
function call(){
  var name=document.getElementById("uname");
  var pswd=document.getElementById("psw");

  alert("USERNAME="+name.value+"\nPassword="+pswd.value);
}
function App() {
  return (

    <body>
      <div>
        <form>
          <center><h3>Login</h3></center>
          <label>Username:</label>
          <input type="text" id="uname"></input><br/><br/>

          <label>Password:</label>
          <input type="password" id="psw"></input><br/><br/>

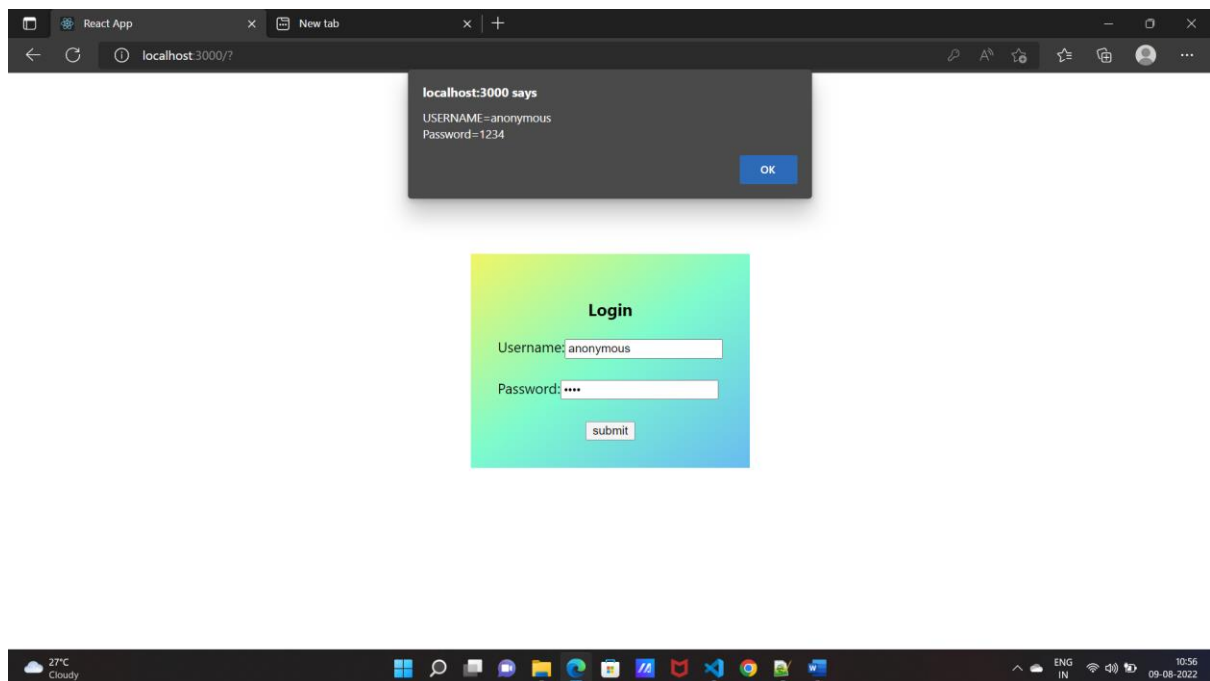
          <center><input type="Submit" value="submit" onClick={call}/></center>
        </form>
      </div>
    </body>
  );
}

export default App;
```

---Copy the following code in App.css file ---

```
body{
  margin-top:100px;
  display:flex;
  justify-content:center;
}
div{
  margin:auto;
  display:flex;
  align-items:center;
  justify-content:center;
}
form{
  padding:30px;
  background-image: linear-gradient(145deg,#F1F645 , #33FFCB,#45BEF6);
}
```

Output:



5.Fetch API Using JSON

Aim:

To write a program using fetch to read the JSON and to display it.

Algorithm:

Step 1: In VS code add live server extension.

Step 2: Create one folder(ex:fetch)

Step 3: Create a JSON file and save as one.json in your folder named fetch

```
[
  {
    "id": "1",
    "firstName": "jaya",
    "lastName": "kumar"
  },
  {
    "id": "2",
    "firstName": "Mary",
    "lastName": "Peter"
  },
  {
    "id": "3",
    "firstName": "George",
    "lastName": "Hansen"
  }
]
```

Step 4: Write a program using fetch() to read the JSON object and display it and save the file as index.html in the fetch folder (Same as step 3)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>JSON Test</title>
</head>
<body>
  <div id="myData"></div>
  <script>
    fetch('people.json')
      .then(function (response) {
        return response.json();
```

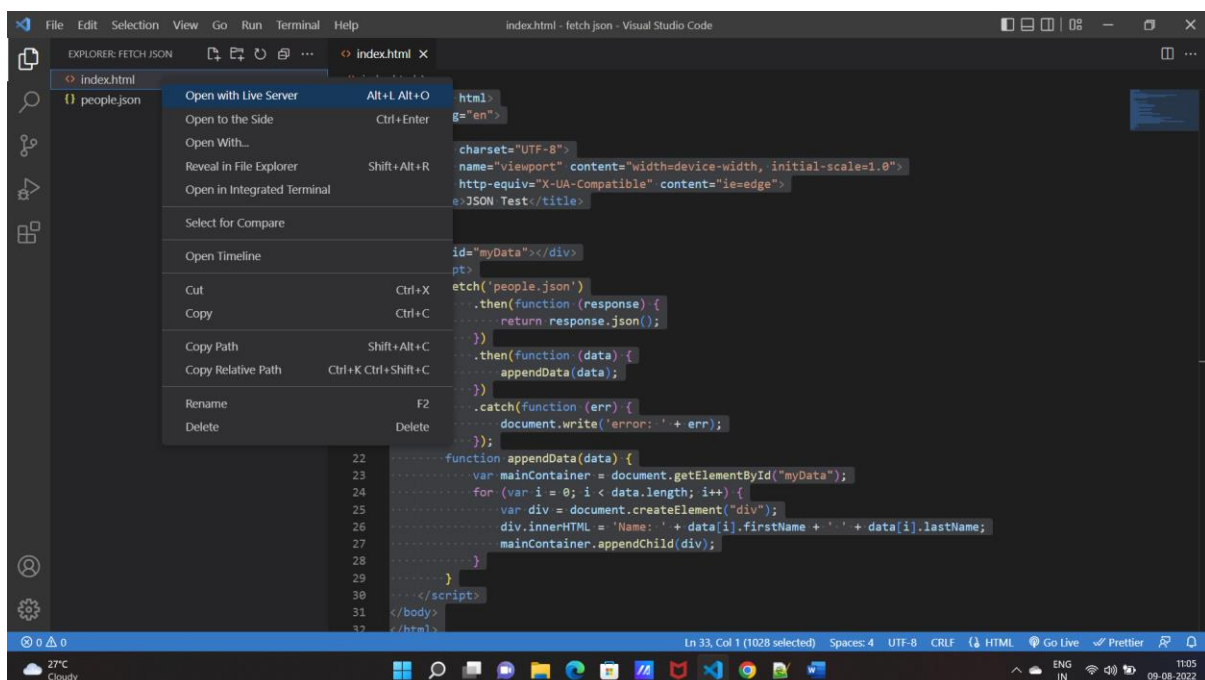
```

    })
    .then(function (data) {
        appendData(data);
    })
    .catch(function (err) {
        document.write('error: ' + err);
    });
function appendData(data) {
    var mainContainer = document.getElementById("myData");
    for (var i = 0; i < data.length; i++) {
        var div = document.createElement("div");
        div.innerHTML = 'Name: ' + data[i].firstName + ' ' +
data[i].lastName;
        mainContainer.appendChild(div);
    }
}
</script>
</body>
</html>

```

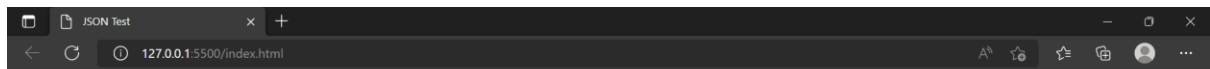
Step 5: open the folder in VS code.

Step 6: Right click the index.html then click open with live server to display the data.



Step 7: Else open web browser and type <http://127.0.0.1:5500/index.html>

Output:



Name: jaya kumar
Name: Mary Peter
Name: George Hansen



6.Counter Using ReactJS

Aim:

To create a counter program using ReactJS

Algorithm:

Step 1: Install VS code

Step 2: Create new ReactJS program inside the ReactProjects using

`npm create-react-app counter`

Step 3: Replace the following program inside into the App.js file

```
import React, {Component} from 'react';
import './App.css';

class App extends Component{
  constructor(props){
    super(props);
    this.state={
      count: 0
    }
  }
  increment=()=>{
    this.setState({count:this.state.count+1});
  }
  decrement=()=>{
    this.setState({count:this.state.count-1});
  }
  render(){
    return(
      <body>
        <div className='App' >
          <h1>Counter</h1>
          <button onClick={this.increment} className="counter">+</button>
          <button onClick={this.decrement} className="counter">-</button>
          <h2>{this.state.count}</h2>
        </div>
      </body>
    );
  }
}
export default App;
```

Step 4: Replace the following program inside into the App.css file

```
body{
  background: #009393;
}
.App {
  text-align: center;
  margin-top: 150px;
  background-color: aliceblue;
}
button{
font-size: 20px;
padding: 15px 32px;
margin-left: 8px;
}
```

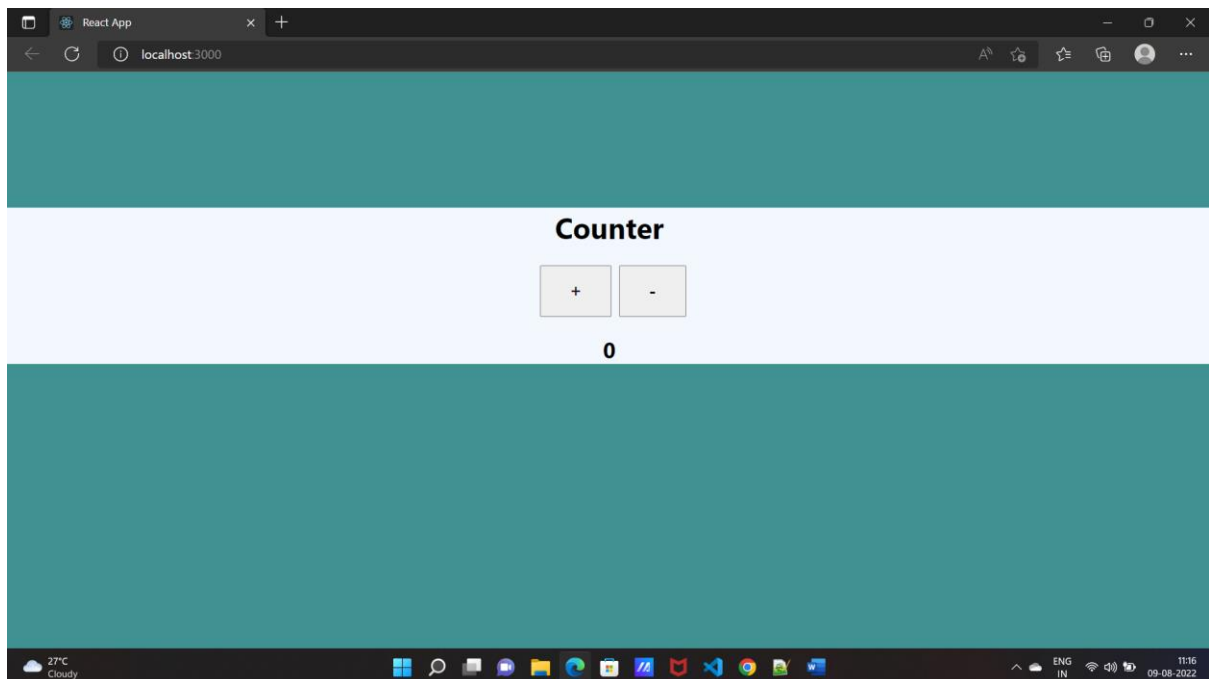
Step 5: Use the following commands to run the program

> cd ReactProjects

> cd counter

> npm start

Output:



7. NodeJS Server without Express

Aim:

Write a program to calculate and display student marks and average using NodeJS server without express

Procedure:

Step 1: Install NodeJS in your machine

Step 2: Create a new folder(D:/demonode)

Step 3: Open the notepad and write the code for demonode.js

```
var t;  
exports.caltotal=function(a,b,c,d,e){  
    t=a+b+c+d+e;  
    return t;  
};  
exports.calavg=function(){  
    return t/5;  
};
```

Step 4: Create the server program demo.js which make use of moduledemo.js by using require function

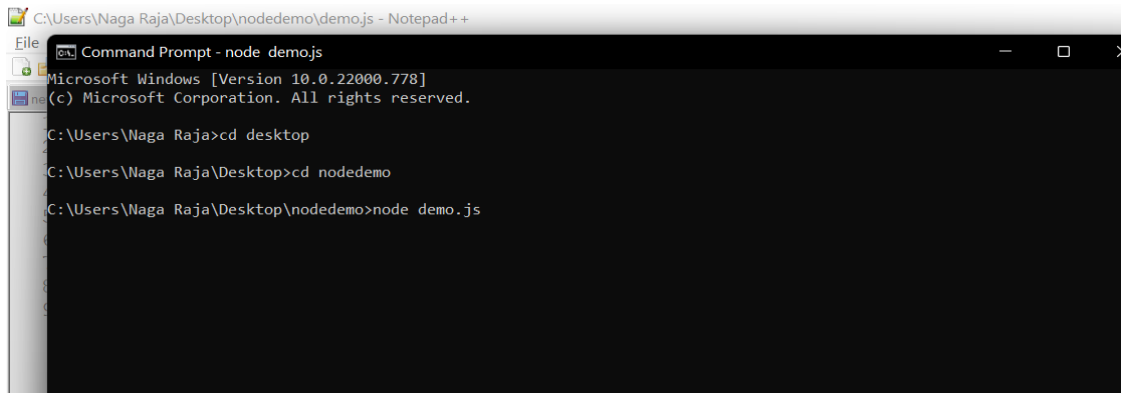
```
var http = require('http');  
var ct=require('./demomodule');  
http.createServer(function (req, res) {  
    res.writeHead(200, {'Content-Type': 'text/html'});  
    res.write("Total:"+ct.caltotal(50,50,50,50,50));  
    res.write("\t\t\t Average:"+ct.calavg());  
    res.end();  
}).listen(3000);
```

Step 5: Open command prompt and enter your folder location(D:/demonode).

Run the server by

>node demo.js

Server will get started

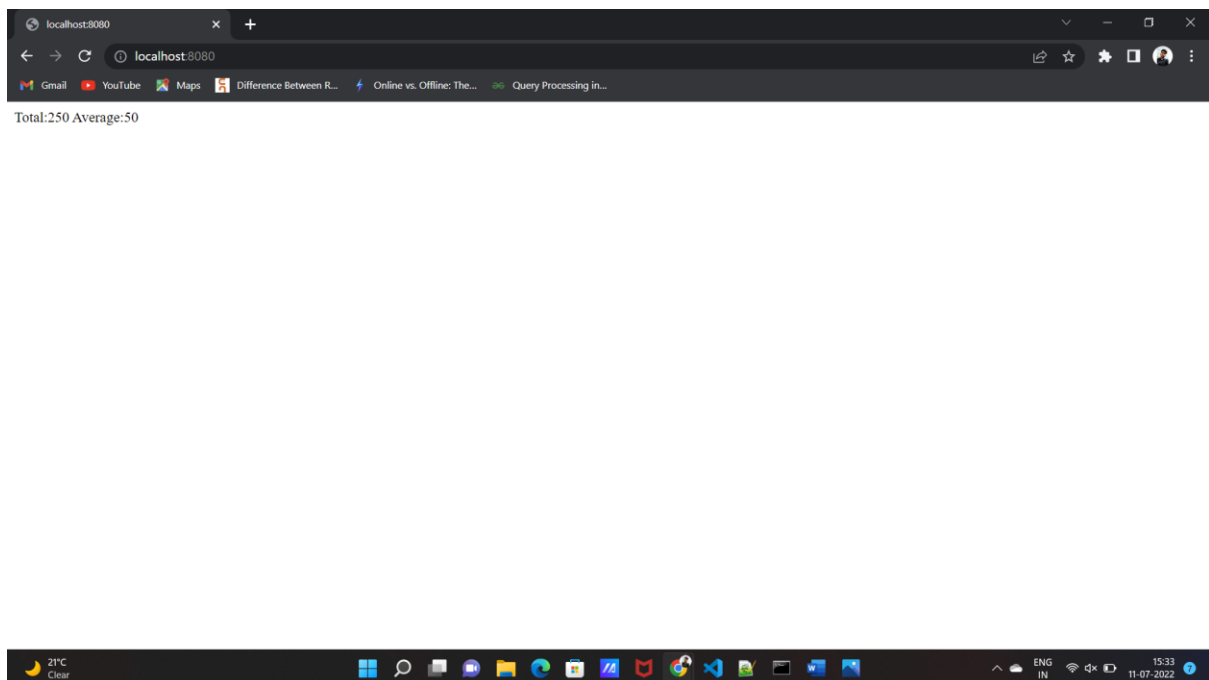


```
C:\Users\Naga Raja\Desktop\nodedemo\demo.js - Notepad++
File Edit Command Prompt - node demo.js
Microsoft Windows [Version 10.0.22000.778]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Naga Raja>cd desktop
C:\Users\Naga Raja\Desktop>cd nodedemo
C:\Users\Naga Raja\Desktop\nodedemo>node demo.js
```

Step 6: Display the result in the web browser by using
<http://localhost:8080>

Output:



8.MongoDB CURD operations

AIM:

To perform CURD operations using MongoDB shell.

Procedure:

1. CREATE THE DATABASE:

Create a database named student in the shell prompt by executing the following command.

```
> use student
```

switched to db student

2. INSERT THE COLLECTIONS INTO THE DATABASE:

Insert document into collections named student by executing the following command in the MongoDB Shell prompt.

```
> db.student.insert({"name":"gayathri","roll no":001,"dept_name":"Maths","college":"holy cross college","extra_curriclr":"book reading"});
```

```
> db.student.insert({"name":"malini","roll no":002,"dept_name":"computer science","college":"HCC college","extra_curriclr":"dance"});
```

```
> db.student.insert({"name":"Malathi","roll no":003,"dept_name":"MCA","college":"ABC college","extra_curriclr":"drawing"});
```

The above document is executed successfully, we assume it as true.

3. READ COLLECTIONS FROM THE DATABASE:

Read the document from the collections named student by executing the following command in the MongoDB Shell prompt.


```
> db.student.find().pretty();
```

```
{
  "_id" : ObjectId("603c0d13371e685b222cd0de"),
  "name" : "gayathri",
  "roll no" : 1,
  "dept_name" : "Maths",
  "college" : "holy cross college",
  "extra_curriclr" : "book reading"
}
{
  "_id" : ObjectId("603c0d9a371e685b222cd0df"),
  "name" : "malini",
  "roll no" : 2,
  "dept_name" : "computer science",
  "college" : "HCC college",
  "extra_curriclr" : "dance"
}
{
  "_id" : ObjectId("603c0e74371e685b222cd0e0"),
  "name" : "Malathi",
  "roll no" : 3,
  "dept_name" : "MCA",
  "college" : "ABC college",
  "extra_curriclr" : "drawing"
}
```

```
> db.student.findOne({"name":"gayathri"});
```

```
{
  "_id" : ObjectId("603c0d13371e685b222cd0de"),
  "name" : "gayathri",
  "roll no" : 1,
  "dept_name" : "Maths",
  "college" : "holy cross college",
  "extra_curriclr" : "book reading"
}
```

4. UPDATING THE COLLECTIONS INTO THE DATABASE:

We can also use the update keyword to update and alter collections using set and unset collections.

```
> db.student.update({"college":"ABC college"},{$set:{"college":"GTN college"}});
```

```
> db.student.findOne({"name":"Malathi"});
```

```
{
  "_id" : ObjectId("603c0e74371e685b222cd0e0"),
  "name" : "Malathi",
  "roll no" : 3,
  "dept_name" : "MCA",
  "college" : "GTN college",
  "extra_curriclr" : "drawing"
}
```

```
> db.student.findOne({"name":"malini"});
```

```
{
  "_id" : ObjectId("603c0d9a371e685b222cd0df"),
  "name" : "malini",
  "roll no" : 2,
```

```
"dept_name" : "computer science",  
"college" : "HCC college",  
"extra_curriclr" : "dance"  
}
```

```
>db.student.update({"name":"malini"},{$unset:{"extra_curriclr":"dance"}  
});
```

```
> db.student.findOne({"name":"malini"});
```

```
{  
  "_id" : ObjectId("603c0d9a371e685b222cd0df"),  
  "name" : "malini",  
  "roll no" : 2,  
  "dept_name" : "computer science",  
  "college" : "HCC college",  
  "extra_curriclr" : "dance"  
}
```

5. DELETING THE COLLECTIONS FROM THE DATABASE:

The following MongoDB delete command is used to remove the documents which Belongs to the collection.

```
> db.student.deleteOne({"name":"malini"});
```

```
> db.student.find().pretty();
```

```
{  
  "_id" : ObjectId("603c0d13371e685b222cd0de"),  
  "name" : "gayathri",  
  "roll no" : 1,  
  "dept_name" : "Maths",  
  "college" : "holy cross college",  
}
```

```
"extra_curriclr" : "book reading"
}
{
  "_id" : ObjectId("603c0e74371e685b222cd0e0"),
  "name" : "Malathi",
  "roll no" : 3,
  "dept_name" : "MCA",
  "college" : "GTN college",
  "extra_curriclr" : "drawing"
}
```

9.React JS with MongoDB

Step 1: Create your own folder(name it as Mongo)

Step 2: Open VS Code and select your folder(Mongo) that you have created, from **File->Open Folder**.

Step 3: Then select the **Terminal->New terminal** and type the following command **>npx create-react-app frontend**.

Step 4: Add the extension **ES7+** in VS Code.

Step 5: Then go to terminal again and type the following commands:

>mkdir backend

>cd backend

>npm init

=>Then click on **Enter button** repeatedly to add **package.json** automatically.

Step 6: After that type the command **>npm i express** to install Express.

Step 7:Then type **>npm i cors** to determine what type of error has occurred.

Step 8: Then right click on  **Backend** (folder), which is at the left side, click open **NewFile** and name it as **app.js**

Step 9: Now write the following code in the **backend** and we have to change the **URL** of **MongoDB** in **backend**.

```
const express = require('express');
const cors = require('cors');
const app = express();
var bodyParser = require('body-parser');
var MongoClient = require('mongodb').MongoClient;
var url='mongodb://localhost:27017/?appName=demo&readPr
eference=primary';
//in the above line you will change your mongodb url
```

```
app.use(cors());
app.use(bodyParser.json());
app.post('/address', (req, res) => {
  MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("user");
    console.log("re", req.body)

    var myobj = { name: req.body.name, address: req.body.address };// db
value name.

    dbo.collection("login").insertOne(myobj, function(err, result) {
      if (err) throw err;
      console.log("1 document inserted");
      db.close();
      if (err) {
        // throw err;
        return res.status(400).json({
          status: false,
          description: "400 bad request"
        })
      }
      else {
        return res.status(200).json({
          status: true,
          description: result
        })
      }
    });
  })
});
```

```
}}
```

```
app.listen(8090, () => console.log('Assessment1 api runs on  
http://localhost:8090/'));
```

Step 10: In your explorer(i.e, left side menus) select **frontend** folder and select **src** folder. then you can see **App.js**, select it. Copy the following program in App.js

```
import React from 'react'  
import { useState } from 'react'  
function App() {  
  const [name,setname]=useState("")  
  const [password,setpassword]=useState("")  
  function Handle(){  
    console.log("data",name,password) ;  
    var myHeaders = new Headers();  
    myHeaders.append("Content-Type", "application/json");  
    var raw = JSON.stringify({  
      name:name,  
      address:password  
    });  
    var requestOptions = {  
      method: 'POST',  
      headers: myHeaders,  
      body: raw,  
      redirect: 'follow'  
    };  
    fetch("http://localhost:8090/address", requestOptions)  
      .then(response => response.json())
```

```

        .then(result => {
            console.log("res", result);
            if (result.status === true) {
                alert("register succefully!")
            }
            else {
                alert("your are not register suces")
            }
        })
        .catch(function(error){
            console.log("siginerror",error)
        })
    }
    return (
        <div className='App'>
            <input placeholder='email'  onChange={(e)=>setname(e.target.value)} />
            <input placeholder='password'
onChange={(e)=>setpassword(e.target.value)} />
            <button onClick={Handle}>submit</button>
        </div>
    )
}

export default App

```

Step 11: To change your **MongoDB URL**, go to **MongoDB compass**(if it is not available , then install it from Online).

Step 12: In MongoDB , click on **New Connection -> Advanced connection** option.

Step 13: Then select **Advanced tab** and then click **Read preference->Primary**.

Step 14: On **URI** options, set **Key=appname** and **Value=DEMO**
=>After that go up and **Copy the URL**,which is to be replaced in the **backend MongoDB**.

Step 15: Click **Save and Continue**.Then enter the name **DEMO**

Step 16: On the left bottom , there is + **Create database button**. Click it and enter the following ,
Database Name: user
Login Name: login

Step 17: Now go back to **VS Code** and change the **URL** in **backend**.

Step 18: Type the following command in the existing **Terminal**,

```
>cd backend  
>node app.js
```

Step 19: Open another **terminal** by clicking + on the **bottom right** of the **existing terminal**.

Step 20: In the **new Terminal** , type the following commands
>cd frontend
>npm start

OUTPUT:

