

WEEKLY REPORT

Work done in last week (Attach supporting Documents):

1. 27/05/2024 Monday

- 9.30am to 11.30 am: Create a top 5 Netflix series in 2024 website and use basic concepts

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import Card from './Components/Card';
4 import Sdata from './Sdata';
5 import './index.css';
6 //using simple method and Array
7 // ReactDOM.render(
8 //   <>
9 //     <Card sname={Sdata[0].sname}
10 //       imgsrc={Sdata[0].imgsrc}
11 //       title={Sdata[0].title}
12 //       links={Sdata[0].links}/>
13 //
14 //     <Card sname={Sdata[1].sname}
15 //       imgsrc={Sdata[1].imgsrc}
16 //       title={Sdata[1].title}
17 //       links={Sdata[1].links}/>
18 //
19 //     <Card sname={Sdata[2].sname}
20 //       imgsrc={Sdata[2].imgsrc}
21 //       title={Sdata[2].title}
22 //       links={Sdata[2].links}/>
23 //
24 //   </>, document.getElementById('root'))
25
```

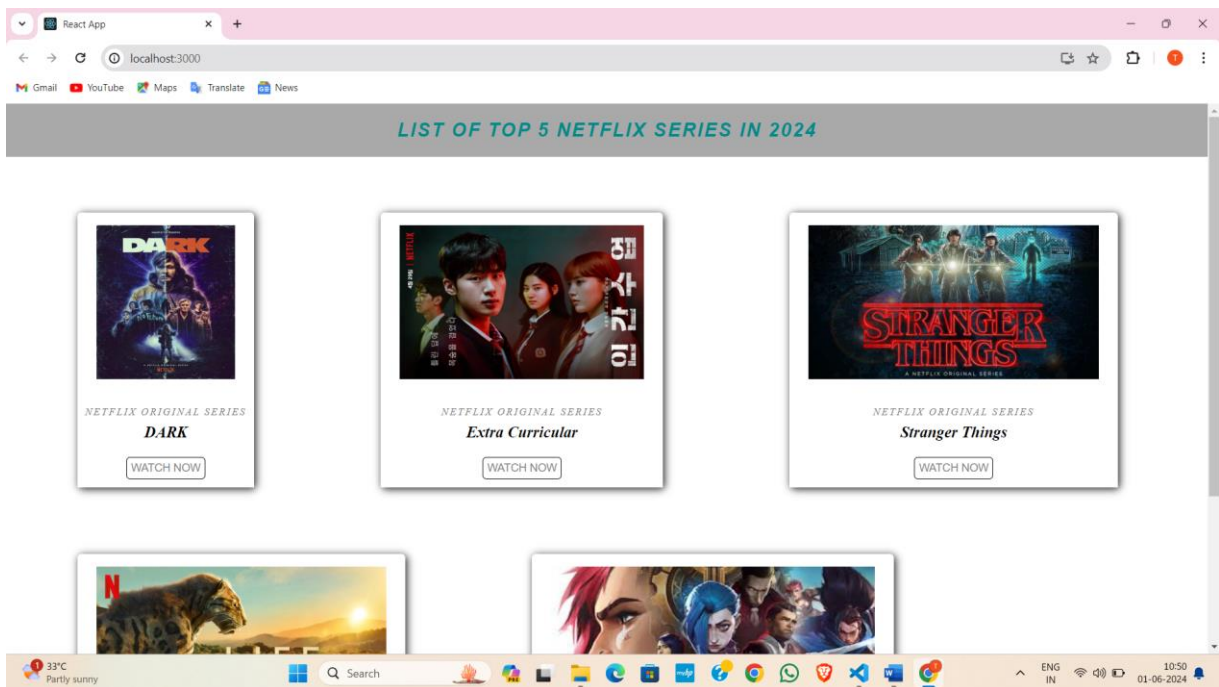
- 12.00pm to 2.30 pm: Replace simple method of Array with Mapping and add Movie and Solve error of image component

```
//using map method
ReactDOM.render(
  <>
    <h1 className="heading_style">List of top 5 Netflix Series in 2024</h1>
    {Sdata.map((val) => {
      return(
        <Card sname={val.sname}
          imgsrc={val.imgsrc}
          title={val.title}
          links={val.links}/>
      );
    })}
  </>, document.getElementById('root'))

```

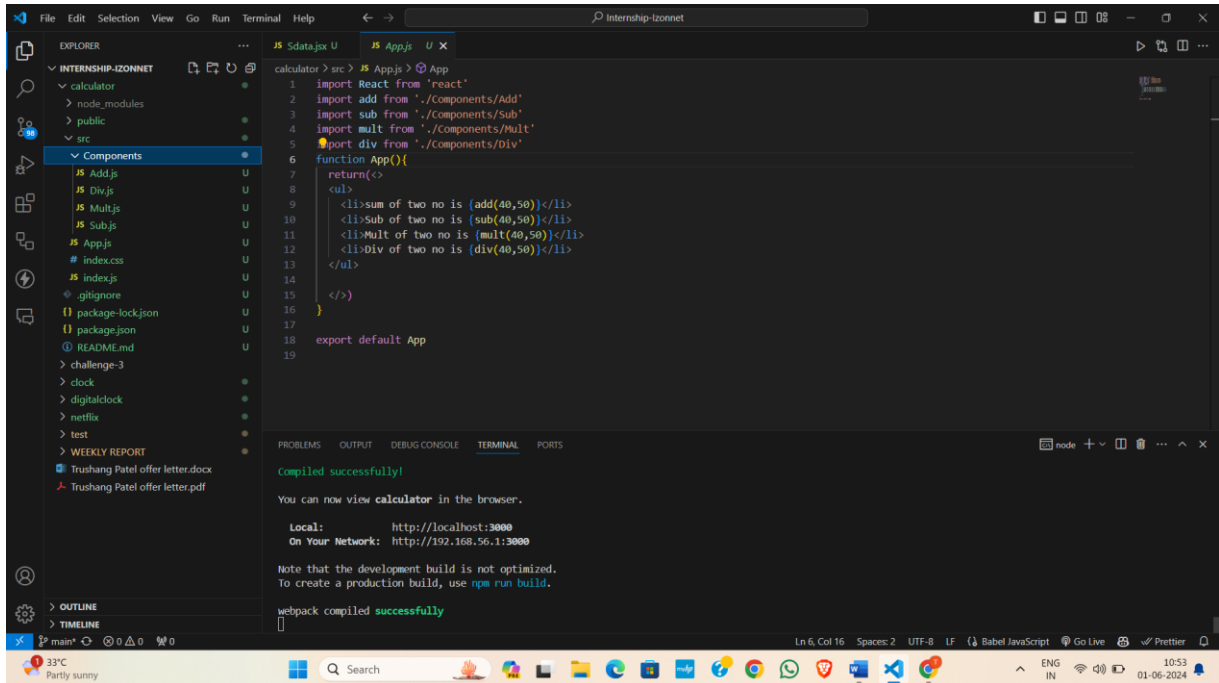
```
const Sdata =[
  {
    sname:"DARK",
    imgsrc:"https://wallpapercave.com/wp/wp4056410.jpg",
    title:"Netflix Original Series",
    links:"https://www.netflix.com/in/title/80100172"
  },
  {
    sname:"Extra curricular",
    imgsrc:"https://pearlandbubbles.wordpress.com/wp-content/uploads/2020/05/extracurricular.jpg",
    title:"Netflix Original Series",
    links:"https://www.netflix.com/in/title/80990668"
  },
  {
    sname:"Stranger Things",
    imgsrc:"https://wallpapercave.com/wp/wp1917154.jpg",
    title:"Netflix Original Series",
    links:"https://www.netflix.com/in/title/80057281"
  },
  {
    sname:"Our Planet",
    imgsrc:"https://occ-0-2794-2219.1.nflxso.net/dnm/api/v6/Qs00mKcPrvrk13HZAN5KwEL1kpE/AAAABaHbJMCHZIS3Wvqu58S5wX2303j8N2uZPrA"
  }
]
```

- 3.00 pm to 5.00 pm: Add CSS styling



2. 28/05/2024 Tuesday

- 9.30am to 11.30 am: Create a Calculator



```
calculator > src > JS App.js > App
1 import React from 'react'
2 import add from './Components/Add'
3 import sub from './Components/Sub'
4 import mult from './Components/Mult'
5 import div from './Components/Div'
6 function App(){
7   return(<
8     <ul>
9       <li>sum of two no is {add(40,50)}</li>
10      <li>Sub of two no is {sub(40,50)}</li>
11      <li>Mult of two no is {mult(40,50)}</li>
12      <li>Div of two no is {div(40,50)}</li>
13    </ul>
14  </>)
15 }
16 export default App
```

Compiled successfully!

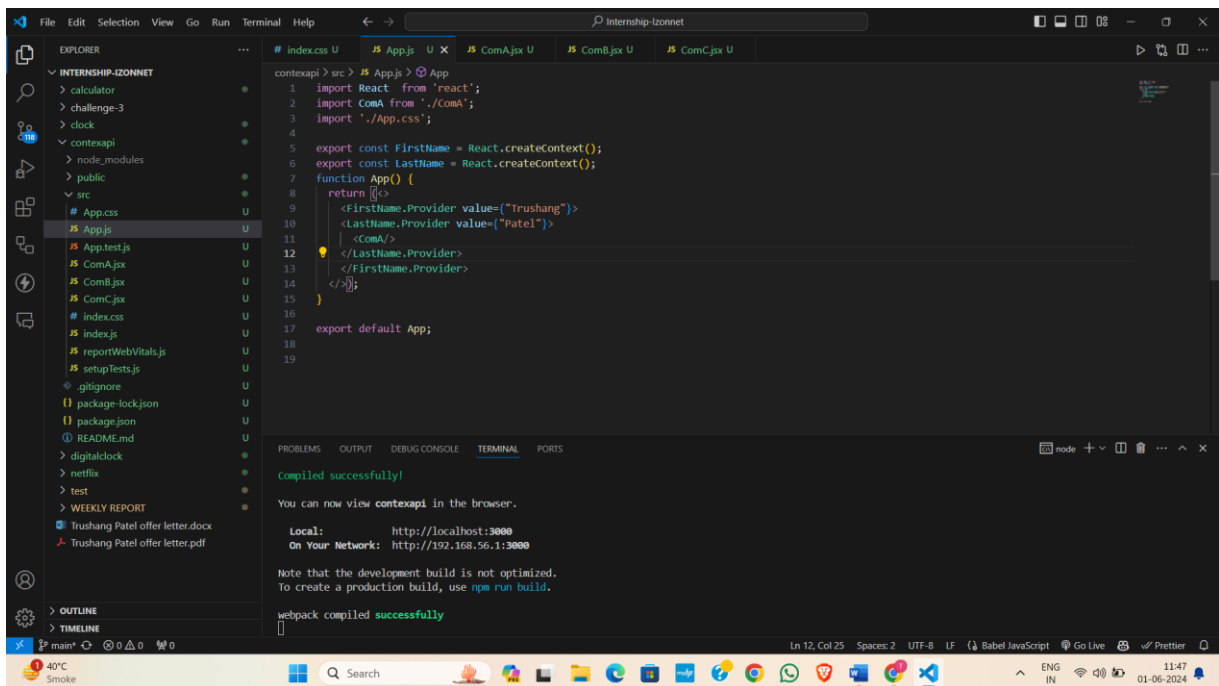
You can now view calculator in the browser.

Local: http://localhost:3000
On Your Network: http://192.168.56.1:3000

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

webpack compiled successfully

- 12.00pm to 2.30 pm: To learn about Context API



```
contextapi > src > JS App.js > App
1 import React from 'react';
2 import ComA from './ComA';
3 import './App.css';
4
5 export const FirstName = React.createContext();
6 export const LastName = React.createContext();
7 function App() {
8   return (
9     <FirstName.Provider value={"Trushang"}>
10     <LastName.Provider value={"Patel"}>
11       <ComA />
12     </LastName.Provider>
13     </FirstName.Provider>
14   </>);
15 }
16 export default App;
```

Compiled successfully!

You can now view contextapi in the browser.

Local: http://localhost:3000
On Your Network: http://192.168.56.1:3000

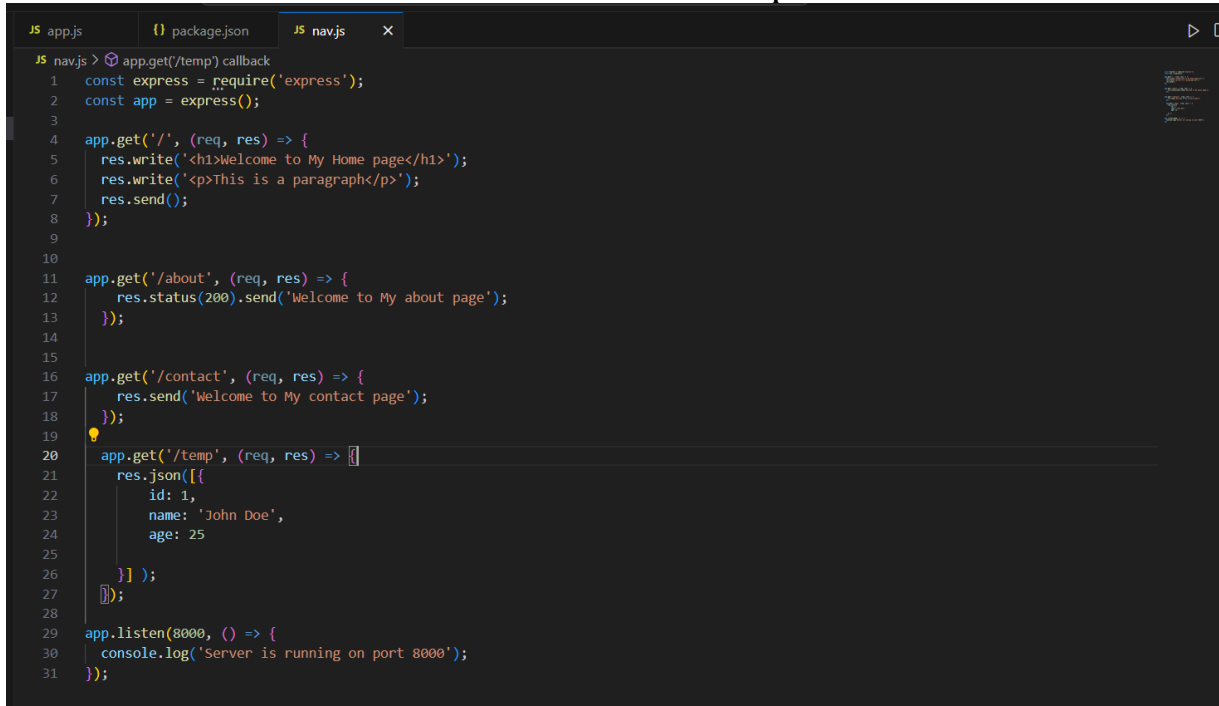
Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

- 3.00 pm to 5.00 pm: Introduction to Express JS and how to install express js

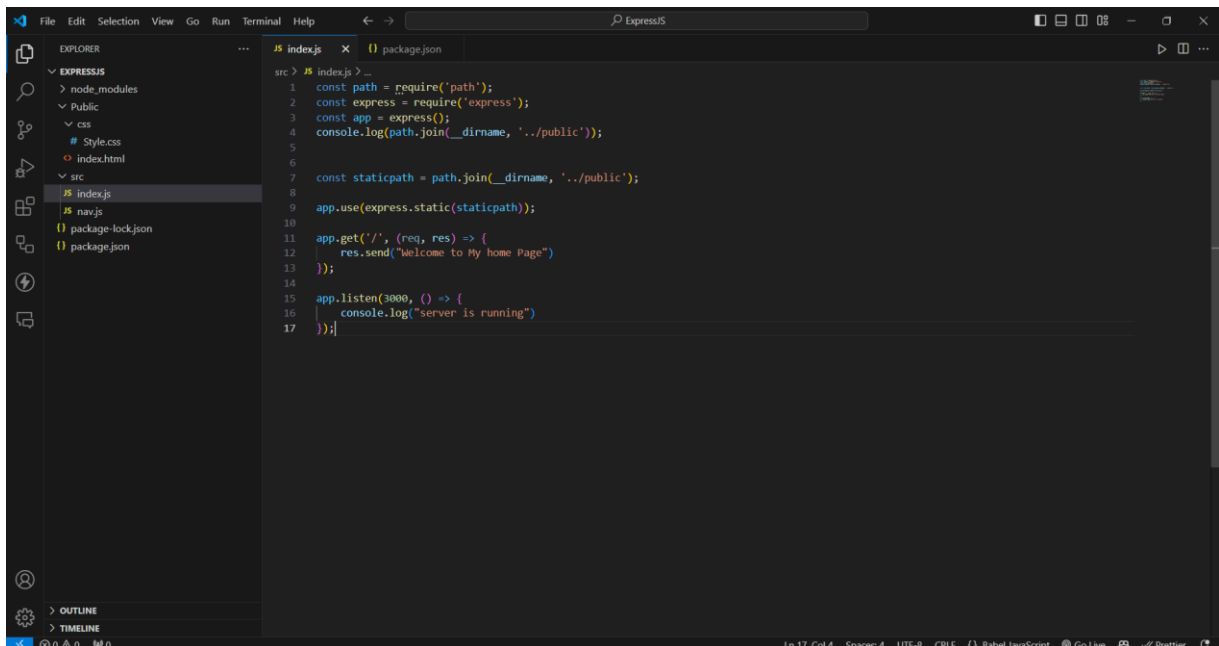
3. 29/05/2024 Wednesday

- 9.30 am to 11.30 am: Express JS routing and how to send HTML and JSON data as a Response




```
JS app.js | {} package.json | JS nav.js x
JS nav.js > app.get('/temp') callback
1  const express = require('express');
2  const app = express();
3
4  app.get('/', (req, res) => {
5    res.write('<h1>Welcome to My Home page</h1>');
6    res.write('<p>This is a paragraph</p>');
7    res.send();
8  });
9
10
11 app.get('/about', (req, res) => {
12   res.status(200).send('Welcome to My about page');
13 });
14
15
16 app.get('/contact', (req, res) => {
17   res.send('Welcome to My contact page');
18 });
19
20 app.get('/temp', (req, res) => {
21   res.json([
22     {
23       id: 1,
24       name: 'John Doe',
25       age: 25
26     }
27   ]);
28 }
29
30 app.listen(8000, () => {
31   console.log('Server is running on port 8000');
32 });
```

- 12.00pm to 2.30 pm: Serve HTML CSS & JS Files in Express JS | Middleware in Express JS



```
File Edit Selection View Go Run Terminal Help
EXPRESSJS
> node_modules
> Public
  > css
    # Style.css
    index.html
  > src
    # index.js
    # nav.js
    {} package-lock.json
    {} package.json
src > # index.js > ...
1  const path = require('path');
2  const express = require('express');
3  const app = express();
4  console.log(path.join(__dirname, '../public'));
5
6
7  const staticpath = path.join(__dirname, '../public');
8
9  app.use(express.static(staticpath));
10
11 app.get('/', (req, res) => {
12   res.send("Welcome to My home Page")
13 });
14
15 app.listen(3000, () => {
16   console.log("server is running")
17 });
```


- 3.00 pm to 5.00 pm: Template Engine (Pug, hbs, EJS)

mustache  4.2.0 • Public • Published 3 years ago

[Readme](#) [Code](#) [Beta](#) [0 Dependencies](#) [5,053 Dependents](#) [39 Versions](#)

mustache.js - Logic-less `{{mustache}}` templates with JavaScript

What could be more logical awesome than no logic at all?

 **Build Status**

mustache.js is a zero-dependency implementation of the **mustache** template system in JavaScript.

Mustache is a logic-less template syntax. It can be used for HTML, config files, source code - anything. It works by expanding tags in a template using values provided in a hash or object.

We call it "logic-less" because there are no if-statements, else-clauses, or for-loops. Instead there

Install

```
> npm i mustache
```

Repository

[github.com/janl/mustache.js](#)

Homepage

[github.com/janl/mustache.js](#)


Weekly Downloads

4,423,530

Version **License**

4. 30/05/2024 Thursday

- 9.30 am to 11.30 am: Customizing the Views Directory in Express JS and Using Partials

Express [Home](#) [Getting started](#) [Guide](#) [API reference](#) [Advanced topics](#) [Resources](#) 

		front-facing proxy, or series of proxies. The 'req.ips' property, then contains an array of IP addresses the client is connected through. To enable it, use the values described in the trust proxy options table . The 'trust proxy' setting is implemented using the proxy-addr package. For more information, see its documentation. NOTE: Sub-apps will inherit the value of this setting, even though it has a default value.	
views	String or Array	A directory or an array of directories for the application's views. If an array, the views are looked up in the order they occur in the array.	<code>process.cwd() + '/views'</code>
view cache	Boolean	Enables view template compilation caching. NOTE: Sub-apps will not inherit the value of this setting in production (when 'NODE_ENV' is 'production').	true in production, otherwise undefined.
view engine	String	The default engine extension to use when omitted. NOTE: Sub-apps will inherit the value of this setting.	N/A (undefined)
x-powered-by	Boolean	Enables the "X-Powered-By: Express" HTTP header.	true

Options for 'trust proxy' setting

Read [Express behind proxies](#) for more information.

- 12.00pm to 2.30 pm: Add 404 error page in Dynamic Website using Express JS, Adding Responsive Styles to Complete Dynamic Website using CSS in Express Website
- 3.00pm to 5.00 pm: Introduction to MongoDB

5. 31/05/2024 Friday

- 9.30 am to 11.30 am: Installation of MongoDB

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>"C:\Program Files\MongoDB\Server\7.0\bin\mongod.exe" --version
db version v7.0.5
Build Info: {
  "version": "7.0.5",
  "gitVersion": "7809d71e84e314b497f282ea8aa06d7ded3eb205",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
```

- 12.00pm to 2.30 pm: MongoDB CRUD operation and test various features
- 3.00pm to 5.00 pm: Install MongoDB GUI (MongoDB Compass) , how to connect MongoDB.

```
JS app.js > ...
const app = express();
console.log(path.join(__dirname, '../public'));

const staticpath = path.join(__dirname, '../public');
app.use(express.static(staticpath));

app.get('/', (req, res) => {
  res.send("Welcome to My home Page")
});

app.listen(3000, () => {
  console.log("server is running")
});

mongoose.connect('mongodb://localhost:27017/firstweb')
  .then(() => {console.log("connection successful")})
  .catch((err) => console.log(err));
```

Plans for next week:

1. To start a Project using MERN stack

References:

1. YouTube Channel: [Thapa Technical](#)
2. React Documentation
3. Express Documentation
4. Node Documentation
5. MongoDB Documentation

Signature of External Guide

Signature of Internal Guide

Student Id: 22IT116

Student Name: TRUSHANGKUMAR SUMANBHAI PATEL