

EX No	Name Of Experiment	Date
1	Personal CV	13-07-23
2	CSS Enabled CV	20-07-23
3	Form Making and Validation Using JS	27-03-23
4	Angular based App Development/Angular	09-08-23
5	React based App Development	15-08-23
6	Web Server Creation using NodeJS	21-09-23
7	Routing Implementation using ExpressJS	28-09-23
8	Building a REST API with Express, Node, and MongoDB	05-10-23

Ex. No: 1	PersonalCV
13.07.2023	

Aim:

To create a CV using basic HTML

Algorithm:

- 1.Create a HTML Page with title and create multiple sub headings <h1...h4>
- 2.Add color to each section
- 3.Include hyperlinks to the linked profile
- 4.Properly format and justify the text displayed
- 5.Ensure the CV is a Single Page

Program:

```
<html>
<title>Ajay Badrinath</title>
<h1 align="center"><font color="grey">Ajay Badrinath</font></h1>
<h5 align="center"><font color="grey">Computer Engineer</font></h5>
<hr color="grey"><br>
<div style="float: left; margin: 40px;">
<table cellpadding="10" cellspacing="10" border="0" width="100%">
<td align="10px">
<h4><font color="grey"> Contact</h4>
<h5> 7550149410</h5>
<h5> ajay21110103@snuchennai.edu.in</h5>
<h5> <a href="github.com/AjayBadrinath"> github Profile </a></h5>
<h5> Nungambakkam, Chennai 600034</h5>
</font>
</td>
</table>
<table cellpadding="10" cellspacing="10" border="0" width="100%">
<td align="10px">
<h4><font color="grey"> Skills</h4>
<h5> Web Development</h5>
<h5> Embeded System Programming</h5>
<h5> Java Development</h5>
<h5> DevOps Developer</h5>
</td>
</table>
<table cellpadding="10" cellspacing="10" border="0" width="100%">
<td align="10px">
<h4><font color="grey"> Degree Name : B.Tech </h4>
<h6> CSE-IoT</h6>
<h4> Institution : SNUChennai</h4>
<h5> Year: 2021-2025</h5>
<h6> Internet of Things specialisation course</h6>
```

<h4> Class 12 </h4>
<h6> HSE-CBSE</h6>
<h4> PSBBNgm</h4>
<h5> 2020-2021</h5>
</td>
</table>
<table cellpadding="10" cellspacing="10" border="0" width="100%">
<td align="10px">
<h4>Languages</h4>
<h5> C </h5>
<h5> x86 asm</h5>

<h5> Java</h5>
<h5> python</h5>
</td>
</table>
</div>
<div style="float:right;margin: 40px 400px;">
<table cellpadding="10" cellspacing="10" border="0" width="100%">
<td align="1000px">
<h4> Profile </h4>
<p>

I am a keen student with a
good coding background.I am currently
intersted
in Artificial Intelligence and
 Quantum Computing

</p>
<hr>

</td>
</table>
<table cellpadding="10" cellspacing="10" border="0" width="100%">
<td align="1000px">
<h4> Expertise/Projects </h4>
<p>

WebDesigning for a Startup

Backend Development

Algorithm Implementation

</p>
<hr>

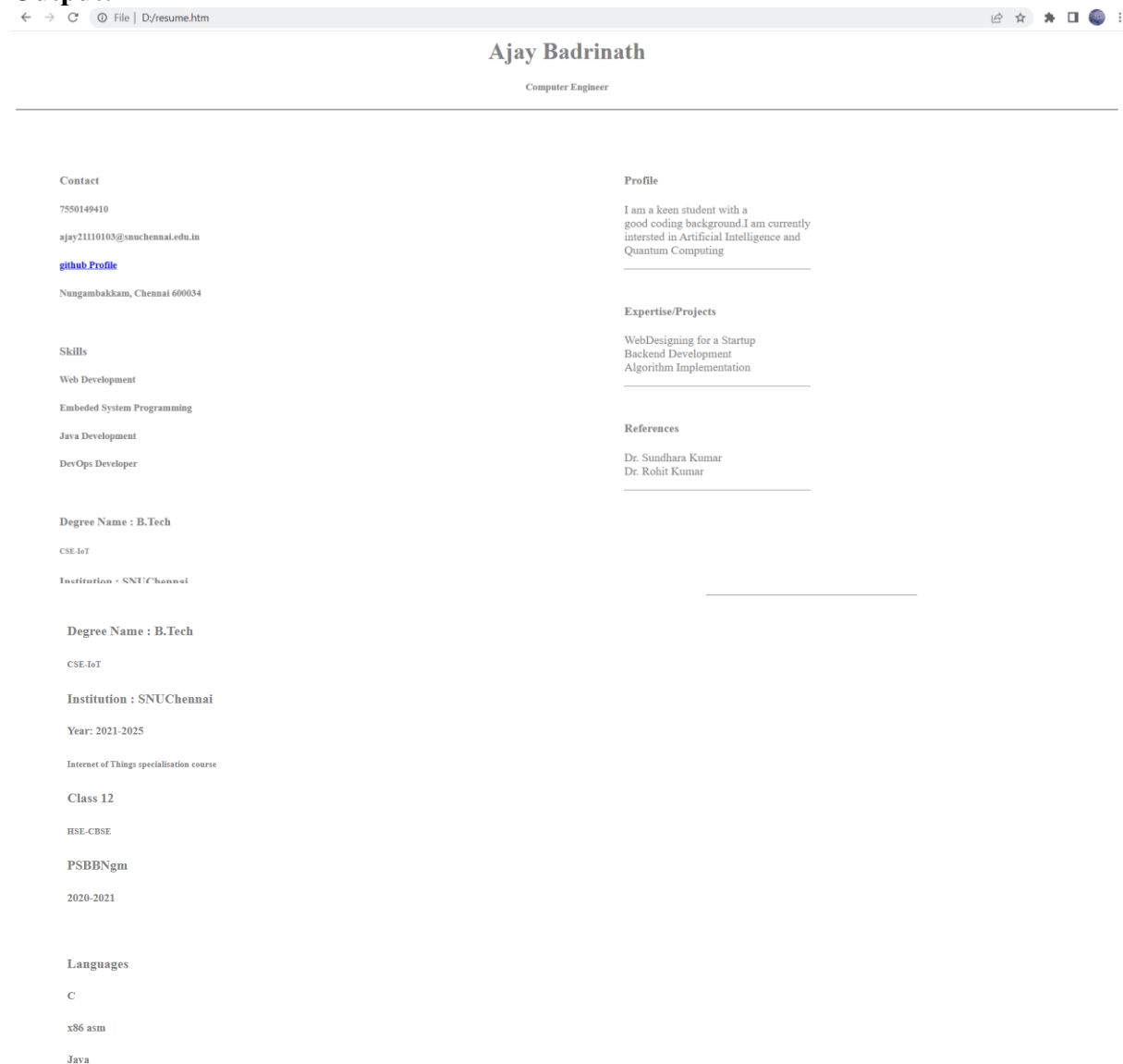
</td>
</table>
<table cellpadding="10" cellspacing="10" border="0" width="100%">
<td align="1000px">
<h4> References </h4>
<p>

Dr. Sundhara Kumar

Dr. Rohit Kumar

```
</p>
<hr>
</font>
</td>
</table>
</div>
</html>
```

Output:



Github Link: <https://github.com/AjayBadrinath/CS3809-WebTechLab/tree/main/Lab1-PersonalCV>

Result:

Thus Personal CV was created using basic HTML

Ex. No: 2	CSS Enabled CV
20.07.2023	

Aim:

To create CV using HTML and CSS

Algorithm:

1. Create a HTML Page with title and create multiple sub headings <h1...h4>
2. Add color to each section
3. Include hyperlinks to the linked profile
4. Include CSS properties to style the webpage
5. Style the text and use box-radius to create a profile pic
6. Use <div> to create two sides on a single CV
7. Properly format and justify the text displayed
8. Ensure the CV is a Single Page

Program:

HTML:

```
<html>
<body>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.2.1/css/fontawesome.min.css" integrity="sha384-
  QYIZto+st3yW+o8+5OHfT6S482Zsvz2WfOzpFSXMF9zqeLcFV0/wlZpMtyFcZALm"
  crossorigin="anonymous">
  <link rel="stylesheet" type="text/css" href="css/style.css">
  <div class="full">
  <div class="left">
    <br><br>
    <div class="imagedisplay">
      
    </div>
    <div class="info">
      <h2>Profile</h2>
      <hr class="hr_1">
      <p class="smallintro">
        A Passionate Student and athlete interested in
        computers and Tennis as a sport
      </p>
      <h4>
        Language
      </h4>
      <p class="langs">
        ANSI C
```

```
</p>
<div class="bar"></div>
<p class="langs">
  Python
</p>
<div class="bar1"></div>
<p class="langs">
  Java
</p>
<div class="bar2"></div>
<p class="langs">
  Assembly x86
</p>
<div class="bar3"></div>
<p class="langs">
  Javascript
</p>
<div class="bar3"></div>
<h4>
  Skils
</h4>
<p class="langs">
  Web Development
</p>
<div class="bars"></div>
<p class="langs">
  IoT Deployment
</p>
<div class="bar4"></div>
<p class="langs">
  Low Level Programming
</p>
<div class="bar5"></div>
<p class="langs">
  Data Analysis
</p>
<div class="bar6"></div>
<p class="langs">
  Machine Learning
</p>
<div class="bar7"></div>
<h4>Interests/Hobbies</h4>
<p class="langs">
  <ul><li>Tennis</li>
    <li>Coding</li>
  </ul>
</p>
</div>
<div class="clr"></div>
```

</div>

```
<div class="right">
  <h1 class="name">
    <span>AJAY</span><br>
    BADRINATH

  </h1>
  <p class="title_name">
    Computer Science Engineer (IoT)
  </p>
  <h2 class="Header">Education </h2>
  <hr class="hr_1"><br>
  <div class="clr"></div>
  <div class="year">
    <p class="yr">2021-2025</p>
  </div>
```

```

  <div class="work1">
    <p class="work">Shiv Nadar University</p>
    <br><br><br> <span class="whatdidido">Computer Science
Engineering (IoT)</span>
  </div>
  <div class="clr"></div><br>
  <div class="year">
    <p class="yr">2020-2021</p>
  </div>
```

```

  <div class="work1">
    <p class="work">Padma Seshadri Bala Bhavan Sr. Sec School</p><br>
    <br><br><br><span class="whatdidido">Class 12 HSE CBSE Examinations -
96%</span>
  </div>
  <div class="clr"></div><br>
  <div class="year">
    <p class="yr">2018-2019</p>
  </div>
```

```

  <div class="work1">
    <p class="work">Padma Seshadri Bala Bhavan Hr. Sec School</p><br>
    <br><br><br><span class="whatdidido">Class 10 SSE CBSE Examinations -
94%</span>
  </div>
```

```

  </div>
  <div class="clr"></div>
  <h2 class="Header">Projects </h2>
  <hr class="hr_1"><br>
  <div class="clr"></div>
  <div class="year">
    <p class="yr">2020-2021</p>
```

```
</div>

<div class ="work1">
  <p class ="work"><a href="https://github.com/AjayBadrinath/SmartAgriSystem">
Smart Agri system</a></p>
  <br><br><br> <span class="whatdidido">Smart Farming System with IoT Enabled
Monitoring and Action Planning</span>
</div>
<div class="clr"></div><br>
<div class="year">
  <p class="yr">2020-2020</p>
</div>

<div class ="work1">
  <p class ="work"><a href="https://github.com/AjayBadrinath/Zoom-bot"> Zoom
Bot </a><br>
  <br><span class="whatdidido">A Bot made in python that can attend online
classes automatically</span>
</div>
<div class="clr"></div><br>
<div class="year">
  <p class="yr">2020-2020</p>
</div>

<div class ="work1">
  <p class ="work"><a href="https://github.com/AjayBadrinath/Zoom-bot">
Password Manager </a><br>
  <br><span class="whatdidido">A Secure Dynamic Password Manager Made with
Java and vaadin Framework</span>
</div>
<div class="clr"></div>

</div>

<div class="clr"></div>

</div>

</body>
</html>
```

CSS:

```
.full{
  width: 700px;
  height: 1000px;
  background: #00fff2c1;
  margin: 50px auto;
  box-shadow: 5px 5px 5px 5px ;
```



```
}  
.left{  
    width: 30%;  
    float: left;  
    height: 600px;  
}  
.imagedisplay{  
    width: 100px;  
    height: 100px;  
    border: 2px solid rgba(0, 0, 0, 0.355);  
    border-radius: 50%;  
    padding: 2px;  
    margin: 20px auto;  
}  
.imagedisplay img{  
    width: 100%;  
    border-radius: 70%;  
}  
.info{  
    margin: 20px auto;  
    padding: 0px 40px 40px 40px;  
    color: green;  
}  
.hr_1{  
    background: white;  
}  
.smallintro{  
    letter-spacing: 3px;  
    font-size: small;  
    color: green;  
    font-family: 'Courier New', Courier, monospace;  
}  
.langs{  
    font-size: 10px;  
    letter-spacing: 1px;  
  
}  
.bar{  
    background:blue;  
    border-radius: 16px;  
    height: 12px;  
    width: 100%;  
}  
.bar1{  
    background:blue;  
    border-radius: 16px;  
    height: 12px;  
    width: 90%;  
}
```

```
.bar2{
  background:blue;
  border-radius: 16px;
  height: 12px;
  width: 80%;
}
.bar3{
  background:blue;
  border-radius: 16px;
  height: 12px;
  width: 60%;
}
.bars{
  background:blue;
  border-radius: 16px;
  height: 12px;
  width: 80%;
}
.work{
  float: left;
  width: 70%;
  color: orange;
}
.year{
  float: left;
  width: 35%;
}
.yr{
  font-weight: 500;
  color: orange;
}
.work1{
  float: left ;
  width: 65%;
}
.bar4{
  background:blue;
  border-radius: 16px;
  height: 12px;
  width: 90%;
}
.bar5{
  background:blue;
  border-radius: 16px;
  height: 12px;
  width: 90%;
}
.Header{
  color:rgba(176, 53, 232, 0.87)
}
```

```
.bar6{
    background:blue;
    border-radius: 16px;
    height: 12px;
    width: 85%;
}
.bar7{
    background:blue;
    border-radius: 16px;
    height: 12px;
    width: 75%;
}
.clr{
    clear: both;
}
.right{

    width: 60%;
    float:left;
    background-color: #ffffffb8;
    height: 900px;
    margin : 50px auto;
    border-radius: 50px 0px 0px 50px;
    padding :10px 10px;
    box-shadow: -7px 2px 15px 2px #00fff2c1;
}
.name{
    letter-spacing: 1px;
    font-weight: bolder;

    color: red;

}
.title_name{
    font-weight: 500;
    color: magenta;
    letter-spacing: 4px;
}
.whatdidido{

    font-size: 12px;
    color: darkgreen;
}
```

Output:



Profile

A Passionate Student and athlete interested in computers and Tennis as a sport

Language

ANSI C

Python

Java

Assembly x86

Javascript

Skills

Web Development

IoT Deployment

Low Level Programming

Data Analysis

Machine Learning

Interests/Hobbies

- Tennis
- Coding

AJAY BADRINATH

Computer Science Engineer (IoT)

Education

2021-2025	Shiv Nadar University	Computer Science Engineering (IoT)
2020-2021	Padma Seshadri Bala Bhavan Sr. Sec School	Class 12 HSE CBSE Examinations - 96%
2018-2019	Padma Seshadri Bala Bhavan Hr. Sec School	Class 10 SSE CBSE Examinations - 94%

Projects

2020-2021	<u>Smart Agri system</u>	Smart Farming System with IoT Enabled Monitoring and Action Planning
2020-2020	<u>Zoom Bot</u>	A Bot made in python that can attend online classes automatically
2020-2020	<u>Password Manager</u>	A Secure Dynamic Password Manager Made with Java and vaadin Framework

Github Link: <https://github.com/AjayBadrinath/CS3809-WebTechLab/tree/main/Lab2-CSS%20Enabled%20CV>

Result:

Thus Personal CV was created using HTML and CSS

Ex. No: 3	Form Making and Validation using JavaScript
27.07.2023	

Aim:

To Create a Form with usual form elements in JavaScript including the Alert(), Confirm(), and Response() functions. Additionally, validate the form elements.

Algorithm:

1. Create a html page that has essential form components like text box password radio button
2. Style the form using css
3. Using inline javascript validate the form using alert() confirm () functions]
4. Once the form has been validated submit/ clear the form

Program:

HTML:

```
<html>
<body>
  <link rel="stylesheet" href="style2.css">

  <script>confirm("Enter the form")</script>
  <script>
    function submit(){
      var name=document.querySelector('.c');
      confirm("Submit?" +name);
    }
  </script>

  </script>
<div class ="align">
  <form class="form">
    <div style="float: left;margin: 40px;" >
      <span><label >Name:</label>
      <div class="spacing">
        <input type="text" id="c" name="Name" value="Enter" minlength="3"
maxlength="12">

      </div>
      <span><label >Email:</label>
      <div class="spacing">
        <input type="email" id="c3" name="Email" value="Enter" required>

      </div>
      <span><label >Password:</label>
```

```
<div class="spacing">
  <input type="password" id="c2" name="Pwd" value="" minlength="8"
pattern="^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?!.*\s).*$" title="Please include at least 1 uppercase
character, 1 lowercase character, and 1 number." required >
```

```
</div>
<!--
add pseudo elem hover effect in button and shadow and size of btn
-->
</span>
<span>
  <label for="Gender">Gender:</label>
  <div class="spacing">
    <label for="Gender">Male</label>
    <input type="radio" id="gen">
    <label for="Gender">Female</label>
    <input type="radio" id="gen"><br></div>
  </span>
  <label for="Qualification">Qualification:</label>
  <input type="checkbox" id="q">
  <label for="Qualification">10th</label>
  <input type="checkbox" id="q">
  <label for="Qualification">12th</label><br>
  <div class="spacing">
    <input type="checkbox" id="q">
    <label for="Qualification">Diploma</label>

    <input type="checkbox" id="q">
    <label for="Qualification">PostGrad</label><br>
    <label for="cars">Choose</label>
    <select id="cars" name="cars" required>
      <option value="volvo">p1</option>
      <option value="saab">p2</option>
      <option value="fiat">p3</option>
      <option value="audi">p4</option>
    </select><br>
    <button class="btn1" onclick="submit()">Submit</button>
    <button class="btn2">Cancel</button>
  </div>
</div>
</form>
</div>
```

```
</body>
```

```
</html>
```

CSS:

```
.spacing{
```

```
margin-left:110px;
```

```
}  
.error  
{  
  color: red;  
  size: 80%  
}  
.hidden  
{  
  display:none;  
}  
  
.align{  
  margin-left: 500px;  
  margin-top: 200px;  
  width: 500px;  
  height: 400px;  
  border-radius: 10px;  
  box-shadow: 5px 10px rebeccapurple;  
  background: rgb(2,0,36);  
  background: rgb(2,0,36);  
  background: -moz-linear-gradient(90deg, rgba(2,0,36,1) 0%,  
rgba(35,212,28,0.7036064425770308) 0%, rgba(9,211,198,1) 0%, rgba(22,212,113,1) 6%,  
rgba(22,212,113,1) 6%, rgba(0,211,255,1) 16%, rgba(15,212,158,1) 21%, rgba(15,212,158,1)  
28%, rgba(7,212,208,1) 35%, rgba(3,212,234,1) 40%, rgba(0,212,255,1) 44%,  
rgba(0,212,255,1) 49%, rgba(0,212,255,1) 54%, rgba(0,212,255,1) 64%, rgba(24,212,99,1)  
76%);  
  background: -webkit-linear-gradient(90deg, rgba(2,0,36,1) 0%,  
rgba(35,212,28,0.7036064425770308) 0%, rgba(9,211,198,1) 0%, rgba(22,212,113,1) 6%,  
rgba(22,212,113,1) 6%, rgba(0,211,255,1) 16%, rgba(15,212,158,1) 21%, rgba(15,212,158,1)  
28%, rgba(7,212,208,1) 35%, rgba(3,212,234,1) 40%, rgba(0,212,255,1) 44%,  
rgba(0,212,255,1) 49%, rgba(0,212,255,1) 54%, rgba(0,212,255,1) 64%, rgba(24,212,99,1)  
76%);  
  background: linear-gradient(90deg, rgba(2,0,36,1) 0%,  
rgba(35,212,28,0.7036064425770308) 0%, rgba(9,211,198,1) 0%, rgba(22,212,113,1) 6%,  
rgba(22,212,113,1) 6%, rgba(0,211,255,1) 16%, rgba(15,212,158,1) 21%, rgba(15,212,158,1)  
28%, rgba(7,212,208,1) 35%, rgba(3,212,234,1) 40%, rgba(0,212,255,1) 44%,  
rgba(0,212,255,1) 49%, rgba(0,212,255,1) 54%, rgba(0,212,255,1) 64%, rgba(24,212,99,1)  
76%);  
  filter:  
progid:DXImageTransform.Microsoft.gradient(startColorstr="#020024",endColorstr="#18d4  
63",GradientType=1);  
}  
.btn1 {  
  background-color: blueviolet;  
  padding: 20px 30px;  
  font-size: medium;  
  border-radius: 10px;  
}  
.btn2{
```

```
background-color: green;
padding: 20px 30px;
font-size: medium;
border-radius: 10px;
}
.btn1:hover{
background-color: gold;
color: blue;
padding: 30px 40px;
}
.btn2:hover{
background-color: red;
color: blue;
padding: 30px 40px;
}
```

Output:



Github Link: <https://github.com/AjayBadrinath/CS3809-WebTechLab/tree/main/Lab3-Form%20Making%20and%20Validation%20Using%20JS>

Result:

Thus Form Validation was done using Javascript using alert() confirm() and response() functions.

Ex. No: 4	Angular based App creation
09.08.2023	

Aim:

To Create an App using ANGULAR with Components, Binding, and Services usage.

Algorithm:

- 1.Create angular app
2. once created create components corresponding to home-component
- 3.inside the template write the required html dom to be rendered
- 4.add Dependency injection
- 5.Add routing if needed for the app
6. use ng serve to serve the app on localhost

Program:

Home-component.css;

```
.results {  
  display: grid;  
  column-gap: 14px;  
  row-gap: 14px;  
  grid-template-columns: repeat(auto-fill, minmax(400px, 400px));  
  margin-top: 50px;  
  justify-content: space-around;  
}
```

```
input[type="text"] {  
  border: solid 1px var(--primary-color);  
  padding: 10px;  
  border-radius: 8px;  
  margin-right: 4px;  
  display: inline-block;  
  width: 30%;  
}
```

```
button {  
  padding: 10px;  
  border: solid 1px var(--primary-color);  
  background: var(--primary-color);  
  color: white;  
  border-radius: 8px;  
}
```

```
@media (min-width: 500px) and (max-width: 768px) {  
  .results {  
    grid-template-columns: repeat(2, 1fr);  
  }  
  input[type="text"] {
```

```
    width: 70%;  
  }  
}  
  
@media (max-width: 499px) {  
  .results {  
    grid-template-columns: 1fr;  
  }  
}
```

Home-component.ts

```
import { Component, inject } from '@angular/core';  
import { CommonModule } from '@angular/common';  
import { HousingLocationComponent } from '../housing-location/housing-  
location.component';  
import { HousingLocation } from '../housinglocation';  
import { HousingService } from '../housing.service';  
@Component({  
  selector: 'app-home',  
  standalone: true,  
  imports: [CommonModule, HousingLocationComponent],  
  template: `  
    <section>  
      <form>  
        <input type="text" placeholder="Filter by city">  
        <button class="primary" type="button">Search</button>  
      </form>  
    </section>  
    <section class="results">  
      <app-housing-location *ngFor="let housingLocation of housingLocationList"  
      [housingLocation]="housingLocation">></app-housing-location>  
    </section>  
  `,  
  styleUrls: ['./home.component.css']  
})  
export class HomeComponent {  
  readonly baseUrl = 'https://angular.io/assets/images/tutorials/faa';  
  housingLocationList: HousingLocation[] = [];  
  housingService: HousingService = inject(HousingService);  
  
  constructor() {  
    this.housingLocationList = this.housingService.getAllHousingLocations();  
  }  
}
```

Housing-location-component.css

```
.listing {  
  background: var(--accent-color);
```

```
border-radius: 30px;
padding-bottom: 30px;
}
.listing-heading {
color: var(--primary-color);
padding: 10px 20px 0 20px;
}
.listing-photo {
height: 250px;
width: 100%;
object-fit: cover;
border-radius: 30px 30px 0 0;
}
.listing-location {
padding: 10px 20px 20px 20px;
}
.listing-location::before {
content: url("/assets/location-pin.svg") / "";
}

section.listing a {
padding-left: 20px;
text-decoration: none;
color: var(--primary-color);
}
section.listing a::after {
content: "\203A";
margin-left: 5px;
}
```

Housing-location-component.ts

```
import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';
import { Input } from '@angular/core';
import { HousingLocation } from '../housinglocation';
import { RouterLink, RouterOutlet } from '@angular/router';
@Component({
  selector: 'app-housing-location',
  standalone: true,
  imports: [CommonModule, RouterLink, RouterOutlet],
  template: `
    <section class="listing">
      <img class="listing-photo" [src]="housingLocation.photo" alt="Exterior photo of
    { {housingLocation.name} }">
      <h2 class="listing-heading">{{ housingLocation.name }}</h2>
      <p class="listing-location">{{ housingLocation.city }}, { {housingLocation.state } }</p>
      <a [routerLink]="['/details', housingLocation.id]">Learn More</a>
    </section>
  `,
  styleUrls: ['./housing-location.component.css']
})
```

```
  })  
  export class HousingLocationComponent {  
    @Input() housingLocation!: HousingLocation;  
  
  }
```

App-component.css

```
:host {  
  --content-padding: 10px;  
}  
header {  
  display: block;  
  height: 60px;  
  padding: var(--content-padding);  
  box-shadow: 0px 5px 25px var(--shadow-color);  
}  
.content {  
  padding: var(--content-padding);  
}
```

App-component.ts

```
import { Component } from '@angular/core';  
import { HomeComponent } from './home/home.component';  
import { RouterModule } from '@angular/router';  
@Component({  
  selector: 'app-root',  
  standalone: true,  
  imports: [HomeComponent,  
    RouterModule,  
  ],  
  template: `<<main>  
    <a [routerLink]="['/']">  
      <header class="brand-name">  
          
      </header>  
    </a>  
    <section class="content">  
      <router-outlet></router-outlet>  
    </section>  
  </main>`,  
  styleUrls: ['./app.component.css'],  
})  
export class AppComponent {  
  title = 'Hi There';  
}
```

Housing-service.ts

```
import { Injectable } from '@angular/core';
import { HousingLocation } from './housinglocation';
@Injectable({
  providedIn: 'root'
})
export class HousingService {
  readonly baseUrl = 'https://angular.io/assets/images/tutorials/faa';

  housingLocationList: HousingLocation[] = [
    {
      id: 0,
      name: 'Acme Fresh Start Housing',
      city: 'Chicago',
      state: 'IL',
      photo: `${this.baseUrl}/bernard-hermant-CLKGGwIBTaY-unsplash.jpg`,
      availableUnits: 4,
      wifi: true,
      laundry: true
    },
    {
      id: 1,
      name: 'A113 Transitional Housing',
      city: 'Santa Monica',
      state: 'CA',
      photo: `${this.baseUrl}/brandon-griggs-wR11KBaB86U-unsplash.jpg`,
      availableUnits: 0,
      wifi: false,
      laundry: true
    },
    {
      id: 2,
      name: 'Warm Beds Housing Support',
      city: 'Juneau',
      state: 'AK',
      photo: `${this.baseUrl}/i-do-nothing-but-love-lAyXdl1-Wmc-unsplash.jpg`,
      availableUnits: 1,
      wifi: false,
      laundry: false
    },
    {
      id: 3,
      name: 'Homesteady Housing',
      city: 'Chicago',
      state: 'IL',
      photo: `${this.baseUrl}/ian-macdonald-W8z6aiwfi1E-unsplash.jpg`,
      availableUnits: 1,
      wifi: true,
      laundry: false
    },
  ]
}
```

```
id: 4,
name: 'Happy Homes Group',
city: 'Gary',
state: 'IN',
photo: `${this.baseUrl}/krzysztof-hepner-978RAXoXnH4-unsplash.jpg`,
availableUnits: 1,
wifi: true,
laundry: false
},
{
  id: 5,
  name: 'Hopeful Apartment Group',
  city: 'Oakland',
  state: 'CA',
  photo: `${this.baseUrl}/r-architecture-JvQ0Q5IkeMM-unsplash.jpg`,
  availableUnits: 2,
  wifi: true,
  laundry: true
},
{
  id: 6,
  name: 'Seriously Safe Towns',
  city: 'Oakland',
  state: 'CA',
  photo: `${this.baseUrl}/phil-hearing-IYfp2Ixe9nM-unsplash.jpg`,
  availableUnits: 5,
  wifi: true,
  laundry: true
},
{
  id: 7,
  name: 'Hopeful Housing Solutions',
  city: 'Oakland',
  state: 'CA',
  photo: `${this.baseUrl}/r-architecture-GGupkreKwxA-unsplash.jpg`,
  availableUnits: 2,
  wifi: true,
  laundry: true
},
{
  id: 8,
  name: 'Seriously Safe Towns',
  city: 'Oakland',
  state: 'CA',
  photo: `${this.baseUrl}/saru-robert-9rP3mxf8qWI-unsplash.jpg`,
  availableUnits: 10,
  wifi: false,
  laundry: false
},
{
```

```
    id: 9,  
    name: 'Capital Safe Towns',  
    city: 'Portland',  
    state: 'OR',  
    photo: `${this.baseUrl}/webaliser-_TPTXZd9mOo-unsplash.jpg`,  
    availableUnits: 6,  
    wifi: true,  
    laundry: true  
  }  
];  
getAllHousingLocations(): HousingLocation[] {  
  return this.housingLocationList;  
}  
  
getHousingLocationById(id: number): HousingLocation | undefined {  
  return this.housingLocationList.find(housingLocation => housingLocation.id === id);  
}  
  
constructor() { }  
}
```

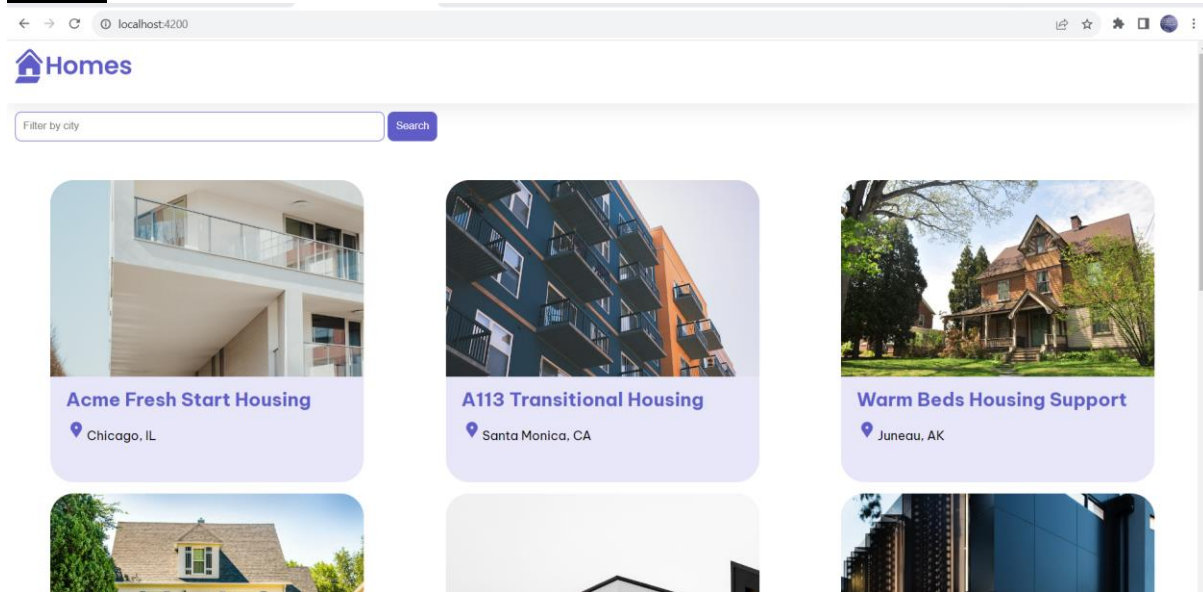
Housing-location.ts

```
export interface HousingLocation {  
  id: number;  
  name: string;  
  city: string;  
  state: string;  
  photo: string;  
  availableUnits: number;  
  wifi: boolean;  
  laundry: boolean;  
}
```

Routes.ts

```
import { Routes } from '@angular/router'  
import { HomeComponent } from './home/home.component'  
import { DetailsComponent } from './details/details.component'  
const routeConfig: Routes=[  
  {  
    path:"",  
    component:HomeComponent,  
    title:'Home Page'  
  },  
  {  
    path: 'details/:id',  
    component: DetailsComponent,  
    title: 'Home details'  
  }  
];  
export default routeConfig;
```

Output:



Github Link: <https://github.com/AjayBadrinath/CS3809-WebTechLab/tree/main/Lab4-Angular%20based%20App%20Development>

Result:

Thus Angular app was created with Components, Binding, and Services.

Ex. No: 5	React based App Development
15.08.2023	

Aim:

To Create an App using React with Components, Rendering, and Data Sharing.

Algorithm:

1. Create React app using npm create react app
2. Once created edit the index.html page to add title
3. Render individual squares and add functions to render player move(x/o)
4. Use React hooks to keep track of state of the game
5. Create another function to evaluate winner and clear the state

Program:

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <div class="hi">
      <h1>Tic - Tac - Toe</h1>
      <div id="root"></div>word
    </div>
  </body>
</html>
```

Styles.css

```
* {
  box-sizing: border-box;
}
.hi {
  background-color: blueviolet;
  margin: 100px 100px 100px 200px;
  margin-right: 100px;
  box-sizing: 40px;
  padding: 40px 40px 40px 40px;
  width: 250px;
  border-radius: 5px;
  animation-name: anim;
  animation-duration: 10s;
  animation-iteration-count: infinite;
}
@keyframes anim {
  0% {
```

```
    background-color: green;
}
25% {
    background-color: orange;
}
50% {
    background-color: blue;
}
75% {
    background-color: red;
}
100% {
    background-color: lightpink;
}
}
body {
    font-family: sans-serif;
    margin: 20px;
    padding: 0;
}

h1 {
    margin-top: 0;
    font-size: 22px;
}

h2 {
    margin-top: 0;
    font-size: 20px;
}

h3 {
    margin-top: 0;
    font-size: 18px;
}

h4 {
    margin-top: 0;
    font-size: 16px;
}

h5 {
    margin-top: 0;
    font-size: 14px;
}

h6 {
    margin-top: 0;
    font-size: 12px;
}
```

```
code {
  font-size: 1.2em;
}

ul {
  padding-inline-start: 20px;
}

* {
  box-sizing: border-box;
}

body {
  font-family: sans-serif;
  margin: 20px;
  padding: 0;
}

.square {
  background: indigo;
  border-radius: 20px;
  border: 1px solid #999;
  float: left;
  font-size: 24px;
  font-weight: bold;
  line-height: 34px;
  height: 34px;
  margin-right: -1px;
  margin-top: -1px;
  padding: 0;
  text-align: center;
  width: 34px;
}

.board-row:after {
  clear: both;
  content: "";
  display: table;
}

.status {
  margin-bottom: 10px;
}

.game {
  display: flex;
  flex-direction: row;
}

.game-info {
```

```
margin-left: 20px;
}
App.js
import { useState } from 'react';

function Square({ value, onSquareClick }) {
  return (
    <button className="square" onClick={onSquareClick}>
      {value}
    </button>
  );
}

export default function Board() {
  const [xIsNext, setXIsNext] = useState(true);
  const [squares, setSquares] = useState(Array(9).fill(null));

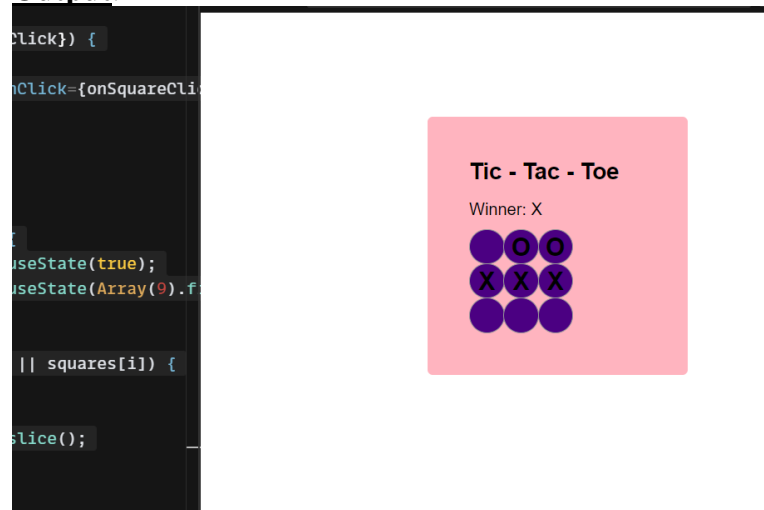
  function handleClick(i) {
    if (calculateWinner(squares) || squares[i]) {
      return;
    }
    const nextSquares = squares.slice();
    if (xIsNext) {
      nextSquares[i] = 'X';
    } else {
      nextSquares[i] = 'O';
    }
    setSquares(nextSquares);
    setXIsNext(!xIsNext);
  }

  const winner = calculateWinner(squares);
  let status;
  if (winner) {
    status = 'Winner: ' + winner;
  } else {
    status = 'Next player: ' + (xIsNext ? 'X' : 'O');
  }

  return (
    <>
    <div className="status">{status}</div>
    <div className="board-row">
      <Square value={squares[0]} onSquareClick={() => handleClick(0)} />
      <Square value={squares[1]} onSquareClick={() => handleClick(1)} />
      <Square value={squares[2]} onSquareClick={() => handleClick(2)} />
    </div>
    <div className="board-row">
      <Square value={squares[3]} onSquareClick={() => handleClick(3)} />
      <Square value={squares[4]} onSquareClick={() => handleClick(4)} />
    </div>
  </>
);
}
```

```
    <Square value={squares[5]} onSquareClick={() => handleClick(5)} />
  </div>
  <div className="board-row">
    <Square value={squares[6]} onSquareClick={() => handleClick(6)} />
    <Square value={squares[7]} onSquareClick={() => handleClick(7)} />
    <Square value={squares[8]} onSquareClick={() => handleClick(8)} />
  </div>
</>
);
}
function calculateWinner(squares) {
  const lines = [
    [0, 1, 2],
    [3, 4, 5],
    [6, 7, 8],
    [0, 3, 6],
    [1, 4, 7],
    [2, 5, 8],
    [0, 4, 8],
    [2, 4, 6],
  ];
  for (let i = 0; i < lines.length; i++) {
    const [a, b, c] = lines[i];
    if (squares[a] && squares[a] === squares[b] && squares[a] === squares[c]) {
      return squares[a];
    }
  }
  return null;
}
```

Output:



Github Link: <https://github.com/AjayBadrinath/CS3809-WebTechLab/tree/main/Lab5-React%20based%20App%20Development/react>

Result:

Thus React App was created. with Components, Rendering, and Data Sharing

Ex. No: 6	Web Server Creation using NodeJS
21.09.2023	

Aim:

To Create a Web Server offering basic web service(s) to the front-end.

Algorithm:

- 1.Create a listener object to listen for requests
- 2.Once created open the file to be served using readFile and set the http header
3. Write out contents and return status of 200

Program:

```
const http = require("http");

const host = 'localhost';
const port = 8000;
const fs = require('fs').promises;
const requestListener = function (req, res) {
  fs.readFile("D:\\webdev\\cssresume.html").then(contents=>{
    res.setHeader("Content-Type", "text/html");
    res.writeHead(200);
    res.end(contents);
  }).catch(err => {
    res.writeHead(400);
    // res.end(err);
    return;
  });
};

const server = http.createServer(requestListener);
server.listen(port, host, () => {
  console.log(`Server is running on http://${host}:${port}`);
});
```

Output:

Profile

A Passionate Student and athlete interested in computers and Tennis as a sport

Language

ANSI C

Python

Java

Assembly x86

Javascript

Skills

Web Development

IoT Deployment

Low Level Programming

Data Analysis

Machine Learning

Interests/Hobbies

Github Link: <https://github.com/AjayBadrinath/CS3809-WebTechLab/blob/main/Lab6-Web%20Server%20Creation%20using%20NodeJS/node/html.js>

Result:

Thus a Web Server offering basic web service(s) to the front-end was created.

Ex. No: 7	Routing Implementation using ExpressJS
28.09.2023	

Aim:

To Implement the routing feature(s) using the ExpressJS

Algorithm:

1. Import express in the js file
2. Create a new instance of express router
3. Use router get function to navigate to some route and render a html page
4. In order to add routes use app.set() and app.use to use the routes
5. In case of any error render the error page

Program:

Index.js

```
var express = require('express');
var router = express.Router();
/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express' });
  res.send("Express is awesome");
});
module.exports = router;
routes/user.js
var express = require('express');
var router = express.Router();
/* GET users listing. */
router.get('/', function(req, res, next) {
  res.send('respond with a resource');
});
module.exports = router;
```

routes/testAPI.js

```
var express = require("express");
var router = express.Router();
router.get("/",function(req, res, next) {
  res.send("API is working properly");
});
module.exports = router;
```

app.js

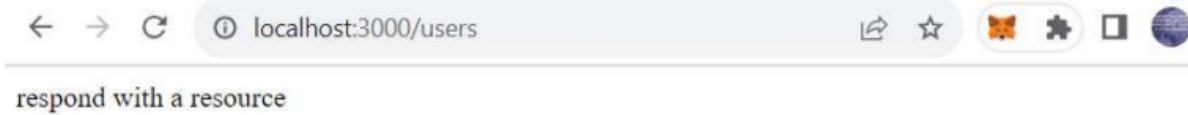
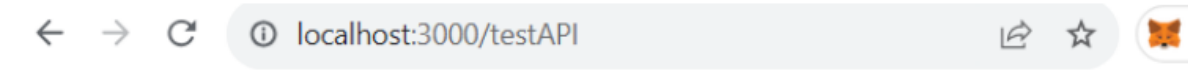
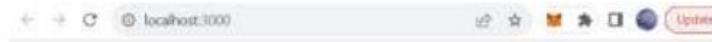
```
var createError = require('http-errors');
var express = require('express');
```



```
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');
var cors=require('cors');
var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');
var testAPIRouter=require('./routes/testAPI');

var index=require('./routes/index');
var s1=require('something')
var app = express();
// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');
app.use(logger('dev'));
app.use(express.json());
app.use(cors());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));
app.use("/index",index);
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use("/testAPI",testAPIRouter);
// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});
// error handler
app.use(function(err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err : {};
  // render the error page
  res.status(err.status || 500);
  res.render('error');
});
module.exports = app;
```

Output:



Github Link: <https://github.com/AjayBadrinath/CS3809-WebTechLab/tree/main/Lab7-Routing%20Implementation%20using%20ExpressJS/express/api/routes>

Result:

Thus routing features were implemented using Express JS.

Ex. No: 8	Building a REST API with Express, Node, and MongoDB
05.10.2023	

Aim:

To Build a REST API using EJS ,NODE,MongoDB

Algorithm:

- 1.Import express and mongoose in index.js
2. use json as the format and establish connection using mongoose and atlas
- 3.once connection is established implement the GET ,PUT POST methods
- 4.test out different API Methods to ensure proper working

Program:

Index.js

```
const express = require('express');
const mongoose = require('mongoose');
const app=express();
app.use(express.json);
app.listen(3000,()=>{
  console.log("OK");
})
require('dotenv').config();
const mongoString = process.env.DATABASE_URL
mongoose.connect(mongoString);
const database = mongoose.connection
database.on('error', (error) => {
  console.log(error)
})
database.once('connected', () => {
  console.log('Database Connected');
})
const routes = require('./routes/routes');
app.use('/api', routes)
route.js
const express = require('express');
const router = express.Router()
module.exports = router;
//Post Method
router.post('/post', (req, res) => {
  res.send('Post API')
})
//Get all Method
router.get('/getAll', (req, res) => {
```

```
res.send('Get All API')
})
//Get by ID Method
router.get('/getOne/:id', (req, res) => {
res.send('Get by ID API')
})

//Update by ID Method
router.patch('/update/:id', (req, res) => {
res.send('Update by ID API')
})
//Delete by ID Method
router.delete('/delete/:id', (req, res) => {
res.send('Delete by ID API')
})
const Model = require('../models/model');
//Post Method
router.post('/post', (req, res) => {
const data = new Model({
name: req.body.name,
age: req.body.age
})
try {
const dataToSave = data.save();
res.status(200).json(dataToSave)
}
catch (error) {
res.status(400).json({ message: error.message })
}
})
```

Model.js

```
const mongoose = require('mongoose');
const dataSchema = new mongoose.Schema({
name: {
required: true,
type: String
},
age: {
required: true,
type: Number
}
})
module.exports = mongoose.model('Data', dataSchema)
```

```
import React,{useState,useEffect} from "react";  
import List from "./List";  
import axios from "axios";  
  
function App() {  
var [card,Setcard]=useState([]);  
const  
token="eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiIsImtpZCI6IjI4YTMxOGY3LTAwMDAtYTFlYiO  
3ZmExLTIjNzQzM2M2Y2NhNSJ9.eyJpc3MiOiJzdXBldcmNlbGwiLCJhdWQiOiJzdXBldcm  
NlbGw6ZWZtZWFwaSIsImp0aSI6IjZmMDhhM2Y4LTg1NDgtNGM4Ny04MjhmdlTl5YWZkMzcwNGViMSIsImhhdCI6MT  
Y5NjY2NzIyNCwic3ViIjoizGV2ZWxvcGVyLzlzZWl3ZWdMLTk3NjYtOTMzNy1lINDRhLTVMODcwZDk2ODY5MA  
ODY5MyIsInR5cyBleCY6WyJyb3lhbmGUieXSwibGltaXRzIjpbeysJ0aWVyIjoizGV2ZWxvcGVyL3NpbH  
ZlcilnR5cGUioiJ0aHJvdHRsaW5nIn0seyJjaWRycyI6WyI0OS4yMDQuMTE1LjlzNCJdlCJ0eXB1I  
loiY2xpZW50In1dfQ._lyLdjqezx2tGH9j5tvH0RWx5KUTGL7DEM8KWMSvx0f4YboME XxBuGbIRFK1qy_jNELhSPLAWTN710DgNoPtw";  
var cors=require('cors');  
const [currenturl,setcurrenturl]=  
useState("https://proxy.royaleapi.dev/v1/cards?limit=20")  
const [load,setload]=useState(true)  
//var express=require('express');  
//var app=express();  
//App.use(cors());  
const cfg={  
headers:{  
"Access-Control-Allow-Origin": "*",  
"Access-Control-Allow-Methods": "GET,PUT,POST,DELETE,PATCH,OPTIONS",  
"Access-Control-Allow-Headers":"Content-Type,X-AuthToken,Origin,Authorization",  
"Authorization":"Bearer $ {eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiIsImtpZCI6IjI4YTMxOGY3LTAwMDAtYTFlYiO3ZmEx LTIjNzQzM2M2Y2NhNSJ9.eyJpc3MiOiJzdXBldcmNlbGwiLCJhdWQiOiJzdXBldcmNlbGw6 Z2FtZWFwaSIsImp0aSI6IjZmMDhhM2Y4LTg1NDgtNGM4Ny04MjhmdlTl5YWZkMzcwNGViMSIsImhhdCI6MTY5NjY2NzIy NCwic3ViIjoizGV2ZWxvcGVyLzlzZWl3ZWdMLTk3NjYtOTMzNy1lINDRhLTVM ODcwZDk2ODY5MA sInR5cGUioiJ0aHJvdHRsaW5nIn0seyJjaWRycyI6WyI0NS43OS4yMTguNzkiXSwidHlw ZiSi6ImNsa
```

WVudCJ9XX0.2EXhoFupm88uj_CITD3hzNnDYWhGTMog9_0y9Q5pnlHd4O2qkyLyNaU
F2QvbBZdHP5i
J0PNFZvBip8mYFItoVg"

```
}  
}  
useEffect(()=>{//Fetch once //[] everytime cururl change fetch  
setload(true)  
let c1  
axios.get(currenturl,cfg,  
{  
cancelToken: new axios.CancelToken(c =>c1=c)  
}  
).then(  

```

```
response =>{  
setload(false)  
Setcard(response.data.items.map(p=>p.iconUrls.medium))  
console.log(card)  
}  
).catch(err=>console.log(err))  
return ()=> c1()  
},[currenturl])  
if(load) return "Loading..."  
return (  
<List card={card}></List>  
);  
}
```

export default App;

index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

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

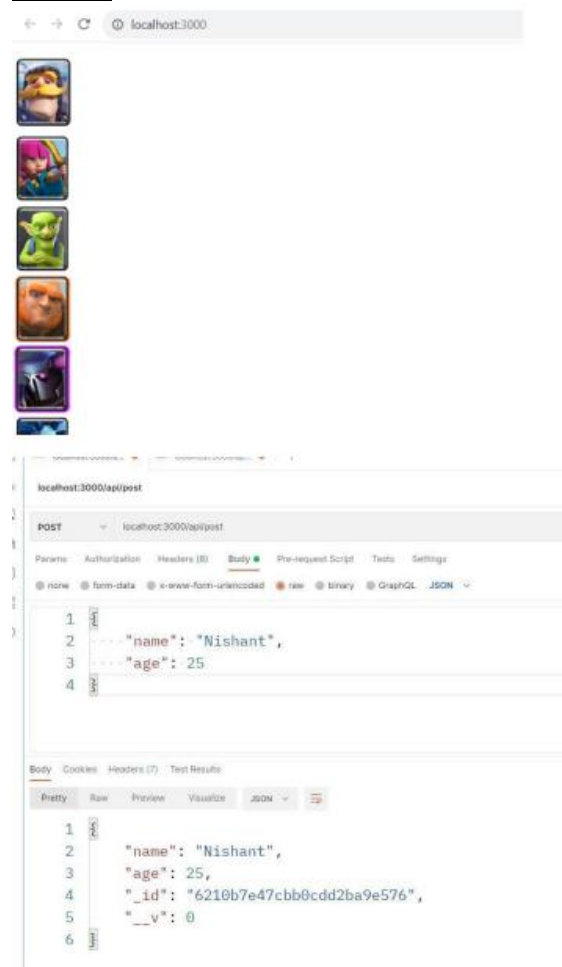
// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: <https://bit.ly/CRA-vitals>

List.js

```
import React from 'react'
export default function List({card}) {
  return (
    <div>
      {card.map(p=>(
        <div key={p}>
          <img src={p}></img>
        </div>
      ))}
    </div>)}
  )
}
```

Output:



Github Link: <https://github.com/AjayBadrinath/CS3809-WebTechLab/tree/main/Lab8-REST%20API%20with%20Express%2CNode%2CMongo>

Result:

Thus REST API Implemented using EJS,Mongo.

