

## Table Of Contents

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

<b>Ex. No: 1</b>	<b>PersonalCV</b>
<b>13.07.2023</b>	

**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:**

```

head>
<title>CV</title>
</head>
<style>
table, th, td {
border:0.5px solid black;
}
</style>
<body style="background-color:rgb(171, 175, 117);">
<table style="width:100%">
<tr>
<td><h1>Harshath Udayakumar</h1></td>
<td></td>
<td><h3>contact : 7358629439 </h3></td>
</tr>
<tr>
<td><h5>Quick learner,problem solver</h5></td>
<td><a href="linkedin.com/in/harshath-udayakumar-91a11a25a"></a></td>
<td><h5><a href="mailto:harshathudayakumar1803@gmail.com"></a></h5></td>
</tr>
</table>
<hr>
<h2>EDUCATION</h2>
<table style="width:100%">
<tr>
<td><h3>DEGREE/GRADE</h3></td>
<td><h3>INSTITUTION</h3></td>
<td><h3>SCORE</h3></td>
</tr>
<tr>
<td><h5>BTech CSE(IoT)</h5></td>
<td><h5>SNU</h5></td>
<td><h5>8.6 cgpa</h5></td>

```

```

</tr>
<tr>
<td><h5>12th</h5></td>
<td><h5>SBOA JC</h5></td>
<td><h5>95%</h5></td>
</tr>
<tr>
<td><h5>10th</h5></td>
<td><h5>SBOA JC</h5></td>
<td><h5>95%</h5></td>
</tr>
</table>
<hr>
<h2>EXPERIENCE</h2>
<h3>MINI PROJECTS</h3>
<table style="width:100%">
<tr>
<td><h3>LINE FOLLOWING ROBOT</h3></td>
<td><h3>FOOD ORDERING WEBSITE</h3></td>
<td><h3>GPS TRACKING SYSTEM</h3></td>
</tr>
<tr>
<td><h5>Developed a basic prototype of a robot which follows a given boundary using an arduino board and sensors.</h5></td>
<td><h5>A simple website which displays various canteens and food items available within our college campus and allows the user to order and pay online. It also has the extension of business handles to verify the received orders.</h5></td>
<td><h5>A basic model which uses google API and maps for tracking the bus routes of the college bus. It is scalable and can be used for developing applications which enables the students to login and track the location of college busses</h5></td>
</tr>
</table>
<hr>
<h2>SKILLS</h2>
<table style="width:100%">
<tr>
<td><h3>PROGRAMMING</h3></td>
<td></td>
<td><h3>WEB DEVELOPMENT</h3></td>
<td></td>
<td><h3>OTHERS</h3></td>
<td></td>
</tr>
<tr>
<td><h5>C</h5></td>
<td><h5> ★ ★ ★ ★ </h5></td>
<td><h5>HTML</h5></td>
<td><h5> ★ ★ ★ ★ ★ </h5></td>
<td><h5>DATA STRUCTURES</h5></td>
<td><h5> ★ ★ ★ </h5></td>
</tr>

```

```

<tr>
<td><h5>PYTHON</h5></td>
<td><h5>★ ★ ★ ★ ★</h5></td>
<td><h5>CSS</h5></td>
<td><h5>★ ★ ★ ★ </h5></td>
<td><h5>SQL</h5></td>
<td><h5>★ ★ ★ ★ ★</h5></td>
</tr>
<tr>
<td><h5>JAVA</h5></td>
<td><h5>★ ★ ★ </h5></td>
<td><h5>JAVASCRIPT</h5></td>
<td><h5>★ ★ ★ </h5></td>
<td><h5>TABLEAU</h5></td>
<td><h5>★ ★ ★ ★ </h5></td>
</tr>
</table>
<hr>
<h2>ADDITIONAL SECTION</h2>
<p>* Hinduja Scholarship for academic excellence </p>
<p>* Secured 86 percentile in the JEE mains 2021 </p>
<p>* Proficiency holder in school </p>
<p>* Won medal in maths olympiad </p>
<p>* Player of the college cricket team </p>
<hr>
<h2>LANGUAGES</h2>
<p>* English</p>
<p>* French</p>
<p>* Tamil</p>
<p>* Telugu</p>
</body>

```

**Output:**

Harshath Udayakumar			contact : 7358629439												
Quick learner problem solver		<a href="https://linkedin.com/in/harshathudayakumar-91a11a25a">https://linkedin.com/in/harshathudayakumar-91a11a25a</a>	<a href="mailto:harshathudeyakumar1993@gmail.com">harshathudeyakumar1993@gmail.com</a>												
<b>EDUCATION</b>															
<table border="1"> <thead> <tr> <th>DEGREE/GRADE</th><th>INSTITUTION</th><th>SCORE</th></tr> </thead> <tbody> <tr> <td>B.Tech CSE(IIT)</td><td>SNU</td><td>8.6 CGPA</td></tr> <tr> <td>12th</td><td>SBOA JC</td><td>95%</td></tr> <tr> <td>10th</td><td>SBOA JC</td><td>95%</td></tr> </tbody> </table>				DEGREE/GRADE	INSTITUTION	SCORE	B.Tech CSE(IIT)	SNU	8.6 CGPA	12th	SBOA JC	95%	10th	SBOA JC	95%
DEGREE/GRADE	INSTITUTION	SCORE													
B.Tech CSE(IIT)	SNU	8.6 CGPA													
12th	SBOA JC	95%													
10th	SBOA JC	95%													
<b>EXPERIENCE</b>															
<b>MINI PROJECTS</b>															
LINE FOLLOWING ROBOT	FOOD ORDERING WEBSITE	GPS TRACKING SYSTEM													
Developed a basic prototype of a robot which follows a given boundary using an arduino board and sensors.	A simple website which displays various canteens and food items available within our college campus and allows the user to order and pay online. It also has the extension of business handles to verify the received orders.	A basic model which uses google API and maps for tracking the bus routes of the college bus. It is scalable and can be used for developing applications which enables the students to login and track the location of college buses													
<b>SKILLS</b>															
PROGRAMMING		WEB DEVELOPMENT	OTHERS												
C	★★★★	HTML	★★★★★												
PYTHON	★★★★★	CSS	★★★★												
JAVA	★★★	JAVASCRIPT	★★★												
<b>ADDITIONAL SECTION</b>															
<ul style="list-style-type: none"> <li>* Hinduja Scholarship for academic excellence</li> <li>* Secured 86 percentile in the JEE mains 2021</li> <li>* Proficiency holder in school</li> <li>* Won medal in maths olympiad</li> <li>* Player of the college cricket team</li> </ul>															
<b>LANGUAGES</b>															
English French Tamil Telugu															

Github link :

[https://github.com/harshath1234/WEB\\_TECH\\_LAB](https://github.com/harshath1234/WEB_TECH_LAB)

### Result:

Thus Personal CV was created using basic HTML

<b>Ex. No: 2</b>	<b>CSS Enabled CV</b>
<b>20.07.2023</b>	

**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 :-

```
<!DOCTYPE html>
<head>
<title>CV</title>
<link rel="stylesheet" href="CVstyles.css">
</head>
<style>
table, th, td {
border:0.5px solid black;
}
/* The container */
.container {
display: block;
position: relative;
padding-left: 35px;
margin-bottom: 12px;
cursor: pointer;
font-size: 22px;
-webkit-user-select: none;
-moz-user-select: none;
-ms-user-select: none;
user-select: none;
}
/* Hide the browser's default checkbox */
.container input {
position: absolute;
opacity: 0;
```

```
cursor: pointer;
height: 0;
width: 0;
}
/* Create a custom checkbox */
.checkmark {
position: absolute;
top: 0;
left: 0;
height: 25px;
width: 25px;
background-color: #eee;
}
/* On mouse-over, add a grey background color */
.container:hover input ~ .checkmark {
background-color: #ccc;
}
/* When the checkbox is checked, add a blue background */
.container input:checked ~ .checkmark {
background-color: #2196F3;
}
/* Create the checkmark/indicator (hidden when not checked) */
.checkmark:after {
content: "";
position: absolute;
display: none;
}
/* Show the checkmark when checked */
.container input:checked ~ .checkmark:after {
display: block;
}
/* Style the checkmark/indicator */
.container .checkmark:after {
left: 9px;
top: 5px;
width: 5px;
height: 10px;
border: solid white;
border-width: 0 3px 3px 0;
-webkit-transform: rotate(45deg);
-ms-transform: rotate(45deg);
transform: rotate(45deg);
}

```

</style>

```
<body style="background-color:rgb(236, 168, 255);">
<table style="width:100%">
<tr class="head1">
<td><h1>Harshath</h1>
<h1>Udayakumar</h1></td>
<td>
```

```
<div class="grow">

</div>
</td>
<td><h3>contact : 7358629439 </h3></td>
</tr>
<tr class="head2">
<td><h3>> Quick learner </h3><h3>> problem solver</h3></td>
<td><a href>linkedin.com/in/harshath-udayakumar-91a11a25a</a></td>
<td><a href>harshathudayakumar1803@gmail.com</a></td>
</tr>
</table>
<hr>
<h2>EDUCATION</h2>
<table style="width:45%">
<tr>
<td><h3>DEGREE/GRADE</h3></td>
<td><h3>INSTITUTION</h3></td>
<td><h3>SCORE</h3></td>
</tr>
<tr>
<td><h5>BTech CSE(IoT)</h5></td>
<td><h5>SNU</h5></td>
<td><h5>8.6 cgpa</h5></td>
</tr>
<tr>
<td><h5>12th</h5></td>
<td><h5>SBOA JC</h5></td>
<td><h5>95%</h5></td>
</tr>
<tr>
<td><h5>10th</h5></td>
<td><h5>SBOA JC</h5></td>
<td><h5>95%</h5></td>
</tr>
</table>
<hr>
<table style="width:100%">
<h2>EXPERIENCE</h2>
<h4>MINI PROJECTS</h4>
<tr>
<td><h3>LINE FOLLOWING ROBOT</h3></td>
<td><h3>FOOD ORDERING WEBSITE</h3></td>
<td><h3>GPS TRACKING SYSTEM</h3></td>
</tr>
<tr>
<td>
<h5>Developed a basic prototype of a robot which follows a
given boundary using an arduino board and sensors.</h5>
```

```

<h5>
<div class="grow">

</div></h5>
<h5><a href="https://circuitdigest.com/arduino-uno-line-follower-robot"></a></h5>
</td>
<td>
<h5>A simple website which displays various canteens and food items available within our college campus and allows the user to order and pay online. It also has the extension of business handles to verify the received orders.</h5>
<h5>
<div class="grow">

</div>
</h5>
<h5><a href="https://www.googleadservices.com"></a></h5>
</td>
<td><h5>A basic model which uses google API and maps for tracking the bus routes of the college bus. It is scalable and can be used for developing applications which enables the students to login and track the location of college busses</h5>
<h5>
<div class="grow">

</div>
</h5>
<h5><a href="https://www.googleadservices.com"></a></h5>
</td>
</tr>
</table>
<hr>
<h2>SKILLS</h2>
<table style="width:80%">
<tr>
<td><h3>PROGRAMMING</h3></td>
<td></td>
<td><h3>WEB DEVELOPMENT</h3></td>
<td></td>
<td><h3>OTHERS</h3></td>
<td></td>
</tr>
<tr>
<td><h5>C</h5></td>
<td><h5 class="star">★ ★ ★ ★ </h5></td>
<td><h5>HTML</h5></td>

```

<td><h5 class="star">★ ★ ★ ★ ★</h5></td>
<td><h5>DATA STRUCTURES</h5></td>
<td><h5 class="star">★ ★ ★</h5></td>
</tr>
<tr>
<td><h5>PYTHON</h5></td>
<td><h5 class="star">★ ★ ★ ★ ★</h5></td>
<td><h5>CSS</h5></td>
<td><h5 class="star">★ ★ ★ ★</h5></td>
<td><h5>SQL</h5></td>
<td><h5 class="star">★ ★ ★ ★ ★</h5></td>
</tr>
<tr>
<td><h5>JAVA</h5></td>
<td><h5 class="star">★ ★ ★</h5></td>
<td><h5>JAVASCRIPT</h5></td>
<td><h5 class="star">★ ★ ★</h5></td>
<td><h5>TABLEAU</h5></td>
<td><h5 class="star">★ ★ ★ ★</h5></td>
</tr>
</table>
<hr>
<h2>ADDITIONAL SECTION</h2>
<p>* Hinduja Scholarship for academic excellence </p>
<p>* Secured 86 percentile in the JEE mains 2021 </p>
<p>* Proficiency holder in school </p>
<p>* Won medal in maths olympiad </p>
<p>* Player of the college cricket team </p>
<hr>
<h2>LANGUAGES</h2>
<table>
<tr>
<td>
<p>* English </p>
</td>
<td>
<label class="container"> spoken
<input type="checkbox" checked="checked">
<span class="checkmark"></span>
</label>
</td>
<td>
<label class="container">written
<input type="checkbox" checked="checked">
<span class="checkmark"></span>
</label>
</td>
</tr>
<tr>

```
<td>
<p>* French</p>
</td>
<td>
<label class="container">spoken
<input type="checkbox" checked="checked">
<span class="checkmark"></span>
</label>
</td>
<td>
<label class="container">written
<input type="checkbox" checked="checked">
<span class="checkmark"></span>
</label>
</td>
</tr>
<tr>
<td>
<p>* Tamil</p>
</td>
<td>
<label class="container">spoken
<input type="checkbox" checked="checked">
<span class="checkmark"></span>
</label>
</td>
<td>
<label class="container">written
<input type="checkbox" checked="checked">
<span class="checkmark"></span>
</label>
</td>
</tr>
<tr>
<td>
<p>* Telugu</p>
</td>
<td>
<label class="container">spoken
<input type="checkbox" checked="checked">
<span class="checkmark"></span>
</label>
</td>
<td>
<label class="container">written
<input type="checkbox" checked="checked">
<span class="checkmark"></span>
</label>
</td>
</tr>
```

```
</table>
</body>
```

CSS:

```
h1{
    font-family: 'Times New Roman', Times, serif;
    font-size: 50px;
    text-align: center;
}
.head1{
    text-align: center;
}
.head2{
    text-align: center;
}
h2{
    font-family: Verdana, Geneva, Tahoma, sans-serif;
}
h3{
    text-align: center;
}
h5{
    text-align: center ;
}
a:hover{
    color:ghostwhite;
}
.star:hover{
    color:darkgoldenrod;
}
hr{
    color: black;
}
.grow img{
    transition: 1s ease;
    border-radius: 50%;
}
.grow img:hover{
    -webkit-transform: scale(1.2);
    -ms-transform: scale(1.2);
    transform: scale(1.2);
    transition: 1s ease;
}
```

**Output:**

<b>Harshath</b> <b>Udayakumar</b>		contact : 7358629439
> Quick learner > problem solver	<a href="https://linkedin.com/in/harshath-udayakumar-91a11a239">linkedin.com/in/harshath-udayakumar-91a11a239</a>	<a href="mailto:harshathudayakumar1803@gmail.com">harshathudayakumar1803@gmail.com</a>

## EDUCATION

DEGREE/GRADE	INSTITUTION	SCORE
BTech CSE(IoT)	SNU	8.6 CGPA
12th	SBOA JC	95%
10th	SBOA JC	95%

## EXPERIENCE

### MINI PROJECTS

LINE FOLLOWING ROBOT	FOOD ORDERING WEBSITE	GPS TRACKING SYSTEM
Developed a basic prototype of a robot which follows a given boundary using an arduino board and sensors.  	A simple website which displays various canteens and food items available within our college campus and allows the user to order and pay online. It also has the extension of business handles to verify the received orders.  	A basic model which uses google API and maps for tracking the bus routes of the college bus. It is scalable and can be used for developing applications which enables the students to login and track the location of college buses.  

<https://circuitdigest.com/arduino/arduino-line-follower-robot>

<http://www.googleadservices.com>

<https://www.googleadservices.com>

## SKILLS

PROGRAMMING		WEB DEVELOPMENT		OTHERS	
C	*****	HTML	*****	DATA STRUCTURES	***
PYTHON	*****	CSS	****	SQL	*****
JAVA	***	JAVASCRIPT	***	TABLEAU	*****

## ADDITIONAL SECTION

- \* Hindujia Scholarship for academic excellence
- \* Secured 86 percentile in the JEE mains 2021
- \* Proficiency holder in school
- \* Won medal in maths olympiad
- \* Player of the college cricket team

## LANGUAGES

* English	<input checked="" type="checkbox"/> spoken	<input checked="" type="checkbox"/> written
* French	<input checked="" type="checkbox"/> spoken	<input checked="" type="checkbox"/> written

Github Link:

[https://github.com/harshath1234/WEB\\_TECH\\_LAB](https://github.com/harshath1234/WEB_TECH_LAB)

**Result:**

Thus Personal CV was created using HTML and CSS

<b>Ex. No: 3</b>	<b>Form Making and Validation using JavaScript</b>
<b>27.07.2023</b>	

**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:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<link rel="stylesheet" href="formCSS.css">
</head>
<body>
<h1 class="heading"> PLAYER REGISTRATION FORM </h1>
<script >
function myFunc()
{
confirm("Make sure you enter the right login credentials");
}
function validateform()
{
var name=document.myform.name.value;
var password=document.myform.password.value;
if (name==null || name=="")
{
alert("Name can't be blank");
return false;
```

```

}
else if(password.length<6)
{
alert("Password must be at least 6 characters long.");
return false;
}
function containsNumber(str)
{
a=/\d/.test(str);
}
if(a==false)
{
alert("password must atleast contain 1 digit")
}
}
</script>
<form style="color: #000000; text-align: center;" name="myform"
method="post" onsubmit="return validateform()" >
<br>
PLAYER NAME: <input type="name" name="name">
<br/>
<br>
<label for="team-names">TEAM NAME : </label>
<select name="team-names" id="team-names">
<option value=""></option>
<option value="CSK">CSK</option>
<option value="MI">MI</option>
<option value="RCB">RCB</option>
<option value="KKR">KKR</option>
<option value="GT">GT</option>
<option value="LSG">LSG</option>
<option value="PBKS">PBKS</option>
<option value="SRH">SRH</option>
<option value="DC">DC</option>
<option value="RR">RR</option>
</select>
<br/>
<br>
<label for="player-skillset">PLAYER SKILLSET : </label>
<select name="player-skillset" id="player-skillset">
<option value=""></option>
<option value="BATSMAN">BATSMAN</option>
<option value="BOWLER">BOWLER</option>
<option value="ALL ROUNDER">ALL ROUNDER</option>
<option value="WICKET KEEPER BATSMAN">WICKET KEEPER BATSMAN</option>
</select>
<br/>
<br>
<label for="batting-stats">BATTING STATS : </label>
<select name="batting_stats" id="batting_stats">

```

```

<option value=""></option>
<option value="RIGHT HAND BATSMAN">RIGHT HAND BATSMAN</option>
<option value="LEFT HAND BATSMAN">LEFT ARM BATSMAN</option>
</select>
<br/>
<br>
<label for="bowling-stats">BOWLING SATS : </label>
<select name="bowling-stats" id="bowling-stats">
<option value=""></option>
<option value="RAMP">RIGHT ARM MEDIUM PACER</option>
<option value="LAMP">LEFT ARM MEDIUM PACER</option>
<option value="RAOS">RIGHT ARM OFF SPIN</option>
<option value="LAOS">LEFT ARM OFF SPIN</option>
<option value="RALS">RIGHT ARM LEG SPIN</option>
<option value="LAC">LEFT ARM CHINAMAN</option>
</select>
<br/>
<br>
LOGIN ID : <input type="id" name="id">
<button onclick="myFunc()">  </button>
<br/>
<br>
PASSWORD : <input type="password" name="password">
<button>register</button>
<div class="pwd">
<h3> Crtriteria for password creation</h3>
<input type="checkbox" id="rule1" name="rule1" value="length">
<label for="vehicle1"> the password must be atleast 6 characters long</label><br><br>
<input type="checkbox" id="rule2" name="rule2" value="digit">
<label for="vehicle2">the password must contain atleast one digit</label><br><br>
<input type="checkbox" id="rule3" name="rule3" value="uppercase">
<label for="vehicle3">the password must contain atleast one uppercase letter</label>
</div>
</form>
</body>
</html>

```

CSS:

```

body{
    text-align: center;
}
.pwd{
    text-align: center;
}

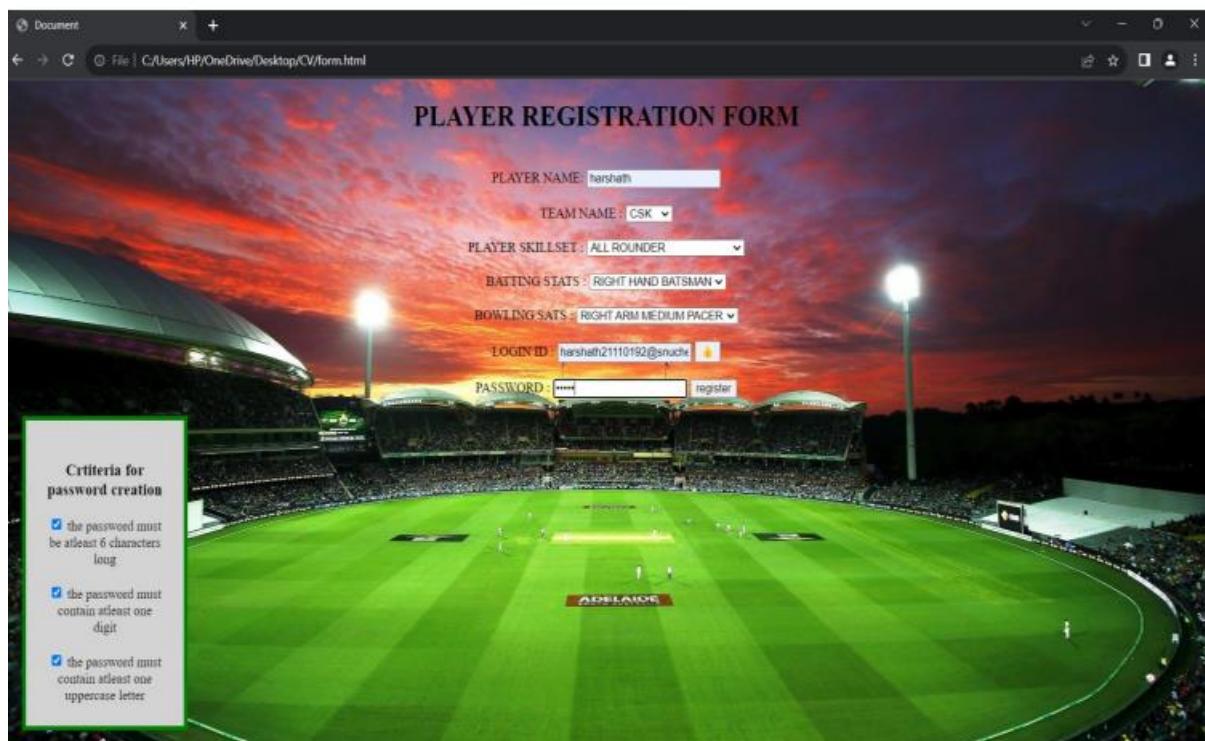
```

```

body {
background-image: url("cricbg3.jpg");
background-size:100%;
}
div {
margin-left: 300px;
text-align: center;
background-color: lightgrey;
width: 150px;
border: 5px solid green;
padding: 25px;
margin:20px
}

```

### **Output:**



The screenshot shows a web browser window titled "Document" with the URL "File | C:/Users/HP/OneDrive/Desktop/CV/form.html". The page has a header "PLAYER REGISTRATION FORM" and a background image of a cricket stadium at sunset. The form fields include:

- PLAYER NAME:
- TEAM NAME:
- PLAYER SKILLSET:
- BATTING STATS:
- BOWLING STATS:
- LOGIN ID:
- PASSWORD:
- register button

A sidebar on the left contains a box titled "Criteria for password creation" with three checked checkboxes:

- the password must be atleast 6 characters long
- the password must contain atleast one digit
- the password must contain atleast one uppercase letter

Github Link:

[https://github.com/harshath1234/WEB\\_TECH\\_LAB](https://github.com/harshath1234/WEB_TECH_LAB)

### **Result:**

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

<b>Ex. No: 4</b>	<b>Angular based App creation</b>
<b>09.08.2023</b>	

**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-W8z6aiwifi1E-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: `.${this.baseUrl}/phil-hearing-IYfp2Ix9nM-unsplash.jpg`,
        availableUnits: 5,
        wifi: true,
        laundry: true
    },
    {
        id: 7,
        name: 'Hopeful Housing Solutions',
        city: 'Oakland',
        state: 'CA',
        photo: `.${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-9rP3mx8qWI-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

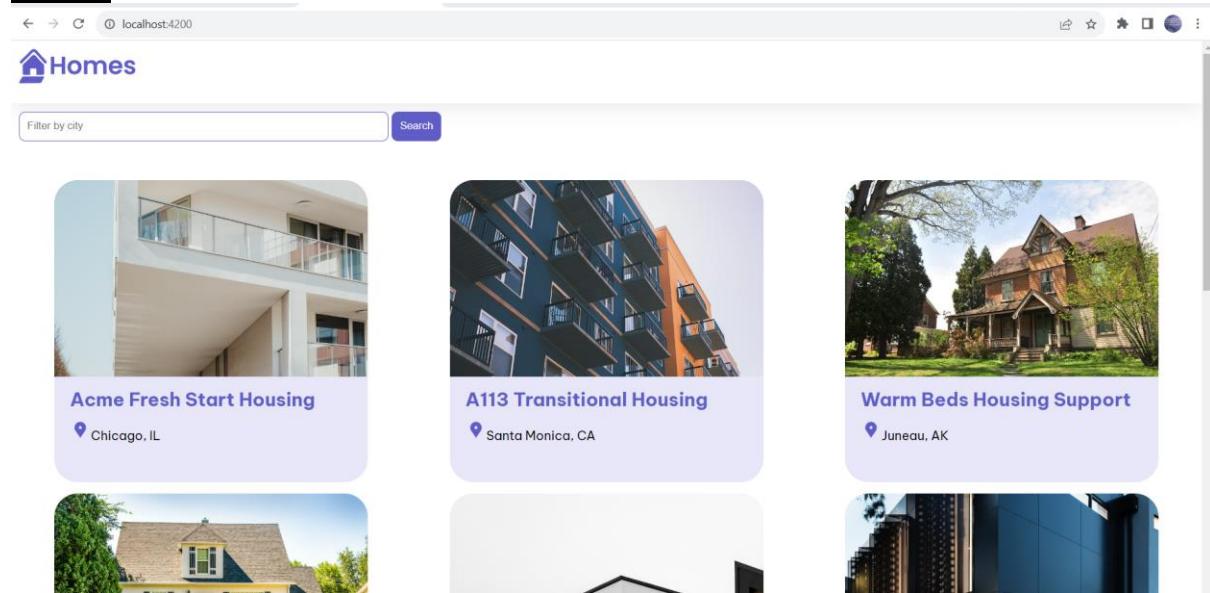
```

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/harshath1234/WEB\\_TECH\\_LAB](https://github.com/harshath1234/WEB_TECH_LAB)

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

<b>Ex. No: 5</b>	<b>React based App Development</b>
<b>15.08.2023</b>	

**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">

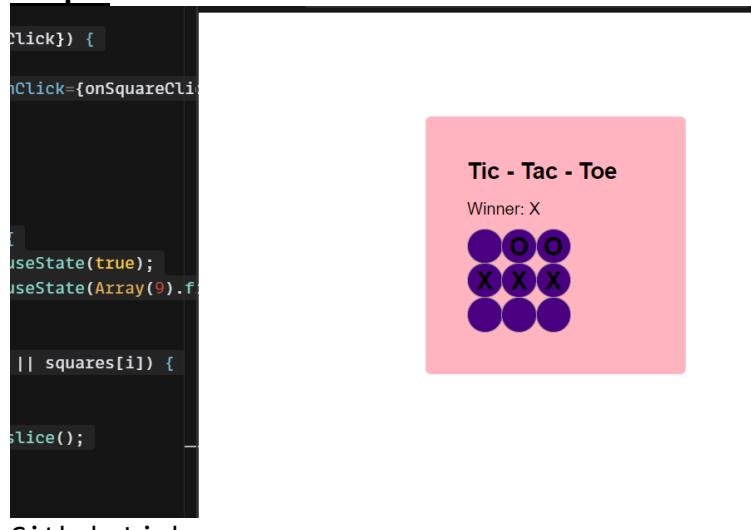
```

```

        <Square value={squares[3]} onSquareClick={() => handleClick(3)} />
        <Square value={squares[4]} onSquareClick={() => handleClick(4)} />
        <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/harshath1234/WEB\\_TECH\\_LAB](https://github.com/harshath1234/WEB_TECH_LAB)

### Result:

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

<b>Ex. No: 6</b>	
<b>21.09.2023</b>	<b>Web Server Creation using NodeJS</b>

**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/harshath1234/WEB\\_TECH\\_LAB](https://github.com/harshath1234/WEB_TECH_LAB)

### **Result:**

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

<b>Ex. No: 7</b>	<b>Routing Implementation using ExpressJS</b>
<b>28.09.2023</b>	

**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

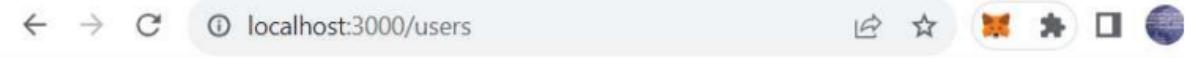
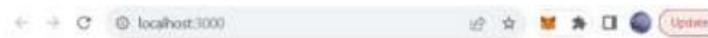
```

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/harshath1234/WEB\\_TECH\\_LAB](https://github.com/harshath1234/WEB_TECH_LAB)

**Result:**

Thus routing features were implemented using Express JS.

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

**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)

```

app.js (API fetching for demo purpose)

```
import React,{useState,useEffect} from "react";
import List from "./List";
import axios from "axios";

function App() {
var [card,Setcard]=useState([]);
const
token="eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiIsImtpZCI6IjI4YTMxOGY3LT AwMDAtYTFIYi0
3ZmExLTJjNzQzM2M2Y2NhNSJ9eyJpc3MiOiJzdXBlc NlbGwiLCJhdWQiOjzdXBlc NlbGw6Z2Ft
ZWFWaSIsImp0aSI6IjZmMDhhM2Y4LTg1NDgtNGM4Ny04Mjh mLT I5YWZkMzcwNGVi
MSIsIm lhdCI6MT
Y5NjY2NzIyNCwic3ViIjoiZGV2ZWxvcGVyLzIzZWI3ZWM3LTk3NjYtOTMzNy1lINDRh
LTVIODcwZDk2
ODY5MyIsInNjb3BlcyI6WyJyb3lh bGU iXS wibGltaXRzIjpbe yJ0aWVyIjoiZGV2ZWxvcGVyL3NpbH
ZlcIIsInR5cGUiOj0aHJvdHRsaW5nIn0seyJjaWRycyI6WyI0OS4yMDQuMTE1LjIzNCJdLCJ0eXB1
IjoiY2xpZW50In1dfQ._lyLdjqe zx2tGH9j5tvH0RWx5KUTGL7DEM8KWMSvx0f4YboME
XxBuGbIRFK1qy_jNELhSPLAWTN710DgNoPt w";
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
${eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiIsImtpZCI6IjI4YTMxOGY3LT AwMDAtYTFIYi03ZmEx
LTJjNzQzM2M2Y2NhNSJ9eyJpc3MiOiJzdXBlc NlbGwiLCJhdWQiOjzdXBlc NlbGw6Z2FtZWFWa
SIsImp0aSI6IjhiYTE3NGI4LTZhOWQtNDViMi1iZjkyLTVlMW NiNDM4NWM2ZiIsIm lhdCI6MTY5Njc
wOTA4OSwic3ViIjoiZGV2ZWxvcGVyLzIzZWI3ZWM3LTk3NjYtOTMzNy1lINDRhLTVI
ODcwZDk2ODY5M
yIsInNjb3BlcyI6WyJyb3lh bGU iXS wibGltaXRzIjpbe yJ0aWVyIjoiZGV2ZWxvcGVyL3NpbHZlciI
sInR5cGUiOj0aHJvdHRsaW5nIn0seyJjaWRycyI6WyI0NS43OS4yMTguNzkiXSwidHlwZ
SI6ImNs a
WVudCJ9XX0.2EXhoFupm88uj_CITD3hzNnDYWhGTMog9_0y9Q5pn lHd4O2qkyLyNaUF2QvbBZdHP5i
```

```

JOPNFZvBip8mYFItOvg"
}
}
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

```

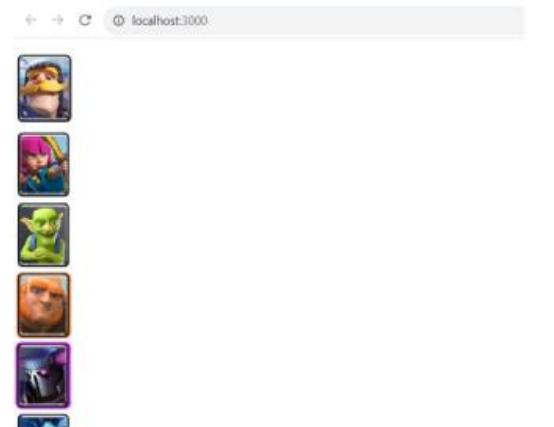
List.js

```

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

```

**Output:**



A screenshot of a web browser window showing a list of five character icons from a game. The icons are arranged vertically in a column. From top to bottom, they are: a yellow bird-like creature, a pink and purple character with wings, a green goblin-like character, an orange bear-like character, and a dark purple character with a hood.


A screenshot of the Postman application interface. The URL is set to `localhost:3000/api/post`. The method is selected as `POST`. The `Body` tab is active, showing the following JSON payload:

```

1 {
2   "name": "Nishant",
3   "age": 25
4 }

```


The response body shows the JSON object returned by the server, including a generated MongoDB document ID:

```

1 {
2   "name": "Nishant",
3   "age": 25,
4   "_id": "6210b7e47cbb0cdd2ba9e576",
5   "__v": 0
6 }

```

Github Link :

[https://github.com/harshath1234/WEB\\_TECH\\_LAB](https://github.com/harshath1234/WEB_TECH_LAB)

**Result:**

Thus REST API Implemented using EJS,Mongo.