SHIV NADAR

UNIVERSITY

CHENNAI

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF ENGINEERING

LABORATORY RECORD

B.TECH

(YEAR : 2023-2024)

NAME	: AMOGHAVARSH . S.H
REG. NO.	: 210111O2014
DEPT.	: BTech CSE SEM. : V CLASS & SEC : IoT.A

SHIV NADAR
UNIVERSITY
CHENNAI

BONAFIDE CERTIFICATE

Certified that this is the bonafide record of the practical work done in the

CS3809 Web Technologies Laboratory by

Name Amoghavarsh. S.H

Register Number 21011102014

Semester V Class & Sec. III - 'A'

Branch BTech CSE

SHIV NADAR UNIVERSITY Chennai

During the Academic year 2023 - 2024

[illegible]
Faculty

[illegible]
Head of the Department

Submitted for the Web Technologies Practical Examination held at SNU CHENNAI on 06|11|2023

[illegible]
Internal Examiner

External Examiner

INDEX

Name : Amoghavarsh. H Reg. No. 21011102014

Sem : V Class & Sec : BTech CSE (IoT) A

Ex. No.	Date of Expt.	Title of the Experiment	Page No.	Signature of the Faculty	Remarks
1	13/7/23	Personal CV using HTML	1		
2	20/7/23	CSS enabled CV	4	[illegible]	
3.	27/7/23	Form making and validation using JS	11		
4.	9/8/23	Angular based app creation	17		
5.	15/8/23	React based app creation	20	[illegible]	
6	21/9/23	Web server creation using Node JS	24		
7.	28/9/23	Routing Implementation using Express JS	26		
8	5/10/23	Building a REST API with Express, Node & Mongo DB	28	[illegible]	

Ex. No: 1	**Personal CV Using HTML**
13.07.2023	

Aim:

To create a CV using only HTML.

Algorithm:

1. Plan your CV.
2. Create a HTML document.
3. Add the required content.
4. Preview and Save

Program:

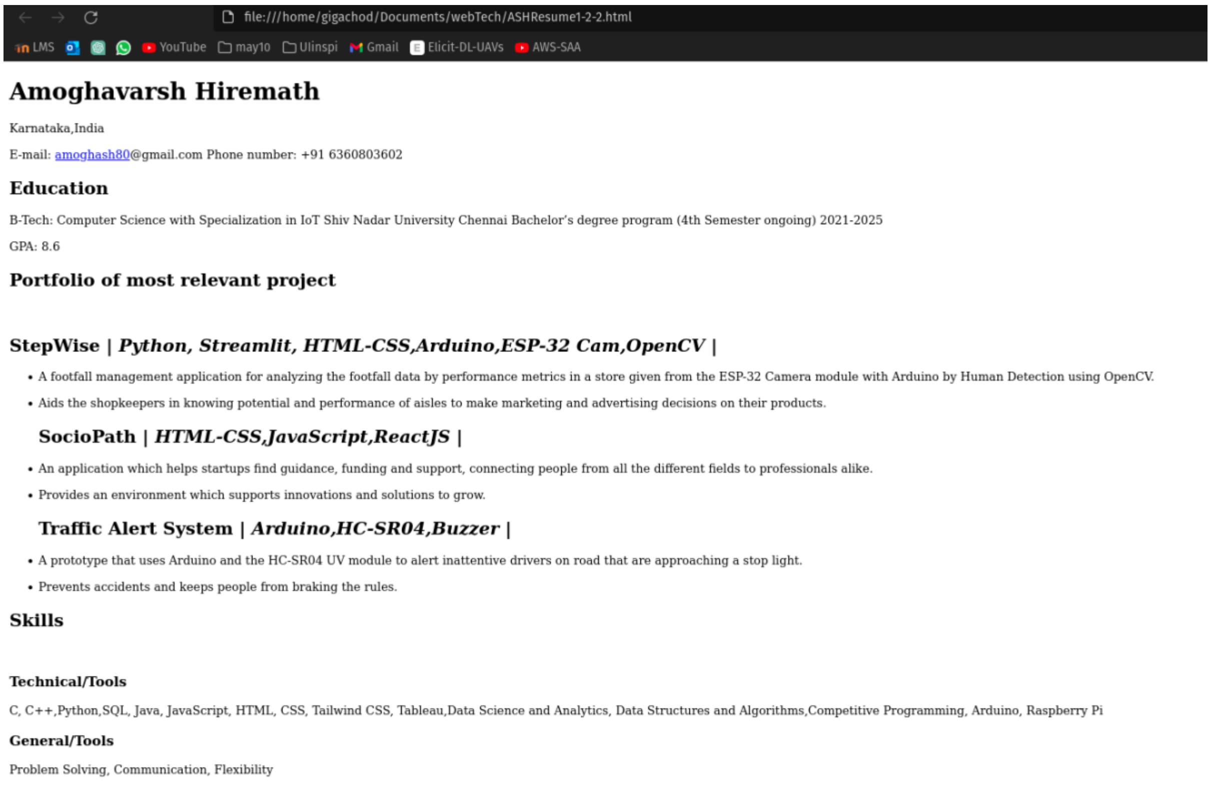
```
<!DOCTYPE  html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-
8"/>
        <title>file_1689783909040</title>
        <style type="text/css"> * {margin:0; padding:0; text-indent:0; }
        h1 { color: #003579; font-family:Calibri, sans-serif; font-style:
italic; font-weight: bold; text-decoration: none; font-size: 20.5pt; }
        .p, p { color: black; font-family:Calibri, sans-serif; font-style:
normal; font-weight: normal; text-decoration: none; font-size: 11pt;
margin:0pt; }
        .s1 { color: #003579; font-family:Calibri, sans-serif; font-style:
italic; font-weight: normal; text-decoration: none; font-size: 11pt; }
        .s2 { color: black; font-family:Calibri, sans-serif; font-style:
italic; font-weight: normal; text-decoration: none; font-size: 11pt; }
        .s3 { color: #003579; font-family:Calibri, sans-serif; font-style:
italic; font-weight: bold; text-decoration: none; font-size: 11pt; }
        .s4 { color: black; font-family:Calibri, sans-serif; font-style:
italic; font-weight: bold; text-decoration: none; font-size: 11pt; }
        h2 { color: black; font-family:Calibri, sans-serif; font-style:
normal; font-weight: bold; text-decoration: none; font-size: 11pt; }
        .s5 { color: black; font-family:Calibri, sans-serif; font-style:
normal; font-weight: bold; text-decoration: none; font-size: 11pt; }
        .s6 { color: black; font-family:Calibri, sans-serif; font-style:
italic; font-weight: normal; text-decoration: none; font-size: 11pt; }
        .s7 { color: black; font-family:Calibri, sans-serif; font-style:
normal; font-weight: normal; text-decoration: none; font-size: 11pt; }
        li {display: block; }
        #l1 {padding-left: 0pt; }
        #l1> li>*:first-child:before {content: " "; color: black; font-
family:Symbol, serif; font-style: normal; font-weight: normal; text-
decoration: none; font-size: 11pt; }
        table, tbody {vertical-align: top; overflow: visible; }
        </style>
    </head>
    <body>
        <h1 style="padding-left: 79pt;text-indent: 0pt;line-height: 
25pt;text-align: center;">Amoghavarsh Hiremath</h1>
        <p style="padding-top: 2pt;padding-left: 79pt;text-indent: 
0pt;text-align: center;">Karnataka,India</p>
        <p class="s1" style="padding-top: 2pt;padding-left: 79pt;text-
indent: 0pt;text-align: center;">E-mail: <a 
href="mailto:austinjaisonj@gmail.com" style=" color: black; font-
family:Calibri, sans-serif; font-style: italic; font-weight: normal; text-
decoration: none; font-size: 11pt;" target="_blank">amoghash80</a><span 
class="p">@gmail.com   </span>Phone number: <span class="p">+91 
6360803602</span></p>
        <p class="s3" style="padding-top: 7pt;padding-left: 11pt;text-
indent: 0pt;text-align: left;"><a name="bookmark0">Education</a></p>
        <p class="s4" style="padding-left: 5pt;text-indent: 23pt;line-
height: 20pt;text-align: left;"><a name="bookmark1">B-Tech: Computer 
Science with Specialization in IoT           </a><span class="s2">Shiv 
Nadar University Chennai Bachelor’s degree program (4th Semester ongoing) 
2021-2025</span></p><p class="s2" style="padding-left: 27pt;text-indent: 
0pt;line-height: 13pt;text-align: left;">GPA: 8.6</p><h2 style="padding-
top: 6pt;padding-left: 27pt;text-indent: 0pt;text-align: left;">Class XII 
<i>Jambagi PU College</i></h2><p class="s2" style="padding-left: 
27pt;text-indent: 0pt;line-height: 13pt;text-align: left;">Department of 
Pre-University Education(Karnataka)                         2020-
2021</p><p class="s2" style="padding-left: 27pt;text-indent: 0pt;line-
height: 13pt;text-align: left;">Final Percentage: 97%</p><h2 
style="padding-top: 6pt;padding-left: 27pt;text-indent: 0pt;text-align: 
left;">Class X                                          <i>Phoenix Public 
School</i></h2>
        <p class="s2" style="padding-left: 27pt;text-indent: 0pt;line-
height: 13pt;text-align: left;">Indian Certificate of Secondary Education 
Examination (ICSE)                    2018-2019</p><p class="s2" 
style="padding-left: 27pt;text-indent: 0pt;line-height: 13pt;text-align: 
left;">Final Percentage: 90.2%</p><p class="s3" style="padding-top: 
6pt;padding-left: 11pt;text-indent: 0pt;text-align: left;"><a 
name="bookmark2">Portfolio of most relevant project</a></p><p style="text-
indent: 0pt;text-align: left;"><br/></p>
        <h2 style="padding-left: 24pt;text-indent: 0pt;line-height: 
13pt;text-align: left;">StepWise                   <i>| Python, 
Streamlit, HTML-CSS,Arduino,ESP-32 Cam,OpenCV </i>|</h2>
        <ul id="l1">
            <li data-list-text="">
                <p style="padding-left: 62pt;text-indent: -18pt;text-
align: left;">A footfall management application for analyzing the footfall 
data by performance metrics in a store given from the ESP-32 Camera module 
with Arduino by Human Detection using OpenCV.</p>
            </li>
            <li data-list-text="">
                <p style="padding-left: 62pt;text-indent: -18pt;text-
align: left;">Aids the shopkeepers in knowing potential and performance of 
aisles to make marketing and advertising decisions on their products.</p>
```

```
            <h2 style="padding-left: 27pt;text-indent: 0pt;line-height: 
13pt;text-align: left;">SocioPath                                     |
<i>HTML-CSS,JavaScript,ReactJS |</i></h2>
            </li>
            <li data-list-text="">
                <p style="padding-left: 62pt;text-indent: -18pt;text-
align: left;">An application which helps startups find guidance, funding
and support, connecting people from all the different fields to
professionals alike.</p>
            </li>
            <li data-list-text="">
                <p style="padding-left: 62pt;text-indent: -18pt;line-
height: 13pt;text-align: left;">Provides an environment which supports
innovations and solutions to grow.</p><h2 style="padding-left: 27pt;text-
indent: 0pt;line-height: 13pt;text-align: left;">Traffic Alert System
| <i>Arduino,HC-SR04,Buzzer |</i></h2></li><li data-list-text=""><p
style="padding-left: 62pt;text-indent: -18pt;text-align: left;">A
prototype that uses Arduino and the HC-SR04 UV module to alert inattentive
drivers on road that are approaching a stop light.</p></li><li data-list-
text=""><p style="padding-left: 62pt;text-indent: -18pt;line-height:
14pt;text-align: left;">Prevents accidents and keeps people from braking
the rules.</p></li></ul><p class="s3" style="padding-left: 11pt;text-
indent: 0pt;line-height: 13pt;text-align: left;"><a
name="bookmark3">Skills</a></p><p style="text-indent: 0pt;text-align:
left;"><br/></p><h2 style="padding-left: 27pt;text-indent: 0pt;text-align:
left;">Technical/Tools</h2><p style="padding-left: 27pt;text-indent:
0pt;text-align: left;">C, C++,Python,SQL, Java, JavaScript, HTML, CSS,
Tailwind CSS, Tableau,Data Science and Analytics, Data Structures and
Algorithms,Competitive Programming, Arduino, Raspberry Pi</p><h2
style="padding-left: 27pt;text-indent: 0pt;line-height: 13pt;text-align:
left;">General/Tools</h2><p style="padding-left: 27pt;text-indent:
0pt;line-height: 13pt;text-align: left;">Problem Solving, Communication,
Flexibility</p><p style="text-indent: 0pt;text-align: left;"><br/></p>
                <p class="s3" style="padding-left: 11pt;text-indent:
0pt;text-align: left;"><a name="bookmark4">Achievements and
Certifications</a></p><p style="text-indent: 0pt;text-align:
left;"><br/></p>
                <p style="text-indent: 0pt;text-align: left;"><br/></p>
                <table style="border-collapse:collapse;margin-
left:24.844pt" cellspacing="0"><tr style="height:15pt"><td
style="width:186pt"><p class="s5" style="padding-left: 2pt;text-indent:
0pt;line-height: 11pt;text-align: left;">Introduction to Soft
Computing</p></td><td style="width:119pt"><p class="s6" style="padding-
left: 15pt;text-indent: 0pt;line-height: 11pt;text-align: left;">April
2023</p></td><td style="width:179pt"><p class="s7" style="padding-right:
3pt;text-indent: 0pt;line-height: 11pt;text-align:
right;">NPTEL</p></td></tr><tr style="height:19pt"><td
style="width:186pt"><p class="s5" style="padding-top: 1pt;padding-left:
2pt;text-indent: 0pt;text-align: left;">Data Science for
Engineers</p></td><td style="width:119pt"><p class="s6" style="padding-
top: 1pt;padding-left: 14pt;text-indent: 0pt;text-align: left;">April
2023</p></td><td style="width:179pt"><p class="s7" style="padding-top:
1pt;padding-right: 3pt;text-indent: 0pt;text-align:
right;">NPTEL</p></td></tr><tr style="height:21pt"><td
```

```
style="width:186pt"><p class="s5" style="padding-top: 2pt;padding-left:
2pt;text-indent: 0pt;text-align: left;">SOI MUMPS Microfabrication
Process</p></td><td style="width:119pt"><p class="s6" style="padding-top:
2pt;padding-left: 13pt;text-indent: 0pt;text-align: left;">March
2023</p></td><td style="width:179pt"><p class="s7" style="padding-top:
2pt;padding-right: 2pt;text-indent: 0pt;text-align: right;">Shiv Nadar
University Chennai</p>
        </td>
        </tr>
            <tr style="height:16pt"><td style="width:186pt"><p class="s5"
style="padding-top: 2pt;padding-left: 2pt;text-indent: 0pt;line-height:
12pt;text-align: left;">Codeforces</p></td><td style="width:119pt"><p
class="s7" style="padding-top: 2pt;padding-left: 11pt;text-indent:
0pt;line-height: 12pt;text-align: left;">Max Rated 848</p></td><td
style="width:179pt"><p class="s7" style="padding-top: 2pt;padding-right:
3pt;text-indent: 0pt;line-height: 12pt;text-align: right;">Handle-
ashtrichh</p></td></tr></table></body></html>
```

Output:

Github Link: https://github.com/AsHtrich/Web_tech2023



Amoghavarsh Hiremath

Karnataka,India

E-mail: amoghash80@gmail.com Phone number: +91 6360803602

Education

B-Tech: Computer Science with Specialization in IoT Shiv Nadar University Chennai Bachelor's degree program (4th Semester ongoing) 2021-2025

GPA: 8.6

Portfolio of most relevant project

StepWise | *Python, Streamlit, HTML-CSS,Arduino,ESP-32 Cam,OpenCV* |

- A footfall management application for analyzing the footfall data by performance metrics in a store given from the ESP-32 Camera module with Arduino by Human Detection using OpenCV.
- Aids the shopkeepers in knowing potential and performance of aisles to make marketing and advertising decisions on their products.

SocioPath | *HTML-CSS,JavaScript,ReactJS* |

- An application which helps startups find guidance, funding and support, connecting people from all the different fields to professionals alike.
- Provides an environment which supports innovations and solutions to grow.

Traffic Alert System | *Arduino,HC-SR04,Buzzer* |

- A prototype that uses Arduino and the HC-SR04 UV module to alert inattentive drivers on road that are approaching a stop light.
- Prevents accidents and keeps people from braking the rules.

Skills

Technical/Tools

C, C++,Python,SQL, Java, JavaScript, HTML, CSS, Tailwind CSS, Tableau,Data Science and Analytics, Data Structures and Algorithms,Competitive Programming, Arduino, Raspberry Pi

General/Tools

Problem Solving, Communication, Flexibility

Result:

Therefore, we've successfully created a CV using HTML.

Ex. No: 2	CSS enabled CV
20.07.2023	

Aim:
To apply CSS to the Assignment done for LAB 1

Algorithm:
1. Create a CSS file
2. Link the CSS file to the HTML file
3. Define Styles
4. Apply Classes and IDs
5. Preview and Refine.

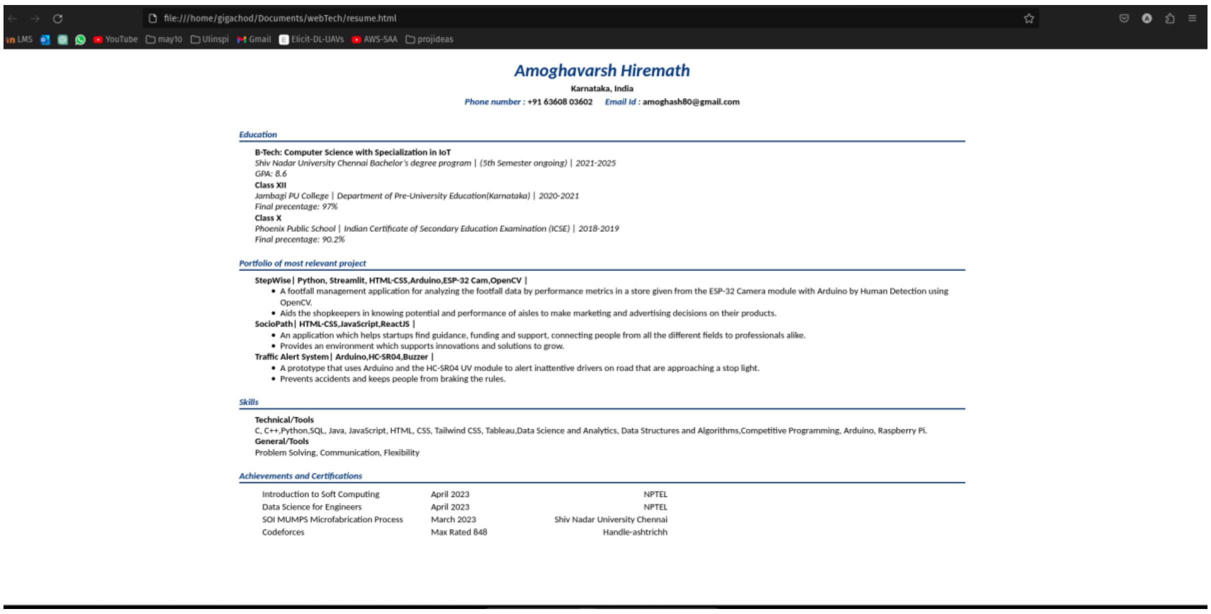
Program:
```
<!DOCTYPE html>
<html>
    <head>
        <title>ASHresume</title>
        <style>
             body {
                margin-left: 20%;
                margin-right: 20%;
                font-family:Calibri, sans-serif;
             }
             .bluetextitalics {
                color : #003579;
                font-style: italic;
             }
             .blacktextitalics {
                color: black;
                font-style: italic;
             }
             .stickit {
                margin-top: 0px; margin-bottom: 0px;
             }
             h1 {font-size: 21pt;}
             h2, h3 {font-size: 11pt; line-height: 13pt; text-align: left; }
             p {font-size: 11pt; line-height: 13pt;}

        </style>
    </head>
    <body>
        <!-- Header div -->
        <div class="stickit">
            <h1 style="text-align: center; margin-bottom: 2px ;"
class="bluetextitalics">Amoghavarsh Hiremath</h1>
            <h3 style="text-align: center; margin-top: 0px ; margin-bottom:
4px;">Karnataka, India</h3>
            <div style=" display: flex; align-items: center; justify-content:
center;">
                <h3 style="margin-right:20px;margin-top: 0px; "> <span
class="bluetextitalics">Phone number : </span>+91 63608 03602  </h3>
                <h3 style="margin-top: 0px; "> <span class="bluetextitalics
">Email Id : </span>amoghash80@gmail.com</h3>
```

```
            </div>
        </div>
        <br>
        <!-- Education -->
        <h2 class="bluetextitalics stickit" style="margin-bottom: 
0px;">Education</h2>
        <hr style="margin-top: 0px; color: #003579;">
        <div style="padding-left: 25px;">
            <h3 class="stickit">B-Tech: Computer Science with Specialization in 
IoT          </h3>
            <p class="blacktextitalics stickit" > Shiv Nadar University Chennai 
Bachelor’s degree program | (5th Semester ongoing) | 2021-2025</p>
            <p class="blacktextitalics stickit" >GPA: 8.6</p>

            <h3 class="stickit">Class XII </h3>
            <p class="blacktextitalics stickit" > Jambagi PU College | Department 
of Pre-University Education(Karnataka) | 2020-2021</p>
            <p class="blacktextitalics stickit" >Final precentage: 97%</p>

            <h3 class="stickit">Class X </h3>
            <p class="blacktextitalics stickit" >Phoenix Public School | Indian 
Certificate of Secondary Education Examination (ICSE) | 2018-2019</p>
            <p class="blacktextitalics stickit" >Final precentage: 90.2%</p>
        </div>
        <br>
        <!-- Projects -->
        <h2 class="bluetextitalics stickit" style="margin-bottom: 0px;">Portfolio 
of most relevant project</h2>
        <hr style="margin-top: 0px; color: #003579;">
        <div style="padding-left: 25px;">
            <h3 class="stickit" >StepWise<span style="text-align: right;">| 
Python, Streamlit, HTML-CSS,Arduino,ESP-32 Cam,OpenCV |</span></h3>
            <ul class="stickit ">
                <li data-list-text="">
                    <p class="stickit">A footfall management application for 
analyzing the footfall data by performance metrics in a
                        store given from the ESP-32 Camera module with Arduino by 
Human Detection using OpenCV.</p>
                </li>
                <li data-list-text="">
                    <p class="stickit">Aids the shopkeepers in knowing potential 
and performance of aisles to
                        make marketing and advertising decisions on their 
products.</p>
                </li>
            </ul>

            <h3 class="stickit" >SocioPath<span style="text-align: right;">| HTML-
CSS,JavaScript,ReactJS |</span></h3>
            <ul class="stickit ">
                <li data-list-text="">
                    <p class="stickit">An application which helps startups find 
guidance, funding and support,
                        connecting people from all the different fields to 
professionals alike.</p>
                </li>
                <li data-list-text="">
                    <p class="stickit">Provides an environment which supports 
innovations and solutions to grow.</p>
```

```
                </li>
            </ul>

            <h3 class="stickit" >Traffic Alert System<span style="text-align:
right;">| Arduino,HC-SR04,Buzzer |</span></h3>
            <ul class="stickit ">
                <li data-list-text="">
                    <p class="stickit">A prototype that uses Arduino and the HC-
SR04 UV module to alert inattentive drivers on road that are approaching a stop
light.</p>
                </li>
                <li data-list-text="">
                    <p class="stickit">Prevents accidents and keeps people from
braking the rules.</p>
                </li>
            </ul>
        </div>
        <br>
        <!-- skills -->
        <h2 class="bluetextitalics stickit" style="margin-bottom:
0px;">Skills</h2>
        <hr style="margin-top: 0px; color: #003579;">
        <div style="padding-left: 25px;">
            <h3 class="stickit" >Technical/Tools</h3>
            <p class="stickit">C, C++,Python,SQL, Java, JavaScript, HTML, CSS,
Tailwind CSS, Tableau,Data Science and Analytics,
                Data Structures and Algorithms,Competitive Programming, Arduino,
Raspberry Pi.</p>
            <h3 class="stickit" >General/Tools</h3>
            <p class="stickit">Problem Solving, Communication, Flexibility</p>
        </div>
        <br>
        <!-- Achievements and Certifications -->
        <h2 class="bluetextitalics stickit" style="margin-bottom:
0px;">Achievements and Certifications</h2>
        <hr style="margin-top: 0px; color: #003579;">
        <table style="border-collapse:collapse;margin-left:24.844pt"
cellspacing="0">
            <tr style="height:15pt">
                <td style="width:186pt">
                    <p class="stickit" style="padding-left: 2pt;text-align:
left;">Introduction to Soft Computing</p>
                </td>
                <td style="width:119pt">
                    <p class="stickit" style="padding-left: 15pt;text-align:
left;">April 2023</p>
                </td>
                <td style="width:179pt">
                    <p class="stickit" style="padding-right: 3pt;text-align:
right;">NPTEL</p>
                </td>
            </tr>

            <tr style="height:15pt">
                <td style="width:186pt">
                    <p class="stickit" style="padding-left: 2pt;text-align:
left;">Data Science for Engineers</p>
                </td>
                <td style="width:119pt">
```

```
                    <p class="stickit" style="padding-left: 15pt;text-align:
left;">April 2023</p>
                </td>
                <td style="width:179pt">
                    <p class="stickit" style="padding-right: 3pt;text-align:
right;">NPTEL</p>
                </td>
            </tr>
            <tr style="height:15pt">
                <td style="width:186pt">
                    <p class="stickit" style="padding-left: 2pt;text-align:
left;">SOI MUMPS Microfabrication Process</p>
                </td>
                <td style="width:119pt">
                    <p class="stickit" style="padding-left: 15pt;text-align:
left;">March 2023</p>
                </td>
                <td style="width:179pt">
                    <p class="stickit" style="padding-right: 3pt;text-align:
right;">Shiv Nadar University Chennai</p>
                </td>
            </tr>
            <tr style="height:15pt">
                <td style="width:186pt">
                    <p class="stickit" style="padding-left: 2pt;text-align:
left;">Codeforces</p>
                </td>
                <td style="width:119pt">
                    <p class="stickit" style="padding-left: 15pt;text-align:
left;">Max Rated 848</p>
                </td>
                <td style="width:179pt">
                    <p class="stickit" style="padding-right: 3pt;text-align:
right;">Handle-ashtrichh</p>
                </td>
            </tr>
        </table>
    </body>
</html>
```
Output:

Github Link: https://github.com/AsHtrich/Web_tech2023



file:///home/gigachod/Documents/webTech/resume.html

Amoghavarsh Hiremath

Karnataka, India

Phone number : **+91 63608 03602** *Email Id :* **amoghash80@gmail.com**

Education

B-Tech: Computer Science with Specialization in IoT
Shiv Nadar University Chennai Bachelor's degree program | (5th Semester ongoing) | 2021-2025
GPA: 8.6
Class XII
Jambagi PU College | Department of Pre-University Education(Karnataka) | 2020-2021
Final precentage: 97%
Class X
Phoenix Public School | Indian Certificate of Secondary Education Examination (ICSE) | 2018-2019
Final precentage: 90.2%

Portfolio of most relevant project

StepWise | Python, Streamlit, HTML-CSS,Arduino,ESP-32 Cam,OpenCV |
- A footfall management application for analyzing the footfall data by performance metrics in a store given from the ESP-32 Camera module with Arduino by Human Detection using OpenCV.
- Aids the shopkeepers in knowing potential and performance of aisles to make marketing and advertising decisions on their products.

SocioPath | HTML-CSS,JavaScript,ReactJS |
- An application which helps startups find guidance, funding and support, connecting people from all the different fields to professionals alike.
- Provides an environment which supports innovations and solutions to grow.

Traffic Alert System | Arduino,HC-SR04,Buzzer |
- A prototype that uses Arduino and the HC-SR04 UV module to alert inattentive drivers on road that are approaching a stop light.
- Prevents accidents and keeps people from braking the rules.

Skills

Technical/Tools
C, C++,Python,SQL, Java, JavaScript, HTML, CSS, Tailwind CSS, Tableau,Data Science and Analytics, Data Structures and Algorithms,Competitive Programming, Arduino, Raspberry Pi.
General/Tools
Problem Solving, Communication, Flexibility

Achievements and Certifications

Introduction to Soft Computing	April 2023	NPTEL
Data Science for Engineers	April 2023	NPTEL
SOI MUMPS Microfabrication Process	March 2023	Shiv Nadar University Chennai
Codeforces	Max Rated 848	Handle-ashtrichh

Result:

Therefore, we've successfully implemented the creation of Thread using C.

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 HTML and CSS file
2. Setup JavaScript in the HTML file
3. Access the form elements
4. Setup a validation logic
5. Use the Alert(), confirm() and response() functions for validation
6. Display validation results

Program:

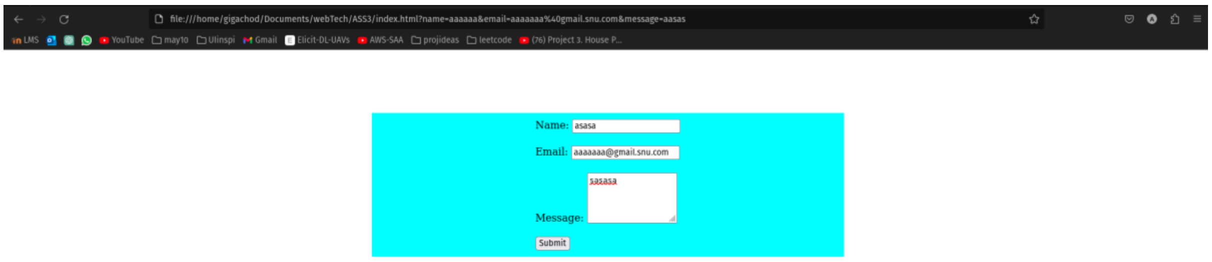
```
HTML
<!DOCTYPE html>
<html>
<head>
  <title>Form with JavaScript Functions</title>
  <style>
    .error {
      color: red;
    }
    .padit {
        padding: 10px;
    }
  </style>

</head>
<body>
    <div style="display:flex; background-color: aqua; justify-content: 
center; max-width:750px ;
    max-height:750px; margin-left: auto; margin-right: auto; margin-top: 
100px; margin-bottom: auto;">
        <form id="myForm">
            <div class="padit">
              <label for="name">Name:</label>
              <input type="text" id="name" name="name" required>
              <span id="nameError" class="error"></span>
            </div>
            <div class="padit">
              <label for="email">Email:</label>
              <input type="email" id="email" name="email" required>
              <span id="emailError" class="error"></span>
            </div>
            <div class="padit">
              <label for="message">Message:</label>
              <textarea id="message" name="message" rows="4" cols="17"
required></textarea>
              <span id="messageError" class="error"></span>
            </div>
            <div class="padit">
              <button type="submit">Submit</button>
            </div>
          </form>
    </div>

    <!-- run validation on #id "myForm" -->
  <script src="index.js"></script>
</body>
</html>
```

Javascript

```
document.getElementById('myForm').addEventListener('submit',
function(event)
{

    const name = document.getElementById('name').value.trim();
    const email = document.getElementById('email').value.trim();
    const message = document.getElementById('message').value.trim();

    const nameError = document.getElementById('nameError');
    const emailError = document.getElementById('emailError');
    const messageError = document.getElementById('messageError');

    let isValid = true;

    function isValidEmail(email) {
        const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+\.[^\s@]+$/;
        return emailRegex.test(email);
      }

    if (name === '') {
      nameError.textContent = 'Name is required';
      isValid = false;
    } else {
      nameError.textContent = '';
    }

    if (email === '') {
      emailError.textContent = 'Email is required';
      isValid = false;
    } else if (!isValidEmail(email)) {
      emailError.textContent = 'Invalid email address';
      isValid = false;
    } else {
        messageError.textContent = '';
      }
    if (message === '') {
      messageError.textContent = 'Message is required';
      isValid = false;
    } else {
      messageError.textContent = '';
    }

    const confirmed = confirm('Are you sure you want to submit the
form?');
    if (confirmed) {
      alert('Form submitted successfully!');
    }
    if (!isValid) {
        event.preventDefault();
        return;
      }
  });
```

Output:

Github Link: https://github.com/AsHtrich/Web_tech2023

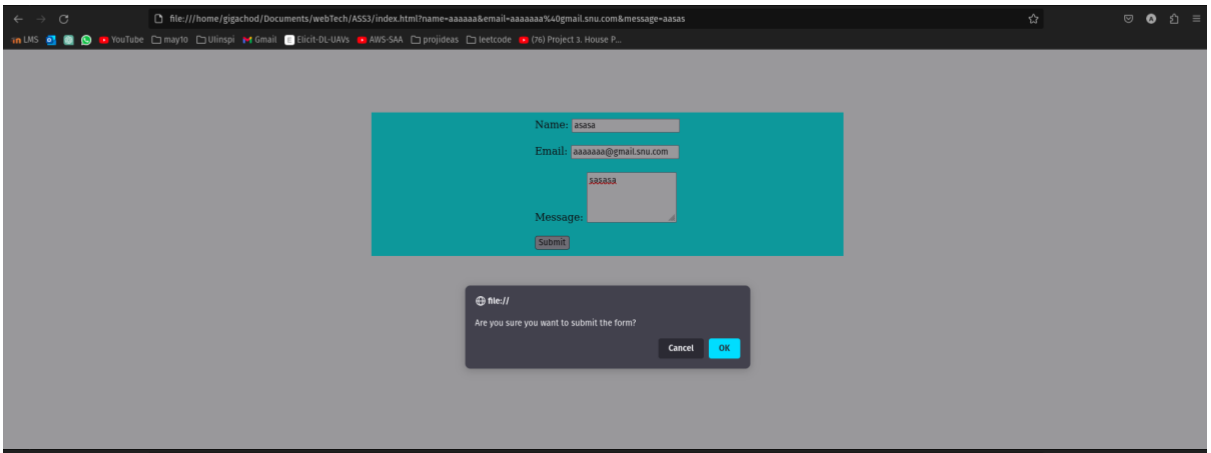


file:///home/gigachod/Documents/webTech/ASS3/index.html?name=aaaaaa&email=aaaaaaa%40gmail.snu.com&message=aasas

Name: asasa

Email: aaaaaaa@gmail.snu.com

Message: sasasa


file:///home/gigachod/Documents/webTech/ASS3/index.html?name=aaaaaa&email=aaaaaaa%40gmail.snu.com&message=aasas

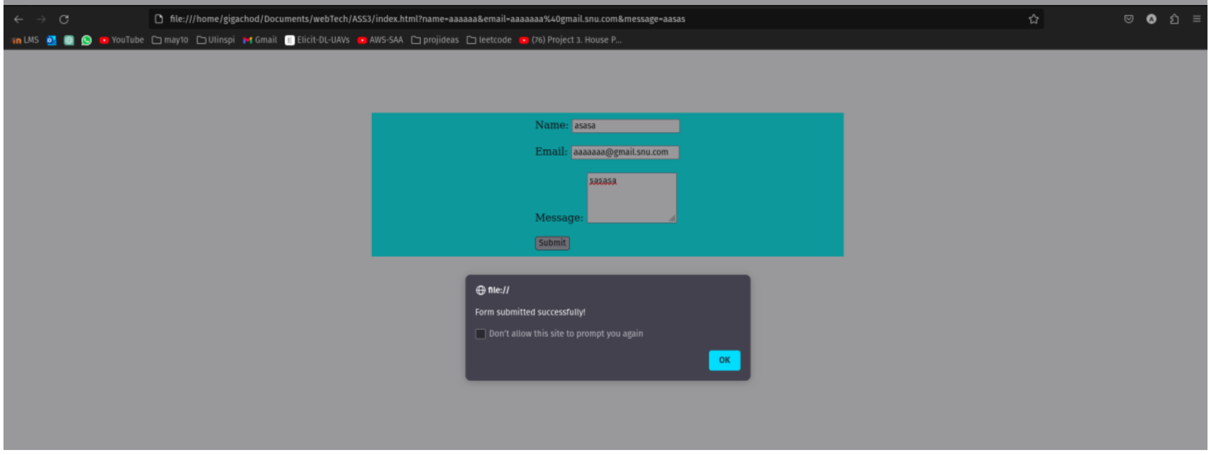
Name: asasa

Email: aaaaaaa@gmail.snu.com

Message: sasasa

file://

Are you sure you want to submit the form?


file:///home/gigachod/Documents/webTech/ASS3/index.html?name=aaaaaa&email=aaaaaaa%40gmail.snu.com&message=aasas

Name: asasa

Email: aaaaaaa@gmail.snu.com

Message: sasasa

file://

Form submitted successfully!

Don't allow this site to prompt you again

Result:
Therefore, created a form and validated it using javascript.

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. Setup angular using the ng serve command
2. Create all the required components.
3. Organize the app structure.
4. Implement the services that are needed.
5. Define component HTML templates with data binding to display dynamic content
6. Enable component communication using input/output properties and event binding.
7. Apply CSS styles to components, optimize for performance, and deploy the app.

Program:

Component code:

```
import { Component, Input, Output, EventEmitter } from "@angular/core";
import { Item } from "../item";


@Component({
  selector: 'app-item',
  templateUrl: './item.component.html',
  styleUrls: ['./item.component.css'],
})

export class ItemComponent {

  editable = false;

  @Input() item!: Item;
  @Output() remove = new EventEmitter<Item>();

  saveItem(description: string) {
    if (!description) return;
    this.editable = false;
    this.item.description = description;
  }
}
```

Service code:

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { ItemComponent } from './item/item.component';
```

```
@NgModule({
  declarations: [
    AppComponent,
    ItemComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

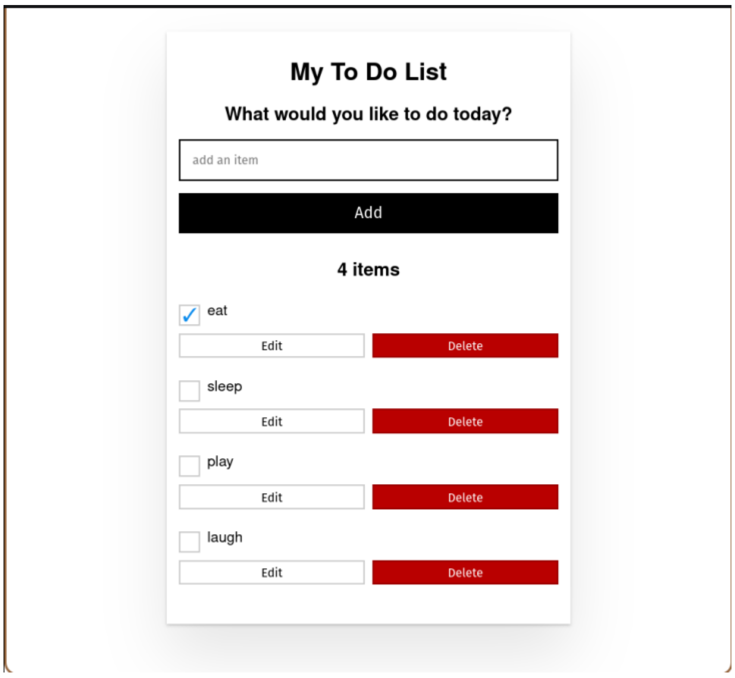
app.component.ts:

```
import { Component } from "@angular/core";
import { Item } from "./item";

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

export class AppComponent {
  title = "todo";
  filter: "all" | "done" = "all";
  allItems = [
    { description: "eat", done: true },
    { description: "sleep", done: false },
    { description: "play", done: false },
    { description: "laugh", done: false },
  ];
  get items() {
    if (this.filter === "all") {
      return this.allItems;
    }
    return this.allItems.filter((item) =>
      this.filter === "done" ? item.done : !item.done
    );
  }
  addItem(description: string) {
    this.allItems.unshift({
      description,
      done: false
    });
  }
  remove(item: Item) {
    this.allItems.splice(this.allItems.indexOf(item), 1);
  }
}
```

Component html code:

```
<div class="main">
  <h1>My To Do List</h1>
  <label for="addItemInput">What would you like to do today?</label>

  <input
   #newItem
   placeholder="add an item"
   (keyup.enter)="addItem(newItem.value); newItem.value = ''"
   class="lg-text-input"
    id="addItemInput" />

  <button class="btn-primary"
(click)="addItem(newItem.value)">Add</button>


  <h2>
    {{items.length}}
    <span *ngIf="items.length === 1; else elseBlock">item</span>
    <ng-template #elseBlock>items</ng-template>
  </h2>

  <ul>
    <li *ngFor="let i of items">
      <app-item (remove)="remove(i)" [item]="i"></app-item>
    </li>
  </ul>

</div>
```

Output:
Github Link: https://github.com/AsHtrich/Web_tech2023



Result:
Therefore, we've successfully created a simple react app.

Ex. No: 5	**React based App Development**
23.02.2023	

Aim:

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

Algorithm:

1. Create a new React project using a tool like Create React App.
2. Build individual components to represent various parts of your app.
3. Arrange components hierarchically, defining parent-child relationships.
4. Define the UI using JSX within components for rendering.
5. Pass data between components using props or consider state management for more complex data sharing.
6. Implement local or global state management for dynamic data and user interactions.
7. Style components using CSS, CSS modules, or CSS-in-JS libraries while keeping styling separate from logic.

Program:

App.js

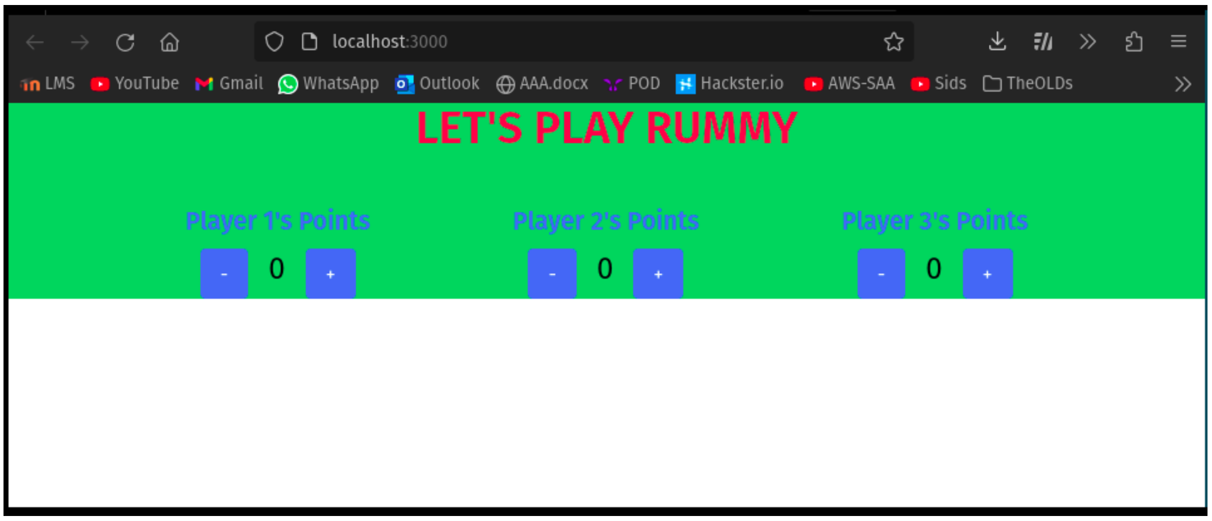
```
import React from 'react';
import PlayerCounter from './components/PlayerCounter';
import './input.css';


function App() {
  return (
    <div class="bg-green-500 w-4xl ">

      <div className="App max-w-2xl mx-auto justify-between items-center
flex flex-col">
        <div >
          <h1 className='font-semibold text-4xl text-red-500 '>LET'S PLAY
RUMMY</h1>
        </div>
      </div>


      <div className="max-w-2xl mx-auto justify-between flex flex-row">
        <div>
        <PlayerCounter player="Player 1" />
        </div>
        <div>
        <PlayerCounter player="Player 2" />
        </div>
        <div>
        <PlayerCounter player="Player 3" />
        </div>
      </div>
    </div>

  );
}
export default App;
```

PlayerCounter.jsx Component

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


function PlayerCounter({ player }) {
  const [points, setPoints] = useState(0);

  const incrementPoints = () => {
    setPoints(points + 10);
  };

  const decrementPoints = () => {
    setPoints(points - 10);
  };

  return (
    <div className="flex flex-col items-center mt-10">
      <h1 className="text-xl font-semibold  text-blue-500 mb-2">{player}'s
Points</h1>
      <div className="flex  space-x-4">
        <button
          className="px-4 py-2 bg-blue-500 text-white rounded"
          onClick={decrementPoints}
        >
          -
        </button>
        <span className="text-2xl">{points}</span>
        <button
          className="px-4 py-2 bg-blue-500 text-white rounded"
          onClick={incrementPoints}
        >
          +
        </button>
      </div>
    </div>
  );
}

export default PlayerCounter
```

Output:

Github Link: https://github.com/AsHtrich/Web_tech2023



Result:
Therefore, we've successfully created a simple react app.

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. Ensure you have Node.js installed on your system.
2. Develop a JavaScript file (e.g., `server.js`) for your web server.
3. In `server.js`, require Node.js's built-in `http` module using `require('http')`.
4. Use the `http.createServer()` method to create an HTTP server, specifying a request handling function.
5. Inside the request handling function, use the `request` and `response` objects to define how your server should respond to different routes and HTTP methods.
6. Test your web server using tools like cURL or Postman. Debug and refine your route handling as needed.
7. Optionally, configure the web server to serve static HTML, CSS, and JavaScript files if your front-end includes them, using the `fs` (file system) module.

Program:

Server.js
```
const http = require("http");
const fs = require("fs");
const path = require("path");
const url = require("url");

const server = http.createServer((req, res) => {
    const reqUrl = url.parse(req.url, true);

    if (reqUrl.pathname === "/" || reqUrl.pathname === "/index.html") {
        // Serve the HTML page
        fs.readFile(path.join(__dirname, "public", "index.html"), (err,
data) => {
            if (err) {
                res.writeHead(500, { "Content-Type": "text/plain" });
                res.end("Internal Server Error");
            } else {
                res.writeHead(200, { "Content-Type": "text/html" });
                res.end(data);
            }
        });
    } else if (reqUrl.pathname === "/styles.css") {
        // Serve the CSS file
        fs.readFile(path.join(__dirname, "public", "styles.css"), (err,
data) => {
            if (err) {
                res.writeHead(500, { "Content-Type": "text/plain" });
                res.end("Internal Server Error");
            } else {
                res.writeHead(200, { "Content-Type": "text/css" });
                res.end(data);
            }
        });
    } else if (reqUrl.pathname === "/script.js") {
        // Serve the JavaScript file
        fs.readFile(path.join(__dirname, "public", "script.js"), (err,
data) => {
            if (err) {
                res.writeHead(500, { "Content-Type": "text/plain" });
                res.end("Internal Server Error");
            } else {
                res.writeHead(200, { "Content-Type": "text/javascript" });
                res.end(data);
            }
        });
    } else if (reqUrl.pathname === "/notes" && req.method === "GET") {
        // Handle GET request to retrieve notes (simulated in-memory
storage)
        const notes = [
            { id: 1, text: "Buy groceries" },
            { id: 2, text: "Call John" },
        ];
        res.writeHead(200, { "Content-Type": "application/json" });
        res.end(JSON.stringify(notes));
    } else {
        // Handle other routes with a 404 Not Found response
        res.writeHead(404, { "Content-Type": "text/plain" });
        res.end("Not Found");
    }
});

const port = process.env.PORT || 3000;
server.listen(port, () => {
    console.log(`Server is running on port ${port}`);
});
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Server-Side Notes</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<h1>Server-Side Notes</h1>
<div class="notes-container">
<textarea id="noteInput" placeholder="Add a new note">
</textarea>
<button id="addNote">Add Note</button>
</div>
<div class="notes-list">
```

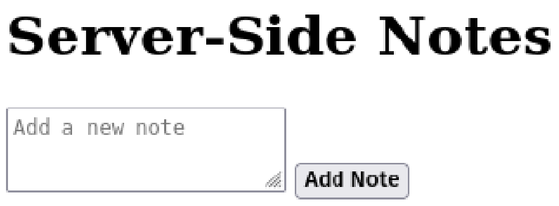
```
</div>
<script</body>
</html>
src="script.js"></script>
```

Script.js

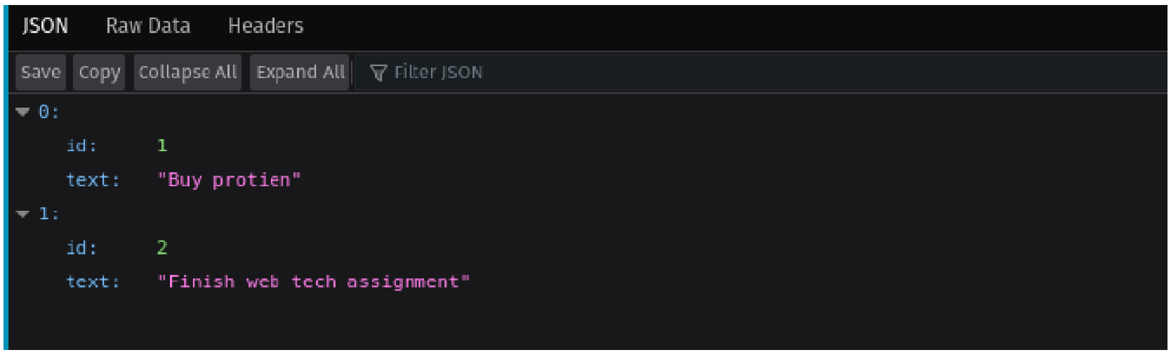
```
document.addEventListener("DOMContentLoaded", () => {
const noteInput = document.getElementById("noteInput");
const addNoteButton = document.getElementById("addNote");
const notesList = document.querySelector(".notes-list");
// Fetch and display notes
fetch("/notes")
.then((response) => response.json())
.then((notes) => {
notes.forEach((note) => {
displayNote(note);
});
})
.catch((error) => {
console.error("Error fetching notes:", error);
});
// Add a new note
addNoteButton.addEventListener("click", () => {
const text = noteInput.value.trim();
if (text) {
fetch("/notes", {
method: "POST",
headers: {
"Content-Type": "application/json",
},
body: JSON.stringify({ text }),
})
.then((response) => response.json())
.then((newNote) => {
displayNote(newNote);
noteInput.value = "";
})
.catch((error) => {
console.error("Error adding note:", error);
});
}
});
// Display a note
function displayNote(note) {
const noteElement = document.createElement("div");
noteElement.className = "note";
noteElement.textContent = note.text;
notesList.appendChild(noteElement);
}
});
```

Output:

Github Link: https://github.com/AsHtrich/Web_tech2023



Server Side output:



```
0:
    id:     1
    text:   "Buy protien"
1:
    id:     2
    text:   "Finish web tech assignment"
```

Result:
Therefore, we've successfully implemented a web server backend using NodeJS .

Ex. No: 7	**Routing Implementation using ExpressJS**
28.09.2023	

Aim: To Implement the routing feature(s) using the ExpressJS.

Algorithm:

1. Include the required header files thread creation and sleep() function.
2. Write a function that executes as a thread when it is called. (sleep – print - return)
3. thread_id is declared to identify the thread in the system, we call pthread_create() function to create a thread.
4. The pthread_join() function for threads is the equivalent of wait() for processes. A call to pthread_join blocks the calling thread until the thread with identifier equal to the first argument terminates.

Program:

```
const express = require('express');
const app = express();
const port = 3000;
app.use(express.json());
let notes = [];
app.use(express.static('public'));
app.use((req, res, next) => {
console.log(`Received ${req.method} request at
${req.url}`);
next();
});
app.get('/api/notes', (req, res) => {
res.json(notes);
});
app.post('/api/notes', (req, res) => {
const { title, content } = req.body;
const newNote = { id: notes.length + 1, title, content };
notes.push(newNote);
console.log(`Added a new note: "${title}"`);
res.status(201).json(newNote);
});
app.delete('/api/notes/:id', (req, res) => {
const idToDelete = parseInt(req.params.id);
notes = notes.filter(note => note.id !== idToDelete);
console.log(`Deleted note with ID: ${idToDelete}`);
res.sendStatus(204);
});
app.listen(port, () => {
console.log(`Server is running on port ${port}`);
});
<!DOCTYPE html>
```

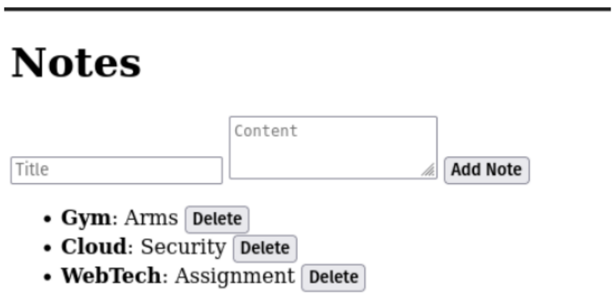
```
<html>
<head>
<title>Note Taking App</title>
</head>
<body>
<h1>Notes</h1>
<form id="noteForm">
<input type="text" id="noteTitle"
placeholder="Title" required>
<textarea id="noteContent" placeholder="Content"
required></textarea>
<button type="submit">Add Note</button>
</form>
<ul id="noteList"></ul>
<script>
const noteForm =
document.getElementById('noteForm');
const noteTitle =
document.getElementById('noteTitle');
const noteContent =
document.getElementById('noteContent');
const noteList =
document.getElementById('noteList');
noteForm.addEventListener('submit', async (e) => {
e.preventDefault();
const title = noteTitle.value;
const content = noteContent.value;
if (!title || !content) return;
try {
const response = await fetch('/api/notes',
{
method: 'POST',
headers: {
'Content-Type':
'application/json',
},
body: JSON.stringify({ title, content
}),
});
const newNote = await response.json();
noteTitle.value = '';
noteContent.value = '';
displayNote(newNote);
} catch (error) {
console.error('Error adding note:',
error);
}
});
async function fetchNotes() {
try {
```

```
const response = await
fetch('/api/notes');
const notes = await response.json();
notes.forEach(displayNote);
} catch (error) {
console.error('Error fetching notes:',
error);
}
}
function displayNote(note) {
const listItem = document.createElement('li');
listItem.innerHTML = `<strong>${note.title}
</strong>: ${note.content} <button>Delete</button>`;
const deleteButton =
listItem.querySelector('button');
deleteButton.addEventListener('click', async
() => {
try {
await fetch(`/api/notes/${note.id}`, {
method: 'DELETE' });
listItem.remove();
} catch (error) {
console.error('Error deleting note:',
error);
}
});
noteList.appendChild(listItem);
}
// Initial fetch
fetchNotes();
</script>
</body>
</html>
```

Output:

Github Link: https://github.com/AsHtrich/Web_tech2023

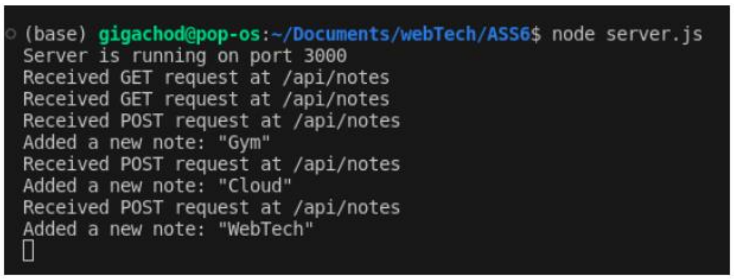
1. Initial webpage when the server is started.



Notes

- **Gym**: Arms Delete
- **Cloud**: Security Delete
- **WebTech**: Assignment Delete

2. All the content for that session can be assesed in the /api/notes route in json formate



```
(base) gigachod@pop-os:~/Documents/webTech/ASS6$ node server.js
Server is running on port 3000
Received GET request at /api/notes
Received GET request at /api/notes
Received POST request at /api/notes
Added a new note: "Gym"
Received POST request at /api/notes
Added a new note: "Cloud"
Received POST request at /api/notes
Added a new note: "WebTech"
```

Result:
Therefore, we've successfully implemented a basic routing implementation using Express JS

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

Aim:

To create a REST API with express node and mongoDB.

Algorithm:

1. Ensure Node.js, npm, and MongoDB are installed on your system.
2. Create a project directory and set up its structure.
3. Use npm to install necessary packages, including Express and a MongoDB driver like Mongoose.
4. Create API routes and handlers for various HTTP methods to manage different data operations.
5. Establish a connection to your MongoDB database using the installed MongoDB driver.
6. Define data models and schemas to structure the data you'll work with in the MongoDB database.
7. Implement Create, Read, Update, and Delete (CRUD) operations in your API routes for database interaction.
8. Test API endpoints using tools like Postman. Debug, refine, and handle errors as needed.

Program:

1) Index.js (server):
• Connecting to mongo dB, mongoose and express

```
const express=require("express");
const mongoose=require("mongoose");
const url='mongodb://127.0.0.1:27017/studentDB';
const app=express();
mongoose.connect(url,{
useNewUrlParser:true
})
const con =mongoose.connection
con.on('open',function(){
console.log("connected to mongodb database")
})
app.use(express.json())
const studentRouter=require('./routes/students')
app.use('/students',studentRouter)
app.listen(3000,function(){
console.log("Server started")
})
```

2) students.js
• Creating routes (GET, PATCH, GET single object by ID, POST)

```
const express=require("express");
const router=express.Router()
const Student=require('../models/student')
```

```
router.get('/',async(req,res)=>{
try{
const stud=await Student.find()
res.json(stud)
}catch(err){
res.send("Error")
}
res.send("Get request made")
})
router.get('/:id',async(req,res)=>{
try{
const stud1=await Student.findById(req.params.id)
res.json(stud1)
}catch(err){
res.send("Error")
}
})
router.patch('/:id', async (req, res) => {
try {
const studPatch = await Student.findById(req.params.id);
studPatch.name = req.body.name;
const s = await studPatch.save();
res.json(studPatch);
} catch (err) {
res.status(500).send("Error"); // Sending an error response
}
});
router.post('/',async(req,res)=>{
const student=new Student({
name: req.body.name,
course: req.body.course
})
try{
const s=await student.save()
res.json(s)
}catch(err){
res.send("Error")
}
})
module.exports=router
```

3) student.js

- **Creating mongoose schema for a single object (student here)**

```
const mongoose=require("mongoose")
const studSchema=new mongoose.Schema({
name:{
type: String,
required:true
},
course:{
type: String,
required:true
}
})
```

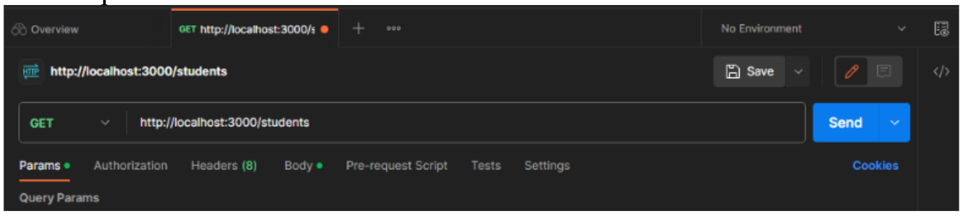
```
module.exports=mongoose.model('Student',studSchema)
```

Output:

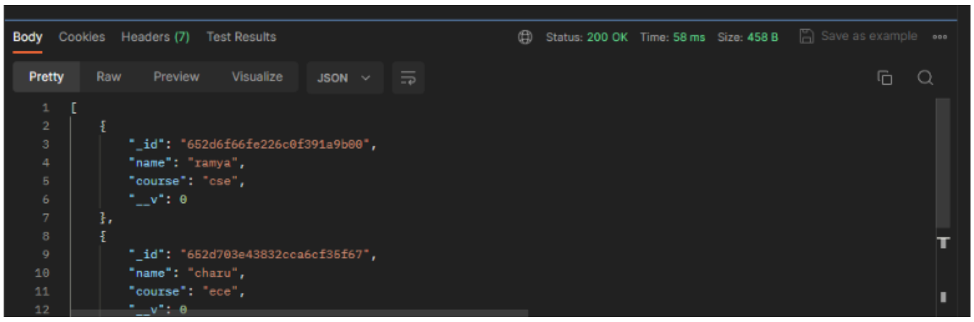
```
Github Link:
```

Request made by the postman API:

1. Get Request



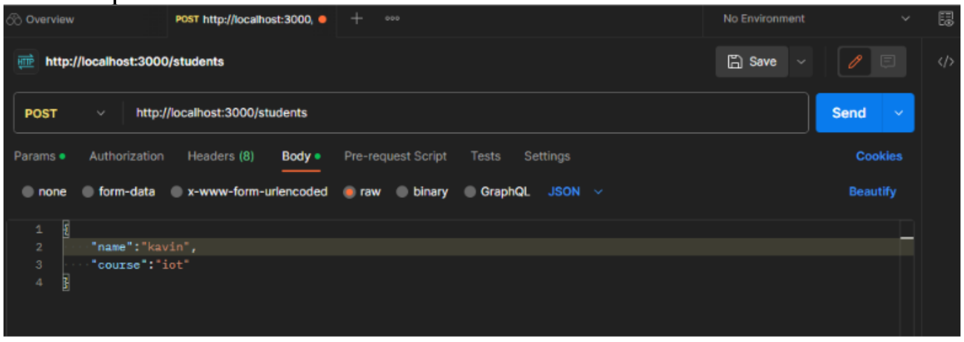
GET http://localhost:3000/students



Status: 200 OK Time: 58 ms Size: 458 B

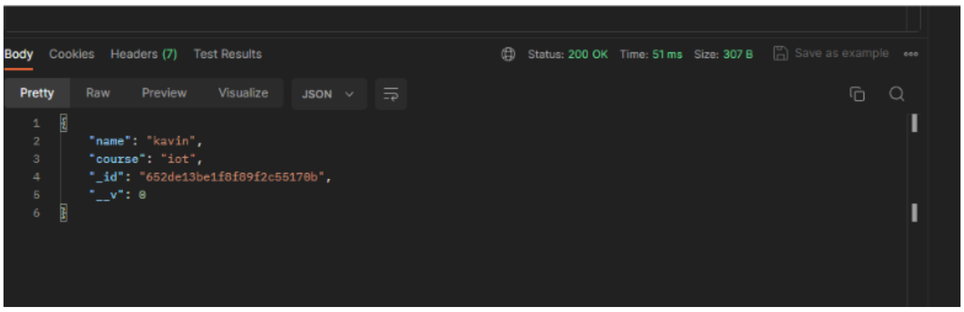
```
[
    {
        "_id": "652d6f66fe226c0f391a9b00",
        "name": "ramya",
        "course": "cse",
        "__v": 0
    },
    {
        "_id": "652d703e43832cca6cf36f67",
        "name": "charu",
        "course": "ece",
        "__v": 0
```

2. Post Request



POST http://localhost:3000/students

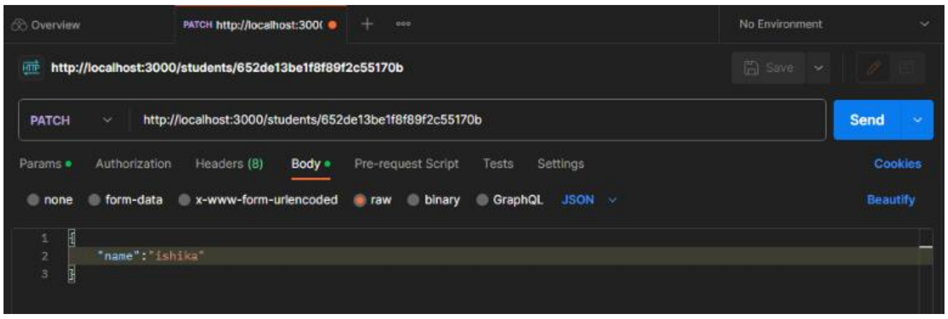
```
{
    "name":"kavin",
    "course":"iot"
}
```



Status: 200 OK Time: 51 ms Size: 307 B

```
{
    "name": "kavin",
    "course": "iot",
    "_id": "652de13be1f8f89f2c55170b",
    "__v": 0
}
```

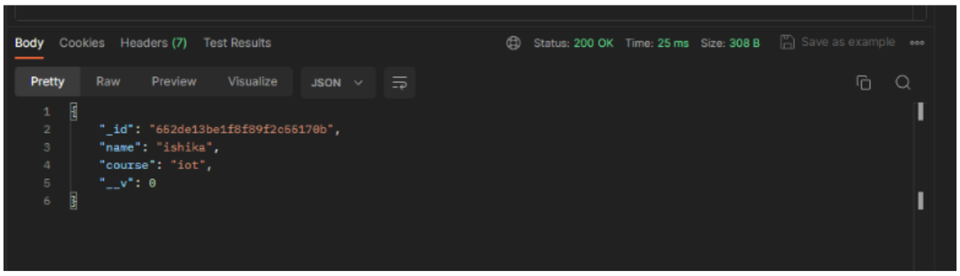
3. Patch Request



http://localhost:3000/students/652de13be1f8f89f2c55170b

PATCH http://localhost:3000/students/652de13be1f8f89f2c55170b

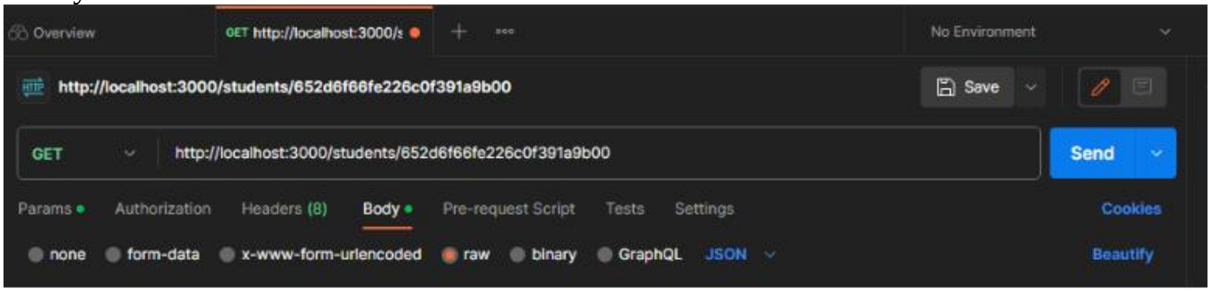
```
{
    "name":"ishika"
}
```



Status: 200 OK Time: 25 ms Size: 308 B

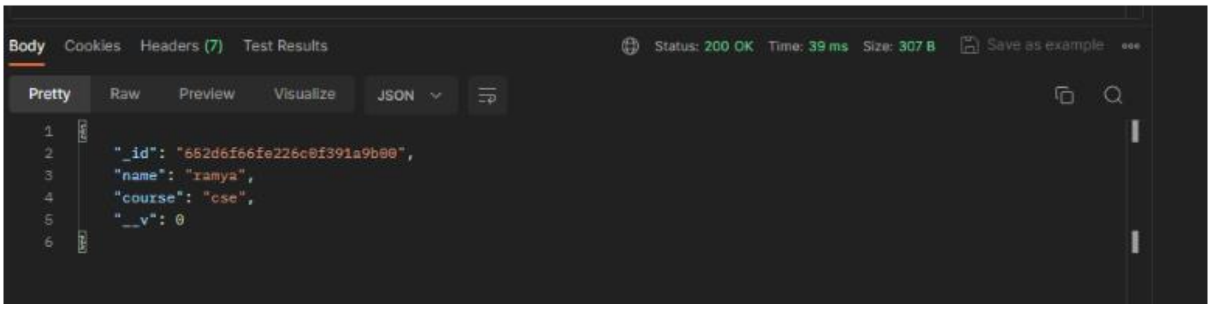
```
{
    "_id": "652de13be1f8f89f2c55170b",
    "name": "ishika",
    "course": "iot",
    "__v": 0
}
```

4. Get by ID



http://localhost:3000/students/652d6f66fe226c0f391a9b00

GET http://localhost:3000/students/652d6f66fe226c0f391a9b00



Status: 200 OK Time: 39 ms Size: 307 B

```
{
    "_id": "652d6f66fe226c0f391a9b00",
    "name": "ramya",
    "course": "cse",
    "__v": 0
}
```

Result:
Therefore, we've successfully implemented the creation of a REST API with express node and mongoDB.