

SHIV NADAR
—UNIVERSITY—
CHENNAI

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF ENGINEERING**

LABORATORY RECORD

**B.TECH
(YEAR : 20 - 20)**

NAME : *JOE MARIAN A*

REG. NO. : *22011102034*

DEPT. : *B.Tech CSE* SEM. : *V* CLASS & SEC : *TOT - A*

SHIV NADAR
—UNIVERSITY—
CHENNAI

BONAFIDE CERTIFICATE

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

WEB TECHNOLOGIES Laboratory by

Name JOE MARIAN A

Register Number 22011102034

Semester V Class & Sec. ToT - A

Branch B.Tech CSE

SHIV NADAR UNIVERSITY Chennai

During the Academic year 2024 - 2025

Faculty

M.Amala
29/10/24

Head of the Department

Submitted for the WEB TECHNOLOGIES Practical Examination held at
SNU CHENNAI on

Internal Examiner

External Examiner

INDEX

Name : JOE MARIAN A Reg. No. 22011102034

Sem : V Class & Sec : IoT - A

Ex. No.	Date of Expt.	Title of the Experiment	Page No.	Signature of the Faculty	Remarks
1	6/8/24	Creation of a webpage	1		
2	13/8/24	ReactJS program to switch between layouts	11		
3	13/8/24	ReactJS program to implement search filter	17		
4	20/8/24	ReactJS program to create a form	19		
5	20/8/24	Implementation of a Navigation Menu	21		
6	27/8/24	Creation of a Video Player	23		
7	27/8/24	Creation of a Text Editor	24		
8	3/9/24	Creation of a basic HTML page	26	M. Adams 3/9/24	
9	10/9/24	Form Validation	28		
10	17/9/24	Fetching data from Server with AJAX	30		
11	1/10/24	Implementation of a Semester Fee Form	32		

ASSIGNMENT – 1 **CREATION OF A WEBSITE**

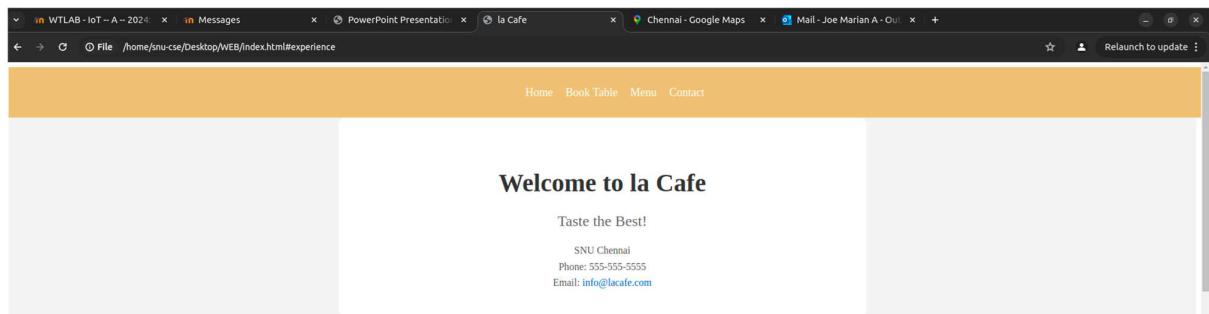
AIM:

To create a website with HTML, CSS and JavaScript.

ALGORITHM:

- Step 1: Create a html and css file.
- Step 2: Link the css file to the main html file.
- Step 3: Give the code in the html and css file
- Step 4: Run the code in web browser.
- Step 5: Close the program.

Website Images:



Our Menu

Soup

Smoked Tomato Basil Classic / Chicken - 225/255
Velvety-smooth and smoky tomato soup, seasoned with fresh Italian herbs and Parmesan cheese

★ Roasted Pumpkin Soup -245
Oven toasted red pumpkin blended with a dash of cream. Topped with pumpkin seeds & a swirl of cream

Wonton Noodle soup - 295
Crispy wonton pocket infused spupy broth with crispy rice noodles and a swirl of house made chilli oil

French Onion Classic / Chicken - 265/315
A classic French soup made with golden-brown caramelized onions, topped with crisp garlic bread and gooey mozzarella cheese

Cream Of Mushroom / Chicken - 265/295
Bringing back an old favourite!

Broccoli Almond soup - 365
Roasted Broccoli blended with creamy almond milk

Side:
Garlic bread 35 | Cheese garlic bread 55

Smoked Tomato Basil Classic / Chicken - 225/255
Velvety-smooth and smoky tomato soup, seasoned with fresh Italian herbs and Parmesan cheese

★ Roasted Pumpkin Soup -245
Oven toasted red pumpkin blended with a dash of cream. Topped with pumpkin seeds & a swirl of cream

Wonton Noodle soup - 295
Crispy wonton pocket infused spupy broth with crispy rice noodles and a swirl of house made chilli oil

French Onion Classic / Chicken - 265/315
A classic French soup made with golden-brown caramelized onions, topped with crisp garlic bread and gooey mozzarella cheese

Cream Of Mushroom / Chicken - 265/295
Bringing back an old favourite!

Broccoli Almond soup - 365
Roasted Broccoli blended with creamy almond milk

Side:
Garlic bread 35 | Cheese garlic bread 55

SOURCE CODE:

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>la Cafe</title>
<link rel="stylesheet" href="style.css">
</head>
<meta charset="UTF-8">
<title>la Cafe</title>
```

```
<link rel="stylesheet" href="style.css">
</head>
<body>
<nav class="navbar">
<div class="navbar-container">
<ul class="nav-links">
<li><a href="#home">Home</a></li>
<li><a href="#book">Book Table</a></li>
<li><a href="#experience">Menu</a></li>
<li><a href="#contact">Contact</a></li>
</ul>
</div>
</nav>
<section class="home-section-wrapper">
<section id="home" class="home-section">
<h1>Welcome to la Cafe</h1>
<p>Taste the Best!</p>
<address>
SNU Chennai<br>
Phone: 555-555-5555<br>
Email: <a href="mailto:info@lacafe.com">info@lacafe.com</a>
</address>
</section>
</section>
<section id="book" class="book-section">
<h1>Book a Table</h1>
<p>Reserve a table at la Cafe for a unique dining experience.</p>
<form>
<label for="name">Name:</label>
<input type="text" id="name" name="name" required><br><br>
<label for="email">Email:</label>
<input type="email" id="email" name="email" required><br><br>
<label for="phone">Phone:</label>
<input type="tel" id="phone" name="phone" required><br><br>
<label for="date">Date:</label>
<input type="date" id="date" name="date" required><br><br>
<label for="time">Time:</label>
<input type="time" id="time" name="time" required><br><br>
<label for="guests">Number of Guests:</label>
<input type="number" id="guests" name="guests" required><br><br>
<label for="special-requests">Special Requests:</label>
<textarea id="special-requests" name="special-requests"></textarea><br><br>
<input type="submit" value="Book Table">
</form>
</section>
<section id="experience" class="menu-section">
<h1>Our Menu</h1>
<div class="menu-slider-container">
<div class="menu-slider active" id="menu1">

</div>
```

```
<div class="menu-slider" id="menu2">

</div>
<div class="menu-slider" id="menu3">

</div>
</div>
<div class="menu-buttons">
<div class="menu-button active" onclick="showMenu(1)"></div>
<div class="menu-button" onclick="showMenu(2)"></div>
<div class="menu-button" onclick="showMenu(3)"></div>
</div>
</section>
<section id="contact" class="contact-section">
<div class="container">
<div class="row">
<div class="col-md-6">
<div class="google-map">
<iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d248756.13130929516!2d80.04420
289216043!3d13.047473319208887!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x3a5265e
a4f7d3361%3A0x6e61a70b6863d433!2sChennai%2C%20Tamil%20Nadu!5e0!3m2!1sen!2sin!4v17229
21573556!5m2!1sen!2sin" width="600" height="450" style="border:0;" allowfullscreen=""
loading="lazy"
referrerPolicy="no-referrer-when-downgrade"></iframe>
</div>
</div>
<div class="col-md-6 contact-info">
<h2>Contact Us</h2>
<ul>
<li>
<i class="fas fa-phone"></i>
<span>Phone:</span>
<a href="tel:+1234567890">+123 456 7890</a>
</li>
<li>
<i class="fas fa-envelope"></i>
<span>Email:</span>
<a href="mailto:info@example.com">info@example.com</a>
</li>
<li>
<i class="fas fa-map-marker-alt"></i>
<span>Address:</span>
<address>
SNU Chennai
</address>
</li>
</ul>
</div>
</div>
```

```
</div>
</section>
</body>
</html>
```

CSS:

```
.navbar {
background:rgb(233, 195, 125) no-repeat center center;
background-size: cover;
padding: 10px 20px;
position: relative;
width: 100%;
height: 60px;
}

.navbar-container {
display: flex;
justify-content: center;
align-items: center;
height: 100%;
position: relative;
}

.nav-links {
display: flex;
list-style: none;
margin: 0;
padding: 0;
gap: 20px;
}

.nav-links li {
margin: 0;
}

.nav-links a {
color: white;
text-decoration: none;
font-size: 18px;
transition: color 0.3s ease;
}

.nav-links a:hover {
color: rgba(245, 245, 245, 0.925);
}

.home-section-wrapper {
width: 100%;
background-color: #f4f4f4;
}
```

```
.home-section {  
text-align: center;  
padding: 40px 20px;  
border-radius: 10px;  
max-width: 800px;  
margin: 0 auto;  
background-color: #ffffff;  
}  
  
.home-section h1 {  
font-size: 2.5em;  
color: #333;  
margin-bottom: 10px;  
}  
  
.home-section p {  
font-size: 1.5em;  
color: #666;  
margin-bottom: 20px;  
}  
  
.home-section address {  
font-style: normal;  
color: #555;  
line-height: 1.6;  
}  
  
.home-section address a {  
color: #0066cc;  
text-decoration: none;  
}  
  
.home-section address a:hover {  
text-decoration: underline;  
}  
  
.book-section {  
text-align: center;  
padding: 40px 20px;  
background-color: #ffffff;  
border-radius: 10px;  
max-width: 800px;  
margin: 0 auto;  
}  
  
.book-section h1 {  
font-size: 2.5em;  
color: #333;  
margin-bottom: 20px;  
}
```

```
.book-section p {  
font-size: 1.5em;  
color: #666;  
margin-bottom: 30px;  
}  
  
.book-section form {  
text-align: left;  
margin: 0 auto;  
max-width: 600px;  
}  
  
.book-section label {  
display: block;  
font-size: 1em;  
color: #333;  
margin-bottom: 5px;  
}  
  
.book-section input[type="text"],  
.book-section input[type="email"],  
.book-section input[type="tel"],  
.book-section input[type="date"],  
.book-section input[type="time"],  
.book-section input[type="number"],  
.book-section textarea {  
width: 100%;  
padding: 10px;  
border: 1px solid #ddd;  
border-radius: 5px;  
margin-bottom: 15px;  
box-sizing: border-box;  
}  
  
.book-section textarea {  
height: 100px;  
resize: vertical;  
}  
  
.book-section input[type="submit"] {  
background-color: #0066cc;  
color: white;  
border: none;  
padding: 10px 20px;  
border-radius: 5px;  
font-size: 1em;  
cursor: pointer;  
transition: background-color 0.3s ease;  
}  
  
.book-section input[type="submit"]:hover {
```

```
background-color: #005bb5;
}

.menu-section {
position: relative;
text-align: center;
max-width: 100%;
overflow: hidden;
}

.menu-slider-container {
position: relative;
width: 80%;
margin: 0 auto;
height: 700px;
overflow: hidden;
border: 1px solid #ddd;
border-radius: 10px;
padding: 0;
}

.menu-slider {
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
transition: opacity 0.5s ease-in-out;
opacity: 0;
}

.menu-slider.active {
opacity: 1;
}

.menu-slider img {
width: 100%;
height: 100%;
object-fit: cover;
object-position: center;
border-radius: 10px;
}

.menu-buttons {
margin-top: 10px;
}

.menu-button {
display: inline-block;
width: 20px;
height: 20px;
```

```
background-color: #ccc;
border-radius: 50%;
margin: 0 5px;
cursor: pointer;
}

.menu-button.active {
background-color: #333;
}
.contact-section {
padding: 40px 20px;
background-color: #f4f4f4;
}

.container {
max-width: 1200px;
margin: 0 auto;
}

.row {
display: flex;
flex-wrap: wrap;
}

.col-md-6 {
flex: 1;
padding: 15px;
}

.google-map iframe {
width: 100%;
height: 100%;
border: 0;
}

.contact-info h2 {
font-size: 2em;
color: #333;
margin-bottom: 20px;
}

.contact-info ul {
list-style: none;
padding: 0;
margin: 0;
}

.contact-info li {
display: flex;
align-items: center;
margin-bottom: 15px;
```

```
}

.contact-info i {
  font-size: 1.5em;
  color: #0066cc;
  margin-right: 10px;
}
.contact-info span {
  font-weight: bold;
  margin-right: 5px;
}
.contact-info a {
  color: #0066cc;
  text-decoration: none;
}
.contact-info a:hover {
  text-decoration: underline;
}
```

JS:

```
<script>
  function showMenu(menuIndex) {
    document.querySelectorAll('.menu-slider').forEach(slider =>
      slider.classList.remove('active'));
    document.querySelectorAll('.menu-button').forEach(button =>
      button.classList.remove('active'));
    document.getElementById(`menu${menuIndex}`).classList.add('active');
    document.querySelector(`.menu-button:nth-child(${menuIndex})`).classList.add('active');
  }
  showMenu(1);
</script>
<script>
  document.addEventListener("DOMContentLoaded", function() {
    const links = document.querySelectorAll('nav ul li a');
    for (const link of links) {
      link.addEventListener('click', function(e) {
        e.preventDefault();
        const targetId = this.getAttribute('href').substring(1);
        const targetSection = document.getElementById(targetId);
        window.scrollTo({
          top: targetSection.offsetTop,
          behavior: 'smooth'
        });
      });
    }
  });
</script>
```

RESULT:

Thus the program is executed and output is verified.

ASSIGNMENT – 2

REACTJS PROGRAM TO SWITCH BETWEEN LAYOUTS

AIM:

To create a react website that switch between layouts

ALGORITHM:

- Step 1: Create a react file named app.js
- Step 2: Create layout.js
- Step 3: Add functions.
- Step 4: Run the app.js

SOURCE CODE:

App.js

```
import React, { useState } from 'react';
import { ThemeProvider, createTheme, CssBaseline, Container, Stack, TextField,
Button } from '@mui/material';
import Navbar from './Navbar';
import GridLayout from './GridLayout';
import ListLayout from './ListLayout';
import RealTimeForm from './RealTimeForm';
import TextEditor from './TextEditor';
import './App.css';

const App = () => {
  const [layout, setLayout] = useState('grid');
  const [activeSection, setActiveSection] = useState('home');
  const [searchTerm, setSearchTerm] = useState('');
  const handleLayoutChange = () => {
    setLayout((prevLayout) => (prevLayout === 'grid' ? 'list' : 'grid'));
  };
  const handleSectionChange = (section) => {
    setActiveSection(section);
  };
  const handleSearchChange = (event) => {
    setSearchTerm(event.target.value.toLowerCase());
  };
  const theme = createTheme({
    palette: {
      mode: 'light',
    },
  });

  return (
    <ThemeProvider theme={theme}>
```

```
<CssBaseline />
<Navbar onClick={handleSectionChange} />
<Container style={{ padding: '20px' }}>
  {activeSection === 'home' && (
    <Stack spacing={2}>
      <TextField
        label="Search Cars"
        variant="outlined"
        fullWidth
        onChange={handleSearchChange}
        className="search-field" // Apply the class here
      />
      <Button
        variant="contained"
        className="switch-button"
        onClick={handleLayoutChange}
      >
        Switch to {layout === 'grid' ? 'List' : 'Grid'} Layout
      </Button>
      {layout === 'grid' ? <GridLayout searchTerm={searchTerm} /> :
      <ListLayout searchTerm={searchTerm} />}
    </Stack>
  )}

  {activeSection === 'form' && <RealTimeForm />}
  {activeSection === 'video' && (
    <div className="video-container">
      <h2>Video Player</h2>
      <iframe
        width="560"
        height="315"
        src="https://www.youtube.com/embed/dQw4w9WgXcQ"
        title="YouTube video player"
        frameBorder="0"
        allow="accelerometer; autoplay; clipboard-write; encrypted-media;
gyroscope; picture-in-picture"
        allowFullScreen
      ></iframe>
    </div>
  )}
  {activeSection === 'editor' && <TextEditor />}
</Container>
</ThemeProvider>
);
};

export default App;
```

ListLayout.js

```
import React from 'react';
import { List, ListItem, ListItemAvatar, Avatar, ListItemText } from
'@mui/material';
import './ListLayout.css'; // Import the CSS file

const carImages = [
  { name: 'Ferrari', img:
'https://cdn.ferrari.com/cms/network/media/resize/5fab9cebfcbbe6062250a1626-
ferrari-sf90-spider-dynamics-cover-darker-desk?width=1920&height=1080' },
  { name: 'Lamborghini', img: 'https://www.lamborghini.com/sites/it-
en/files/DAM/lamborghini/facelift_2019/model_detail/temerario/temerario/gallery/gal-
lery1.jpg' },
  { name: 'Porsche', img: 'https://images-porsche.imgix.net/-
/media/87672F7AEDD44522BF03A438A52E7E22_8A00202E839145B48307FE95AAF14888_CZ25W010X0
002-911-carrera-rear?w=1759&q=45&dpr=2&auto=format' },
  { name: 'Maserati', img:
'https://maserati.scene7.com/is/image/maserati/maserati/international/Models/grantu-
rismo/granturismo-gt/gt-23/design/GT-New.jpg?$1920x2000$&fit=constrain' },
  { name: 'Bugatti', img: 'https://d2ox13tjqpoxop5.cloudfront.net/BUGATTI-
2024/Models/Tourbillon/07_Gallery/BUGATTI_Tourbillon_1.jpg' },
  { name: 'McLaren', img:
'https://mclaren.scene7.com/is/image/mclaren/MCLAREN_ARTURA_SPIDER_HERO_FRONT_34_HI
GH-1:crop-16x9?wid=1786&hei=1005' },
  { name: 'Rolls-Royce', img: 'https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQyx-ww8_61Ph5Hfvu_TF6VFEoD_ytyVCdMog&s' },
  { name: 'Bentley', img:
'https://www.bentleymotors.com/content/dam/bm/websites/bmcom/bentleymotors-
com/models/25my/25my-gt-
testing/Dynamic%20Stage%20Main%20v2.jpg/_jcr_content/renditions/original.image_file
.1920.823.file/Dynamic%20Stage%20Main%20v2.jpg' },
  { name: 'Aston Martin', img: 'https://www.astonmartin.com/-/media/new-model-
page-design---models/all_models_imagery-for-new-
component/vantage/vantage_main.jpg?mw=1920&rev=7da1cea5c4704f698461f91a986cbd0c&ext
ension=webp&hash=D4698F30B56FE3F57D19C4CCB9EFA1D2' },
  { name: 'Koenigsegg', img:
'https://mainwebstorage.blob.core.windows.net/mediacontainers/styles/1920x1000/azur
e/2024-02/94c29aca8e7e6b5d_org%20%281%29.png?h=a8e100de&itok=1dvWKPUc' }
];

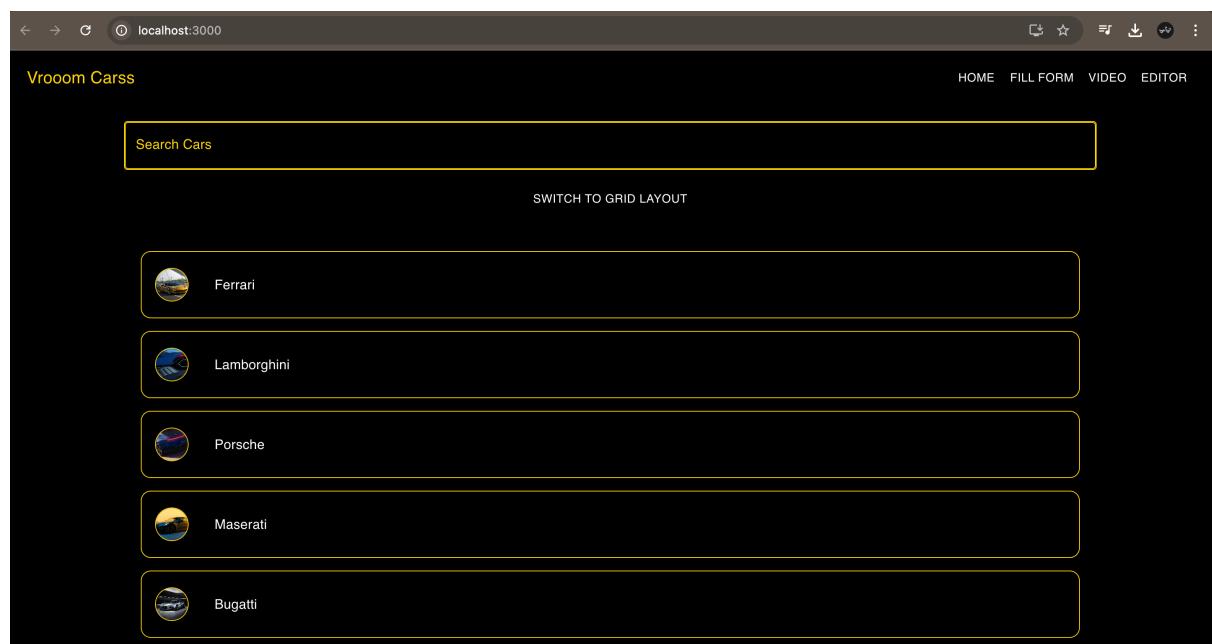
const ListLayout = ({ searchTerm }) => {
  const filteredCars = carImages.filter(car =>
    car.name.toLowerCase().includes(searchTerm.toLowerCase())
  );

  return (
    <div className="list-container">
      <List>
        {filteredCars.map((car, index) => (
          <ListItem key={index} className="list-item">
            <ListItemAvatar>
              <Avatar src={car.img} className="avatar" />
            <ListItemText primary={car.name} />
          </ListItem>
        ))
      </List>
    </div>
  );
}
```

```
</ListItemAvatar>
<div className="list-item-content">
    <ListItemText primary={car.name} className="list-item-text" />
</div>
</ListItem>
)}
</List>
</div>
);
};

export default ListLayout;
```

OUTPUT:



GridLayout.js

```
import React from 'react';
import { Card, CardMedia, CardContent, Typography, Grid } from '@mui/material';
import './GridLayout.css'; // Import the CSS file

const carImages = [
    { name: 'Ferrari', img:
'https://cdn.ferrari.com/cms/network/media/img/resize/5fab9cebfcbef062250a1626-ferrari-sf90-spider-dynamics-cover-darker-desk?width=1920&height=1080' },
    { name: 'Lamborghini', img: 'https://www.lamborghini.com/sites/it-en/files/DAM/lamborghini/facelift_2019/model_detail/temerario/temerario/gallery/gallery1.jpg' },
];
```

```
{ name: 'Porsche', img: 'https://images-porsche.imgix.net/-/media/87672F7AEDD44522BF03A438A52E7E22_8A00202E839145B48307FE95AAF14888_CZ25W010X002-911-carrera-rear?w=1759&q=45&dpr=2&auto=format' },
  { name: 'Maserati', img:
    'https://maserati.scene7.com/is/image/maserati/maserati/international/Models/granturismo/granturismo-gt/gt-23/design/GT-New.jpg?$1920x2000$&fit=constrain' },
    { name: 'Bugatti', img: 'https://d2ox13tjqplop5.cloudfront.net/BUGATTI-2024/Models/Tourbillon/07_Gallery/BUGATTI_Tourbillon_1.jpg' },
      { name: 'McLaren', img:
        'https://mclaren.scene7.com/is/image/mclaren/MCLAREN_ARTURA_SPIDER_HERO_FRONT_34_HI-GH-1:crop-16x9?wid=1786&hei=1005' },
        { name: 'Rolls-Royce', img: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQyx-ww8_61Ph5Hfvu_TF6VFEoD_ytyVCdMog&s' },
          { name: 'Bentley', img:
            'https://www.bentleymotors.com/content/dam/bm/websites/bmcom/bentleymotors-com/models/25my/25my-gt-testing/Dynamic%20Stage%20Main%20v2.jpg/_jcr_content/renditions/original.image_file.1920.823.file/Dynamic%20Stage%20Main%20v2.jpg' },
              { name: 'Aston Martin', img: 'https://www.astonmartin.com/-/media/new-model-page-design---models/all_models_imagery-for-new-component/vantage/vantage_main.jpg?mw=1920&rev=7da1cea5c4704f698461f91a986cbd0c&extension=webp&hash=D4698F30B56FE3F57D19C4CCB9EFA1D2' },
                { name: 'Koenigsegg', img:
                  'https://mainwebstorage.blob.core.windows.net/mediacontainers/styles/1920x1000/azure/2024-02/94c29aca8e7e6b5d_org%20%281%29.png?h=a8e100de&itok=1dwWKPUc' }
                ],
  ];

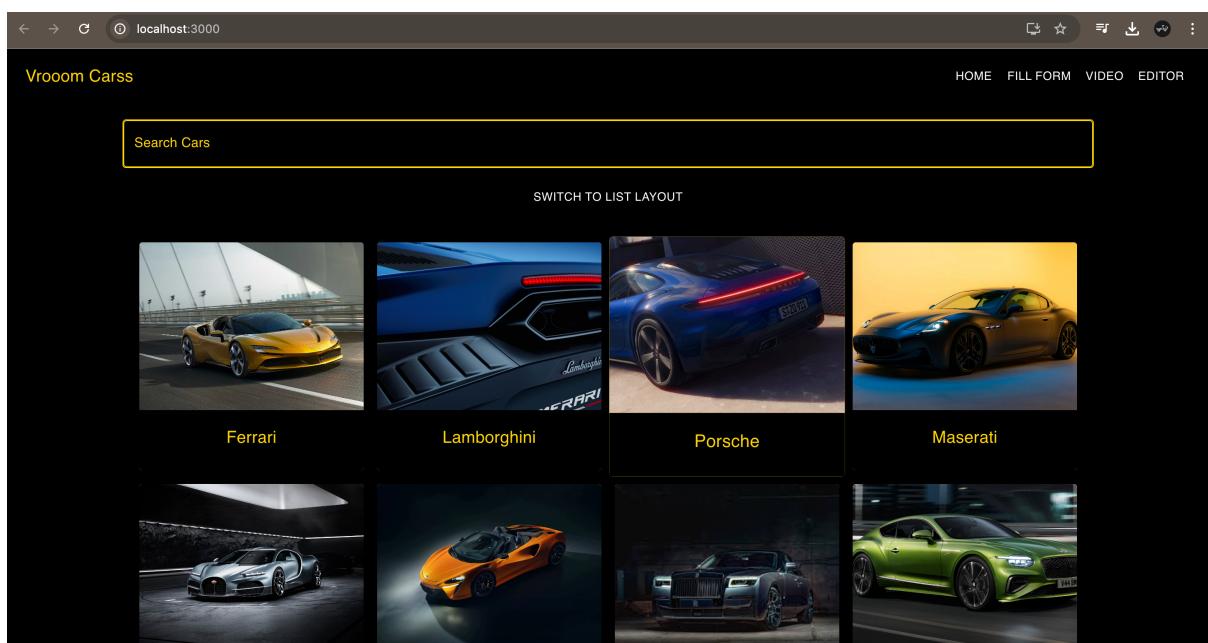
const GridLayout = ({ searchTerm }) => {
  const filteredCars = carImages.filter(car =>
    car.name.toLowerCase().includes(searchTerm.toLowerCase())
  );

  return (
    <div className="grid-container">
      <Grid container spacing={2}>
        {filteredCars.map((car, index) => (
          <Grid item xs={12} sm={6} md={4} lg={3} key={index}>
            <Card className="card">
              <CardMedia
                component="img"
                className="card-media"
                image={car.img}
                alt={car.name}
              />
              <CardContent className="card-content">
                <Typography className="card-text" variant="h6">
                  {car.name}
                </Typography>
              </CardContent>
            </Card>
          </Grid>
        ))}
      </Grid>
    </div>
  );
}
```

```
        })}
    </Grid>
</div>
);
};

export default GridLayout;
```

OUTPUT:



RESULT:

Thus the program has been executed and output is verified.

ASSIGNMENT – 3 **REACTJS PROGRAM TO IMPLEMENT SEARCH FILTER**

AIM:

To create a reactjs website to implement search filter.

ALGORITHM:

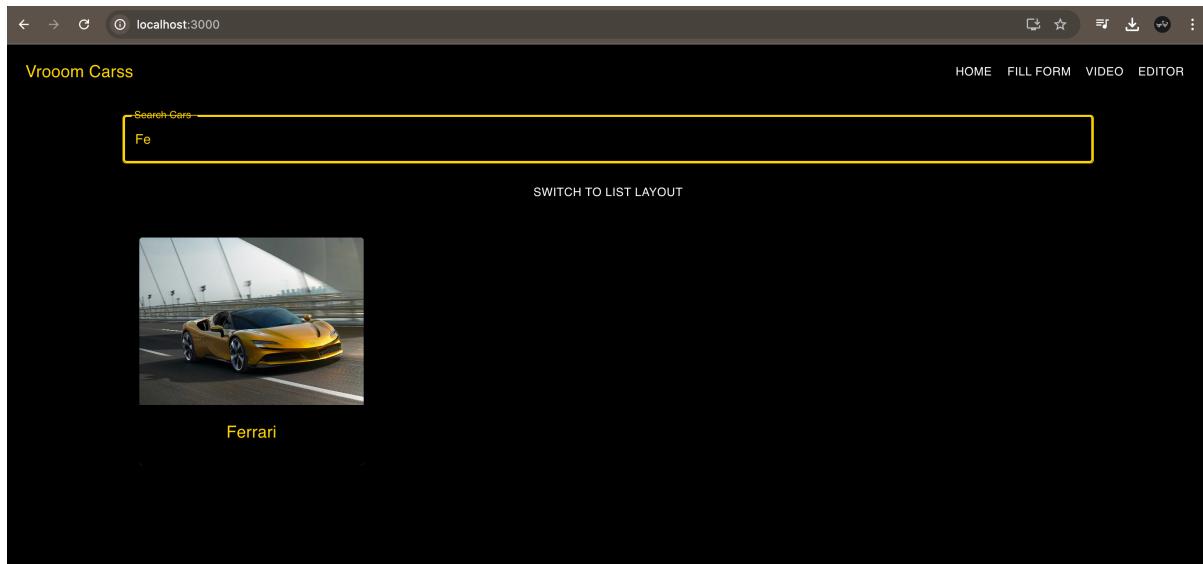
- Step 1: Start the program.
- Step 2: Create the app.js
- Step 3: Create a search filter box with necessary js code.
- Step 4: Run the app.js

SOURCE CODE:

SearchFilter.js

```
return (
  <ThemeProvider theme={theme}>
    <CssBaseline />
    <Navbar onNavClick={handleSectionChange} />
    <Container style={{ padding: '20px' }}>
      {activeSection === 'home' && (
        <Stack spacing={2}>
          <TextField
            label="Search Cars"
            variant="outlined"
            fullWidth
            onChange={handleSearchChange}
            className="search-field" // Apply the class here
          />
          <Button
            variant="contained"
            className="switch-button"
            onClick={handleLayoutChange}
          >
            Switch to {layout === 'grid' ? 'List' : 'Grid'} Layout
          </Button>
          {layout === 'grid' ? <GridLayout searchTerm={searchTerm} /> :
        <ListLayout searchTerm={searchTerm} />}
        </Stack>
      )}
```

OUTPUT:



RESULT:

Thus program is executed and output is verified.

ASSIGNMENT - 4

REACTJS PROGRAM TO CREATE A FORM

AIM:

To create a reactjs website with a form

ALGORITHM:

- Step 1: Create a app.js
- Step 2: Create form.js
- Step 3: Run app.js

SOURCE CODE:

RealTimeForm.js

```
import React, { useState } from 'react';
import './RealTimeForm.css';

const RealTimeForm = () => {
  const [formData, setFormData] = useState({
    name: '',
    email: '',
    message: ''
  });

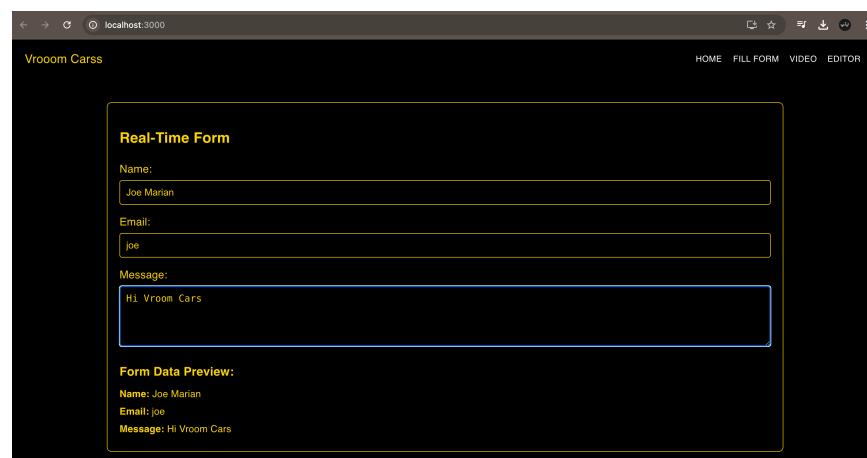
  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData({
      ...formData,
      [name]: value
    });
  };

  return (
    <div className="form-container" id="form">
      <h2>Real-Time Form</h2>
      <form>
        <div className="form-group">
          <label htmlFor="name">Name:</label>
          <input
            type="text"
            id="name"
            name="name"
            value={formData.name}
            onChange={handleChange}
            className="form-input"
          />
        </div>
        <div className="form-group">
```

```
<label htmlFor="email">Email:</label>
<input
  type="email"
  id="email"
  name="email"
  value={formData.email}
  onChange={handleChange}
  className="form-input"
/>
</div>
<div className="form-group">
  <label htmlFor="message">Message:</label>
  <textarea
    id="message"
    name="message"
    value={formData.message}
    onChange={handleChange}
    className="form-textarea"
/>
</div>
</form>
<div className="preview">
  <h3>Form Data Preview:</h3>
  <p><strong>Name:</strong> {formData.name}</p>
  <p><strong>Email:</strong> {formData.email}</p>
  <p><strong>Message:</strong> {formData.message}</p>
</div>
</div>
);
};

export default RealTimeForm;
```

OUTPUT:



RESULT:

Thus the program is executed and output is verified.

ASSIGNMENT – 5

IMPLEMENTATION OF NAVIGATION MENU

AIM:

To create an ReactJS program for navigation menu.

ALGORITHM:

- Step 1: Create a app.js
- Step 2: Create a navbar code.
- Step 3: Run the app.js

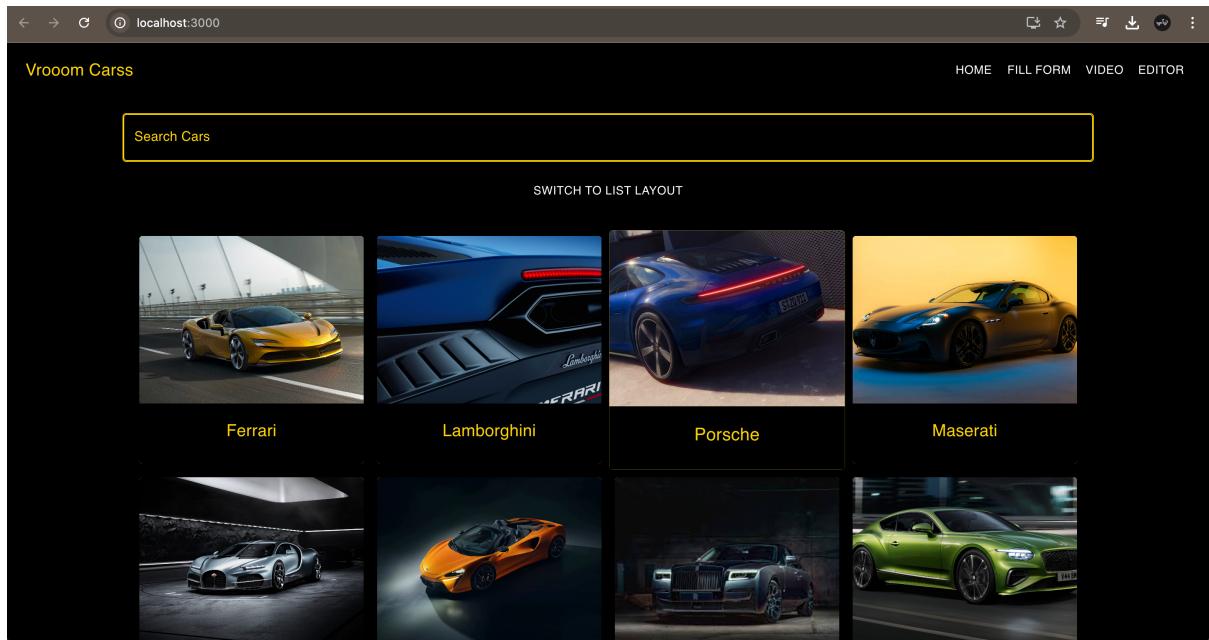
SOURCE CODE:

```
import React from 'react';
import { AppBar, Toolbar, Button, Typography } from '@mui/material';
import './Navbar.css'; // Import the CSS file

const Navbar = ({ onNavClick }) => {
  return (
    <AppBar position="static" className="navbar">
      <Toolbar>
        <Typography variant="h6" className="navbar-title">
          Vrooom Carss
        </Typography>
        <Button className="navbar-button" color="inherit" onClick={() => onNavClick('home')}>Home</Button>
        <Button className="navbar-button" color="inherit" onClick={() => onNavClick('form')}>Fill Form</Button>
        <Button className="navbar-button" color="inherit" onClick={() => onNavClick('video')}>Video</Button>
        <Button className="navbar-button" color="inherit" onClick={() => onNavClick('editor')}>Editor</Button>
      </Toolbar>
    </AppBar>
  );
};

export default Navbar;
```

OUTPUT:



RESULT:

Thus the program is executed and output is verified.

ASSIGNMENT – 6 **CREATION OF A VIDEO PLAYER**

AIM:

To create a video player using ReactJS.

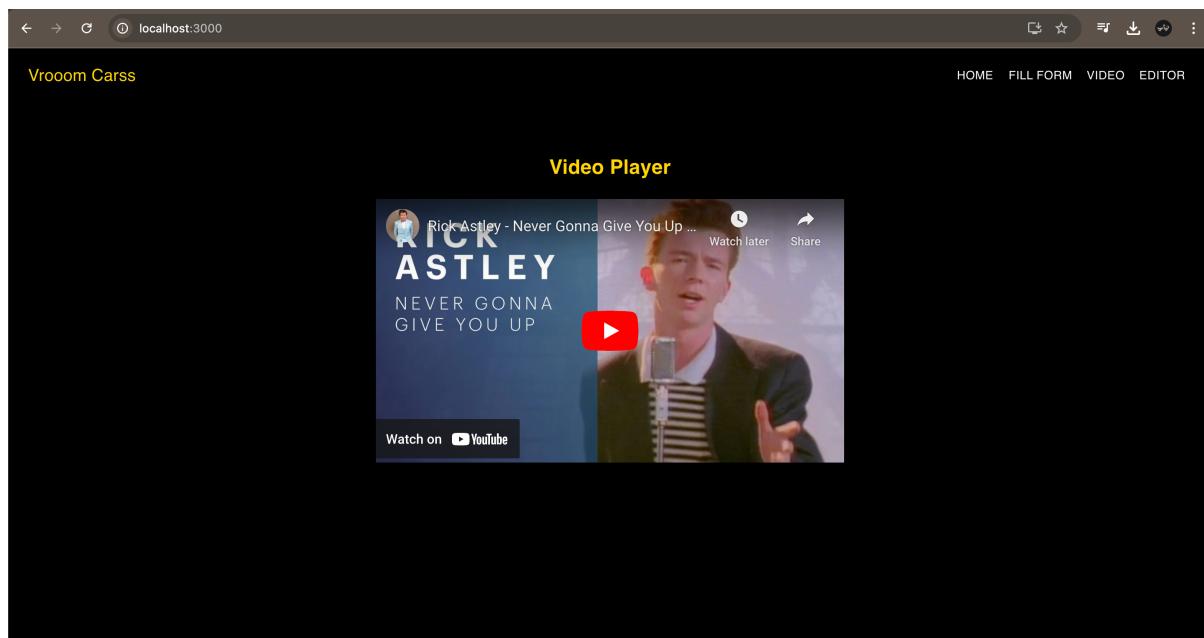
ALGORITHM:

- Step 1: Start the app.js
- Step 2: Include the iframe
- Step 3: Run the app.js

SOURCE CODE:

```
<div className="video-container">
    <h2>Video Player</h2>
    <iframe
        width="560"
        height="315"
        src="https://www.youtube.com/embed/dQw4w9WgXcQ"
        title="YouTube video player"
        frameBorder="0"
        allow="accelerometer; autoplay; clipboard-write; encrypted-media;
gyroscope; picture-in-picture"
        allowFullScreen
    ></iframe>
</div>
```

OUTPUT:



RESULT:

Thus the program is executed and output is verified.

ASSIGNMENT – 7 **CREATION OF TEXT EDITOR**

AIM:

To create an editor using ReactJS.

ALGORITHM:

- Step 1: Start the app.js
- Step 2: Create the texyeditor.js file.
- Step 3: Run the app.js

SOURCE CODE:

TextEditor.js

```
import React, { useState } from 'react';
import ReactQuill from 'react-quill';
import 'react-quill/dist/quill.snow.css'; // Import Quill's CSS
import './TextEditor.css';

const TextEditor = () => {
  const [editorHtml, setEditorHtml] = useState('');

  const handleChange = (value) => {
    setEditorHtml(value);
  };

  return (
    <div className="text-editor-container" id="editor">
      <h2>Text Editor</h2>
      <ReactQuill
        value={editorHtml}
        onChange={handleChange}
        modules={TextEditor.modules}
        formats={TextEditor.formats}
        className="text-editor"
      />
      <div className="editor-preview">
        <h3>Editor Preview:</h3>
        <div dangerouslySetInnerHTML={{ __html: editorHtml }} />
      </div>
    </div>
  );
};

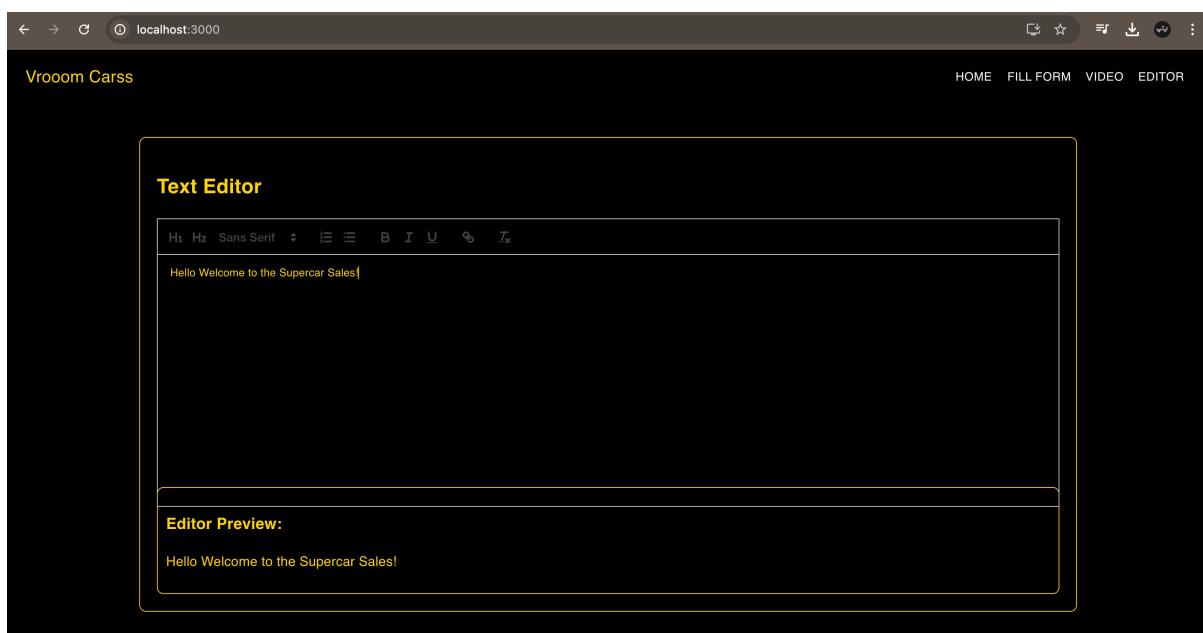
TextEditor.modules = {
  toolbar: [
    [{ 'header': '1' }, { 'header': '2' }, { 'font': [] }],
    [{ 'list': 'ordered' }, { 'list': 'bullet' }],
    ['bold', 'italic', 'underline'],
  ],
};
```

```
['link'],
['clean']
],
};

TextEditor.formats = [
  'header', 'font', 'list', 'bullet', 'bold', 'italic', 'underline', 'link'
];

export default TextEditor;
```

OUTPUT:



RESULT:

Thus the program is executed and output is verified.

ASSIGNMENT – 8
CREATION OF BASIC HTML

AIM:

To create a basic html website.

ALGORITHM:

1. Create a new HTML file named index.html.
2. Add the <!DOCTYPE html> declaration at the top of the file.
3. Open the <html> tag with lang="en".
4. Inside <html>, add the <head> section with character encoding, viewport settings, title, and CSS link.
5. Open the <body> tag.
6. Inside <body>, add the <header> section with an <h1> heading.
7. Add <section> elements with headings, paragraphs, images, and links.
8. Close the <body> and <html> tags

SOURCE CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>My Basic Web Page</title>
<link rel="stylesheet" href="css/styles.css">
</head>
<body>
<header>
<h1>Welcome to My Web Page</h1>
</header>
<section>
<h2>About This Page</h2>
<p>This is a simple HTML page to demonstrate basic elements such as headings, paragraphs, and images created by Saran.</p>
</section>
<section>
<h2>Image Example</h2>

</section>
<section>
<h2>External Link</h2>
<p>Check out <a href="https://www.youtube.com" target="_blank">Youtube.com</a> for more information.</p>
</section>
Joe Marian A
22011102034
B.Tech CSE(IoT) A
</body>
</html>
```

OUTPUT:



Welcome to My Web Page

About This Page

This is a simple HTML page to demonstrate basic elements such as headings, paragraphs, and images created by Saran.

Image Example

Logo Image

External Link

Check out [Youtube.com](#) for more information.

Joe Marian A 22011102034 B.Tech CSE(IoT) A

RESULT:

Thus the program is executed and output is verified.

ASSIGNMENT – 9
FORM VALIDATION

AIM:

To create a website for form validation.

ALGORITHM:

- Create a new HTML file named index.html.
- Add the <!DOCTYPE html> declaration at the top of the file.
- Open the <html> tag with lang="en".
- Inside <html>, add the <head> section with character encoding, viewport settings, and the page title.
- Open the <body> tag.
- Inside <body>, add an <h1> heading for the form title.
- Create a <form> element with action and method attributes.
- Add <label> and <input> elements for name, email, and password fields, ensuring to
- Include validation attributes, and close the <form> and <body> tags.

SOURCE CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Form Validation</title>
</head>
<body>
<h1>Registration Form</h1>
<form action="/submit_form" method="post">
<label for="name">Name:</label>
<input type="text" id="name" name="name" required>
<br><br>
<label for="email">Email:</label>
<input type="email" id="email" name="email" required>
<br><br>
<label for="password">Password:</label>
<input type="password" id="password" name="password" minlength="6" required>
<br><br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

OUTPUT:

The screenshot shows a web browser window titled "Form Validation". The address bar indicates the file is located at "/home/snucse/sample.html". The main content of the page is a "Registration Form". It contains three input fields: "Name" with value "Joe", "Email" with value "joe@gmail.com", and "Password" with value "****". Below these fields is a "Submit" button.

RESULT:

Thus the program is executed and output is verified.

ASSIGNMENT - 10
FETCHING DATA FROM SERVER USING AJAX

AIM:

To create a website for fetching data from server.

ALGORITHM:

- Create an HTML File: Create a file named index.html.
- Add Basic HTML Structure: Include the <!DOCTYPE html>, <html>, <head>, and <body> tags. Set up the meta tags for character encoding and viewport settings.
- Add the Registration Form: Inside the <body> tag, add a heading <h1>, and create a <form> element with action and method attributes. Include form fields for name, email, and password, ensuring that each field has appropriate validation attributes (e.g., required, type="email", minlength).
- Add a Submit Button: Inside the form, add an <input type="submit"> button to allow users to submit the form.
- Create a JavaScript File: Create a file named script.js.
- Define the AJAX Function: In script.js, define a function loadDoc() that creates an XMLHttpRequest object.
- Configure the AJAX Request: Set up the onreadystatechange event handler to check if the request is complete and successful. Update the HTML content with the response text from the server.
- Add the AJAX Request Code: Call the open() method to configure the request and send() to initiate it. Ensure the request is pointing to a valid path, such as data.txt.

SOURCE CODE:

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Form Validation</title>
</head>
<body>
<h1>Registration Form</h1>
<form action="/submit_form" method="post">
<label for="name">Name:</label>
<input type="text" id="name" name="name" required>
<br><br>
<label for="email">Email:</label>
```

```
<input type="email" id="email" name="email" required>
<br><br>
<label for="password">Password:</label>
<input type="password" id="password" name="password" minlength="6" required>
<br><br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

JS:

```
function loadDoc() {
var xhttp = new XMLHttpRequest(); // Create XMLHttpRequest object
xhttp.onreadystatechange = function() { // Define event handler
if (this.readyState == 4 && this.status == 200) { // Check request completion
document.getElementById("data").innerHTML = this.responseText; // Update HTML
content
}
};
// Use the file protocol with the correct path
xhttp.open("GET", "data.txt", true); // Configure the request
xhttp.send(); // Send the request
}
```

OUTPUT:

AJAX Data Fetching Example

AJAX Data Fetching Example

This is the data fetched from the server.

RESULT:

Thus the code is executed and output is verified.

ASSIGNMENT – 11 **IMPLEMENTATION OF A SEMESTER FEE FORM**

AIM:

To create a website for semester fee form.

ALGORITHM:

- Create the HTML File: Set up the HTML file (index.html) with the necessary structure and metadata.
- Add Form Elements: Inside the <body>, create a <form> with action and method attributes. Include form fields for user inputs (e.g., name, fee amount, payment date) with appropriate HTML5 attributes.
- Define Validation Function: In the JavaScript, define a function (e.g., validateForm()) that will be called when the form is submitted.
- Get Form Field Values: In the validation function, retrieve the values from the form fields using document.getElementById().
- Perform Validations: Implement logic to check for valid input in the form fields.
For example, check if the fee amount is a number and if the payment date is provided.
- Handle Validation Errors: Collect any validation errors into a string variable. If there are errors, display them using alert() and prevent form submission by returning false.
- Allow Submission if Valid: If there are no errors, allow the form to be submitted by returning true.

SOURCE CODE:

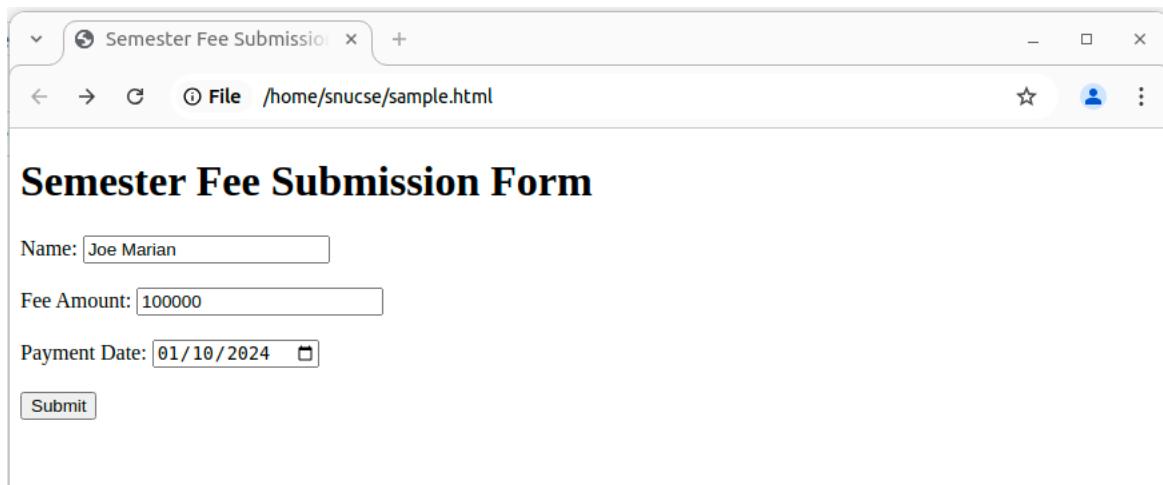
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Semester Fee Submission</title>
<script>
function validateForm() {
var amount = document.getElementById("amount").value;
var date = document.getElementById("date").value;
var errorMsg = "";
if(isNaN(amount) || amount <= 0) {
errorMsg += "Please enter a valid fee amount.\n";
}
if(date === "") {
errorMsg += "Please select a date.\n";
}
}
```

```
}

if(errorMsg !== "") {
  alert(errorMsg);
  return false; // Prevents form submission
}
return true; // Allows form submission
}

</script>
</head>
<body>
<h1>Semester Fee Submission Form</h1>
<form action="/submit_fee" method="post" onsubmit="return validateForm()">
<label for="name">Name:</label>
<input type="text" id="name" name="name" required>
<br><br>
<label for="amount">Fee Amount:</label>
<input type="text" id="amount" name="amount" required>
<br><br>
<label for="date">Payment Date:</label>
<input type="date" id="date" name="date" required>
<br><br>
<input type="submit" value="Submit">
</form>
</body>
```

OUTPUT:



This page says
Please enter a valid fee amount.

OK

RESULT:

Thus the program is executed and output is verified.