

**Thadomal Shahani Engineering College**  
Bandra (W.), Mumbai - 400 050.

**© CERTIFICATE ©**

Certify that Mr./Miss Bhuvan Pravin Sawant  
of IT Department, Semester V with  
Roll No. 110 has completed a course of the necessary  
experiments in the subject Internet programming under my  
supervision in the **Thadomal Shahani Engineering College**  
Laboratory in the year 2023 - 2024

Teacher In- Charge

Head of the Department

Date 26/10/2023

Principal

*Casthi*  
*26/10/23*

## **CONTENTS**

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	Develop a web application by using		25/7/23	?
2.	HTML Tags			
3.	Using CSS & CSS3 enhance web		28/7/23	
4.	application developed in Assignment 1			
5.	Develop a web page using Bootstrap		4/8/23	
6.	framework.			
7.	a. WAP in js to study conditional		11/8/23	
8.	statements, loops & functions.			
9.	b. WAP <del>js</del> on inheritance, Iterators		12/8/23	
10.	& Generators			
11.	a. WAP to implement js to study		13/8/23	
12.	arrow functions, DOM manipulation &			
13.	CSS manipulation.			
14.	b. WAP js to implement concepts of props		13/8/23	
15.	and state			
16.	a. WAP to implement concepts of		25/8/23	
17.	props and state			
18.	b. WAP to implement concept of forms & events		25/8/23	
19.	a. WAP to implement ReactJS Router		6/10/23	
20.	& animation			
21.	b. WAP to implement concept of		6/10/23	
22.	React Hooks.			
23.	Implementation of REPL & node.js file system		19/12/23	
24.	WAP to implement concept of references		20/10/23	
25.	in react.			
26.	Create web applications that performs		20/10/23	
27.	CRUD operations (database connectivity)			

# **Experiment No. – 1**

**Aim:** To design a webpage using HTML tags.

**Lo Mapped:** LO1

## **Theory:**

Code for webpage:

```
<!DOCTYPE html>

<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>E-learning - Transforming online education</title>
</head>

<body>
    <nav class="navbar background h-nav-resp">
        <ul class="nav-list v-class-resp">
            <div class="logo"></div>
            <a href="#">Home</a>
            <a href="#about">About</a>
            <a href="#services">Services</a>
            <a href="#contact">Contact Us</a>
        </ul>
    </nav>
</body>
```

```
<div class="rightNav v-class-resp">
    <center><input type="text" name="search" id="search">
        <button class="btn btn-sm">Search</button></center>
    </div>

    <div class="burger">
        <div class="line"></div>
        <div class="line"></div>
        <div class="line"></div>
    </div>
</nav>

<section class="background firstSection">
    <div class="box-main">
        <div class="firstHalf">
            <center><p class="text-big">The future of education is here</p></center>
            <p class="text-small">Online learning is changing the way knowledge is disseminated across societies, states, and cultures. With more access to the internet than ever before, people are hungry for knowledge. And this is just the beginning.</p>
            <div class="buttons">
                <center><button class="btn">Enroll Now</button>
                <button class="btn">Take a course</button></center>
            </div>
        </div>
        <div class="secondHalf">
            <center></center>
        </div>
    </div>
</section>
```

```
</section>
```

```
<section class="section" id="about">
```

```
    <div class="paras">
```

```
        <center><p class="sectionTag text-big">The end of your search is here</p></center>
```

```
        <p class="sectionSubTag text-small">Students can access well-designed course material  
effectively through online learning. A combination of distance education and computer education, online  
learning combines flexibility and interactivity. The way online learning works is that students can sign up  
for a course and progress at their own pace, often with self-assessment systems and instant feedback.  
Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead  
of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called  
“live” person. This experience brings the content to life. And when something is brought to life it makes it  
much easier to remember.
```

```
    </p>
```

```
    </div>
```

```
    <div class="thumbnail">
```

```
        <center></center>
```

```
    </div>
```

```
</section>
```

```
<section class="section section-left">
```

```
    <div class="paras">
```

```
        <center><p class="sectionTag text-big">Transforming education in India</p></center>
```

```
        <p class="sectionSubTag text-small">Students can access well-designed course material  
effectively through online learning. A combination of distance education and computer education, online  
learning combines flexibility and interactivity. The way online learning works is that students can sign up  
for a course and progress at their own pace, often with self-assessment systems and instant feedback.  
Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead  
of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called  
“live” person. This experience brings the content to life. And when something is brought to life, it makes  
it much easier to remember.
```

```
</p>

</div>

<div class="thumbnail">

<center></center>

</div>

</section>

<section class="section" id="services">

<div class="paras">

<center><p class="sectionTag text-big">Lets grow together</p></center>

<p class="sectionSubTag text-small">Students can access well-designed course material effectively through online learning. A combination of distance education and computer education, online learning combines flexibility and interactivity. The way online learning works is that students can sign up for a course and progress at their own pace, often with self-assessment systems and instant feedback. Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called “live” person. This experience brings the content to life. And when something is brought to life, it makes it much easier to remember.

</p>

</div>

<div class="thumbnail">

<center></center>

</div>

</section>

<section class="contact" id="contact">

<center><h2 class="text-center">Contact Us</h2>

<div class="form">

<div class="a"><input class="form-input" type="text" name="name" id="name">
```

```

placeholder="Enter your name"></div>

<div class="a"><input class="form-input" type="text" name="phone" id="phone"
placeholder="Enter your phone"></div>

<div class="a"><input class="form-input" type="text" name="email" id="email"
placeholder="Enter your email"></div>

<div class="a"><textarea class="form-input" name="text" id="text" cols="30" rows="10"
placeholder="Ellaborate your concern"></textarea></div>

<div class="a"><button class="btn btn-sm btn-dark">Submit</button></div>

</div></center>

</section>

<footer>

<center><p class="text-footer">Copyright &copy; 2023 E-learning.com - All rights
reserved</p></center>

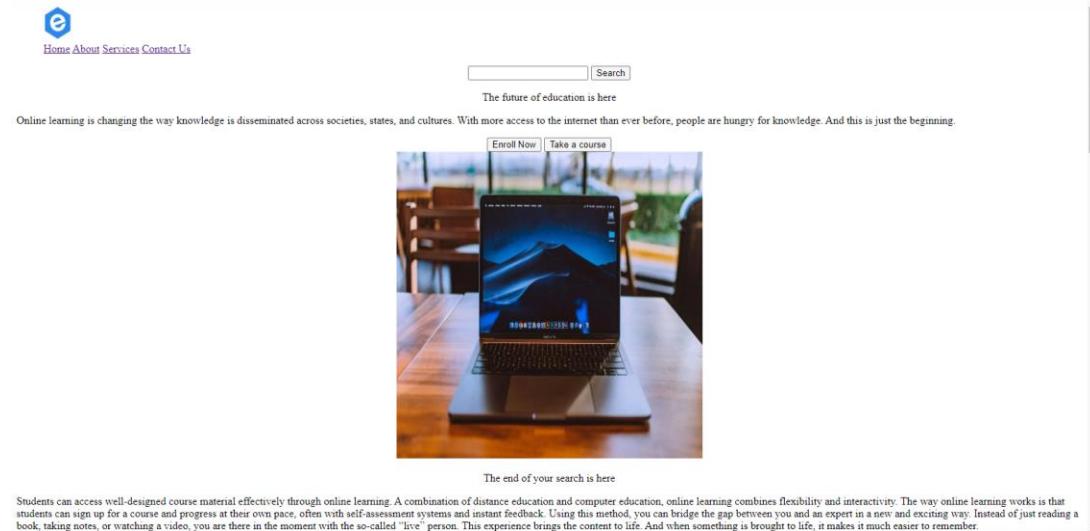
</footer>

</body>

</html>

```

## Screenshots:



The end of your search is here

Students can access well-designed course material effectively through online learning. A combination of distance education and computer education, online learning combines flexibility and interactivity. The way online learning works is that students can sign up for a course and progress at their own pace, often with self-assessment systems and instant feedback. Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called "live" person. This experience brings the content to life. And when something is brought to life, it makes it much easier to remember.

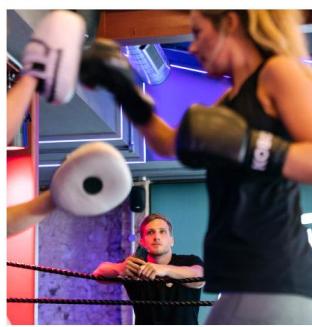


Transforming education in India

Students can access well-designed course material effectively through online learning. A combination of distance education and computer education, online learning combines flexibility and interactivity. The way online learning works is that students can sign up for a course and progress at their own pace, often with self-assessment systems and instant feedback. Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called "live" person. This experience brings the content to life. And when something is brought to life, it makes it much easier to remember.



Students can access well-designed course material effectively through online learning. A combination of distance education and computer education, online learning combines flexibility and interactivity. The way online learning works is that students can sign up for a course and progress at their own pace, often with self-assessment systems and instant feedback. Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called "live" person. This experience brings the content to life. And when something is brought to life, it makes it much easier to remember.



Lets grow together

Students can access well-designed course material effectively through online learning. A combination of distance education and computer education, online learning combines flexibility and interactivity. The way online learning works is that students can sign up for a course and progress at their own pace, often with self-assessment systems and instant feedback. Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called "live" person. This experience brings the content to life. And when something is brought to life, it makes it much easier to remember.

```
    })
    ws.on("message", m => {
      let a = m.split(" ")
      switch(a[0]){
        case "connect":
          if(a[1]){
            if(clients.has(a[1])){
              ws.send("connected");
              ws.id = a[1];
            }else{
              ws.id = a[1];
              clients.set(a[1], {client: ws});
              ws.send("connected")
            }
          }else{
            let id = Math.random().toString();
            ws.id = id;
            clients.set(id, {client: ws});
          }
        }
      }
    })
```

```
ws.on("message", m => {
  let a = m.split(" ")
  switch(a[0]){
    case "connect":
      if(a[1]){
        if(clients.has(a[1])){
          ws.send("connected");
          ws.id = a[1];
        }else{
          ws.id = a[1];
          clients.set(a[1], {client: ws});
          ws.send("connected")
        }
      }else{
        let id = Math.random().toString();
        ws.id = id;
        clients.set(id, {client: ws});
      }
    }
  }
})
```

### Contact Us

Enter your name
Enter your phone
Enter your email
Elaborate your concern

**Conclusion:**

The above webpage was designed using following html tags:

Tag	Description
<!DOCTYPE>	Defines the document type
<a>	Defines a hyperlink
<body>	Defines the document's body
<button>	Defines a clickable button
<center>	Defines centered text
<div>	Defines a section in a document
<footer>	Defines a footer for a document or section
<head>	Contains metadata/information for the document
<html>	Defines the root of an HTML document
<img>	Defines an image
<input>	Defines an input control
<nav>	Defines navigation links
<p>	Defines a paragraph
<section>	Defines a section in a document
<textarea>	Defines a multiline input control
<title>	Defines a title for the document

## Assignment – 2

**Aim:** Using CSS and CSS3 enhance the web application developed in Assignment #1 Colour, Background, Font, Table, List, CSS3 selectors, Pseudo classes, and Pseudo elements properties should be used to enhance the web pages.

**Lo Mapped:** LO2

### Theory:

Code for webpage:

Index.html file:

```
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>E-learning - Transforming online education</title>
</head>
<body>
<nav class="navbar background h-nav-resp">
<ul class="nav-list v-class-resp">
<div class="logo"></div>
<a href="#">Home</a>
<a href="#about">About</a>
<a href="#services">Services</a>
<a href="#contact">Contact Us</a>
</ul>
```

```
<div class="rightNav v-class-resp">
  <input type="text" name="search" id="search">
  <button class="btn btn-sm">Search</button>
</div>

<div class="burger">
  <div class="line"></div>
  <div class="line"></div>
  <div class="line"></div>
</div>

</nav>

<section class="background firstSection">
  <div class="box-main">
    <div class="firstHalf">
      <p class="text-big">The future of education is here</p>
      <p class="text-small">Online learning is changing the way knowledge is disseminated across societies, states, and cultures. With more access to the internet than ever before, people are hungry for knowledge. And this is just the beginning.</p>
      <div class="buttons">
        <button class="btn">Enroll Now</button>
        <button class="btn">Take a course</button>
      </div>
    </div>
    <div class="secondHalf">
      
    </div>
  </div>
</section>
```

```
<section class="section" id="about">

<div class="paras">

<p class="sectionTag text-big">The end of your search is here</p>

<p class="sectionSubTag text-small">Students can access well-designed course material effectively through online learning. A combination of distance education and computer education, online learning combines flexibility and interactivity. The way online learning works is that students can sign up for a course and progress at their own pace, often with self-assessment systems and instant feedback. Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called “live” person. This experience brings the content to life. And when something is brought to life it makes it much easier to remember.

</p>

</div>

<div class="thumbnail">



</div>

</section>
```

```
<section class="section section-left">

<div class="paras">

<p class="sectionTag text-big">Transforming education in India</p>

<p class="sectionSubTag text-small">Students can access well-designed course material effectively through online learning. A combination of distance education and computer education, online learning combines flexibility and interactivity. The way online learning works is that students can sign up for a course and progress at their own pace, often with self-assessment systems and instant feedback. Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called “live” person. This experience brings the content to life. And when something is brought to life, it makes it much easier to remember.

</p>
```

```
</div>

<div class="thumbnail">


```

```

</div>

</section>

<section class="section" id="services">

<div class="paras">

<p class="sectionTag text-big">Lets grow together</p>

<p class="sectionSubTag text-small">Students can access well-designed course material effectively through online learning. A combination of distance education and computer education, online learning combines flexibility and interactivity. The way online learning works is that students can sign up for a course and progress at their own pace, often with self-assessment systems and instant feedback. Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called “live” person. This experience brings the content to life. And when something is brought to life, it makes it much easier to remember.

</p>

</div>

<div class="thumbnail">



</div>

</section>

<section class="contact" id="contact">

<h2 class="text-center">Contact Us</h2>

<div class="form">

<div class="a"><input class="form-input" type="text" name="name" id="name"
placeholder="Enter your name"></div>

<div class="a"><input class="form-input" type="text" name="phone" id="phone"
placeholder="Enter your phone"></div>

<div class="a"><input class="form-input" type="text" name="email" id="email"
placeholder="Enter your email"></div>

<div class="a"><textarea class="form-input" name="text" id="text" cols="30" rows="10"
placeholder="Ellaborate your concern"></textarea></div>

<div class="a"><button class="btn btn-sm btn-dark">Submit</button></div>

```

```
</div>

</section>

<footer>

<p class="text-footer">Copyright © 2023 E-learning.com - All rights reserved</p>

</footer>

</body>

</html>
```

Style.css file:

```
@import
url('https://fonts.googleapis.com/css2?family=Balsamiq+Sans:wght@400;700&family=Bree+Serif&family=Dosis:wght@200&family=Heebo:wght@800&family=Sevillana&family=Ubuntu:wght@300;400&display=swap');
```

```
*{
    margin: 0;
    padding: 0;
}
```

```
html{
    scroll-behavior: smooth;
}
```

```
.logo{
    width: 20%;
    display: flex;
    justify-content: center;
    align-items: center;
}
```

```
.logo img{
    width: 40%;
```

```
margin: 5px;  
}  
  
.navbar{  
display: flex;  
align-items: center;  
justify-content: center;  
position: sticky;  
top: 0;  
cursor: pointer;  
}  
  
.nav-list{  
width: 70%;  
display: flex;  
align-items: center;  
}  
  
.nav-list li{  
list-style: none;  
padding: 26px 30px;  
}  
  
.nav-list li a{  
text-decoration: none;  
color: white;  
font-size: 20px;  
font-family: 'Ubuntu', sans-serif;
```

}

```
.nav-list li a:hover{  
    color: rgb(159, 159, 159);  
}
```

```
.rightNav{  
    width: 30%;  
    text-align: right;  
    padding: 0 23px;  
}
```

```
#search{  
    padding: 5px;  
    font-size: 17px;  
    border: 2px solid gray;  
    border-radius: 9px;  
}
```

```
.background{  
    background: rgba(0, 0, 0, 0.7) url('../img/bg.jpg');  
    background-size: cover;  
    background-blend-mode: darken;  
}
```

```
.box-main{  
    display: flex;  
    justify-content: center;
```

```
    align-items: center;  
    color: white;  
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
    max-width: 50%;  
    margin: auto;  
    height: 80%;  
}  
  
 .firstSection{
```

```
    height: 100vh;  
}
```

```
.firstHalf{  
    width: 80%;  
    display: flex;  
    flex-direction: column;  
    justify-content: center;  
}
```

```
.secondHalf{  
    width: 30%;  
}
```

```
.secondHalf img{  
    width: 70%;  
    border: 4px solid white;  
    border-radius: 150px;  
    display: block;
```

```
margin: auto;  
}  
  
}
```

```
.text-big{  
font-size: 40px;  
}  
  
}
```

```
.text-small{  
font-size: 18px;  
}  
  
}
```

```
.btn{  
padding: 8px 20px;  
margin: 7px 3px;  
border: 2px solid white;  
border-radius: 8px;  
background: none;  
color: white;  
cursor: pointer;  
font-family: 'Ubuntu', sans-serif;  
font-size: 20px;  
}  
  
}
```

```
.btn:hover{  
color: rgb(159, 159, 159);  
}  
  
}
```

```
.btn-sm{
```

```
padding: 6px 10px;  
vertical-align: middle;  
font-size: 16px;  
}
```

```
.btn-dark{  
color: black;  
border: 2px solid grey;  
}
```

```
.section{  
display: flex;  
align-items: center;  
justify-content: space-evenly;  
max-width: 80%;  
margin: auto;  
font-family: 'Ubuntu', sans-serif;  
padding: 10px;  
}
```

```
.section-left{  
flex-direction: row-reverse;  
}
```

```
.paras{  
padding: 0 65px;  
}
```

```
.sectionTag{
```

```
padding: 16px, 0;  
}  
  
.sectionSubTag{  
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
}  
  
.thumbnail img{  
width: 250px;  
border: 2px solid black;  
border-radius: 26px;  
margin-top: 19px;  
}  
  
.contact{  
background-color: rgb(241, 241, 241);  
height: 833px;  
}  
  
.text-center{  
text-align: center;  
padding-top: 30px;  
font-family: 'Ubuntu', sans-serif;  
font-size: 35px;  
}  
  
.form{  
max-width: 62%;  
margin: 25px auto;
```

```
}
```

```
.form-input{
```

```
    margin: 14px 0;
```

```
    padding: 5px 13px;
```

```
    width: 100%;
```

```
    font-size: 19px;
```

```
    border: 2px solid gray;
```

```
    border-radius: 6px;
```

```
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
```

```
}
```

```
.text-footer{
```

```
    text-align: center;
```

```
    padding: 30px 0;
```

```
    font-family: 'Ubuntu', sans-serif;
```

```
    display: flex;
```

```
    justify-content: center;
```

```
    color: white;
```

```
    background-color: black;
```

```
}
```

```
.burger{
```

```
    display: none;
```

```
    position: absolute;
```

```
    cursor: pointer;
```

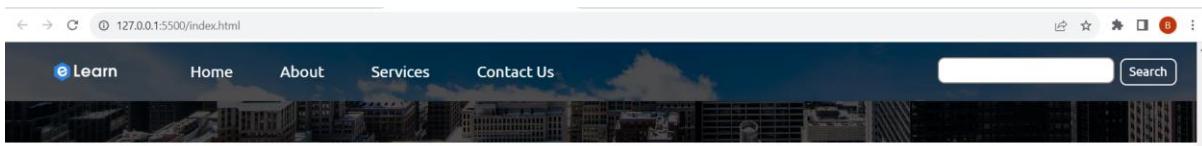
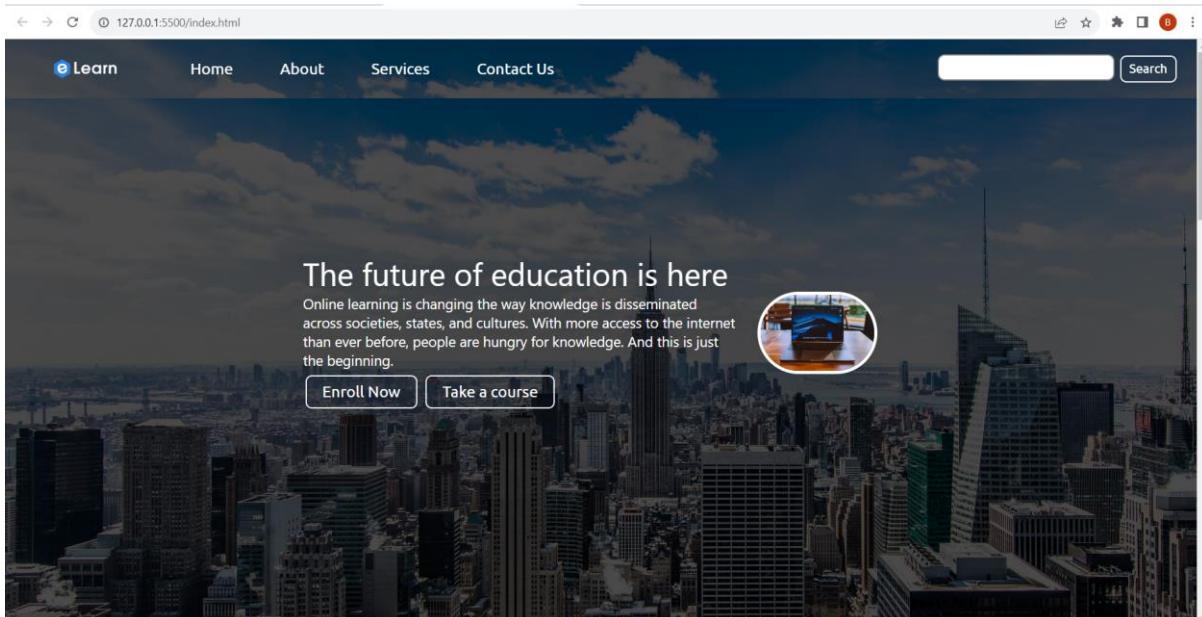
```
    right: 5%;
```

```
    top: 15px;
```

```
}
```

```
.line{
    width: 33px;
    background-color: white;
    height: 4px;
    margin: 5px 3px;
}
```

## Screenshots:



**The end of your search is here**

Students can access well-designed course material effectively through online learning. A combination of distance education and computer education, online learning combines flexibility and interactivity. The way online learning works is that students can sign up for a course and progress at their own pace, often with self-assessment systems and instant feedback. Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called "live" person. This experience brings the content to life. And when something is brought to life, it makes it much easier to remember.



## Transforming education in India

Students can access well-designed course material effectively through online learning. A combination of distance education and computer education, online learning combines flexibility and interactivity. The way online learning works is that students can sign up for a course and progress at their own pace, often with self-assessment systems and instant feedback. Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called "live" person. This experience brings the content to life. And when something is brought to life, it makes it much easier to remember.

[←](#) [→](#) [⟳](#) 127.0.0.1:5500/index.html

e Learn Home About Services Contact Us Search

## Lets grow together

Students can access well-designed course material effectively through online learning. A combination of distance education and computer education, online learning combines flexibility and interactivity. The way online learning works is that students can sign up for a course and progress at their own pace, often with self-assessment systems and instant feedback. Using this method, you can bridge the gap between you and an expert in a new and exciting way. Instead of just reading a book, taking notes, or watching a video, you are there in the moment with the so-called "live" person. This experience brings the content to life. And when something is brought to life, it makes it much easier to remember.



### Contact Us

Enter your name

Enter your phone

Enter your email

Ellaborate your concern

[←](#) [→](#) [⟳](#) 127.0.0.1:5500/index.html

e Learn Home About Services Contact Us Search

Enter your name

Enter your phone

Enter your email

Ellaborate your concern

Submit

Copyright © 2023 E-learning.com - All rights reserved

**Conclusion:** In this assignment we have add CSS to the HTML file made in Assignment #1. With the help of CSS we make the navbar sticky. For making attractive images we set border-radius and border for it. We made attractive buttons using hover effects. We made navigation in navbar using scroll-behavior is smooth. So, enhance the webpage using CSS.

## Assignment – 3

**Aim:** Develop a web page using the Bootstrap framework. Grid system, Forms, Button, Navbar, Breadcrumb, Jumbotron should be used.

**LO Mapped:** LO3

### Theory:

Code for webpage:

Index.html file:

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>CodeKnights - Heaven for Programmers</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">
</head>

<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
            <a class="navbar-brand" href="/index.html">CodeKnights</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="#">Home
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">About
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Services
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Contact
                    </li>
                </ul>
                <form class="d-flex" role="search">
                    <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search" style="width: 150px;">
                    <button class="btn btn-outline-success" type="submit">Search</button>
                </form>
            </div>
        </div>
    </nav>
    <div class="container mt-5" style="text-align: center;">
        <h1>Welcome to CodeKnights</h1>
        <p>Heaven for Programmers</p>
        
        <p>Bootstrap is a popular front-end framework for developing responsive, mobile-first web pages. It features a flexible grid system, pre-designed components like buttons and forms, and a range of utility classes for styling and layout. By using Bootstrap, you can quickly build professional-looking websites without having to start from scratch.</p>
    </div>
</body>
```

```
aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
<li class="nav-item">
<a class="nav-link active" aria-current="page" href="/index.html">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" href="/about.html">About</a>
</li>
<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle" href="#" role="button" data-bs-
toggle="dropdown"
aria-expanded="false">
Topics
</a>
<ul class="dropdown-menu">
<li><a class="dropdown-item" href="/technology.html">Technology</a></li>
<li><a class="dropdown-item" href="#">Web Development</a></li>
<li>
<hr class="dropdown-divider">
</li>
<li>
<a class="dropdown-item" href="#">Support</a>
<a class="dropdown-item" href="#">Write for us</a>
</li>
</ul>
</li>
<li class="nav-item">
```

```
<a class="nav-link" href="/contact.html">Contact Us</a>
</li>
</ul>
<form class="d-flex" role="search">
    <input class="form-control me-2" type="search" placeholder="Search" aria-
label="Search">
        <button class="btn btn-outline-light" type="submit">Search</button>
    </form>
    <div class="mx-2">
        <button class="btn btn-danger" data-bs-toggle="modal" data-bs-
target="#loginModal">Login</button>
        <button class="btn btn-danger" data-bs-toggle="modal" data-bs-
target="#signupModal">SignUp</button>
    </div>
    </div>
</div>
</nav>

<div class="modal fade" id="loginModal" tabindex="-1" aria-
labelledby="loginModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5" id="loginModalLabel">Login to CodeKnights</h1>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
            </div>
            <div class="modal-body">
                <form>
                    <div class="mb-3">
                        <label for="exampleInputEmail1" class="form-label">Email
address</label>
```

```
<input type="email" class="form-control" id="exampleInputEmail1"
       aria-describedby="emailHelp">
    <div id="emailHelp" class="form-text">We'll never share your email with
anyone else.</div>
</div>
<div class="mb-3">
    <label for="exampleInputPassword1" class="form-
label">Password</label>
    <input type="password" class="form-control"
id="exampleInputPassword1">
</div>
<div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me
out</label>
</div>
<button type="submit" class="btn btn-primary">Login</button>
</form>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>
</div>
</div>
</div>
```

```
<div class="modal fade" id="signupModal" tabindex="-1" aria-
labelledby="signupModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
```

```
<h1 class="modal-title fs-5" id="signupModalLabel">Get an CodeKnights  
account</h1>  
  
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-  
label="Close"></button>  
  
</div>  
  
<div class="modal-body">  
  
<form>  
  
<div class="mb-3">  
  
<label for="exampleInputEmail1" class="form-label">Email address</label>  
  
<input type="email" class="form-control" id="exampleInputEmail1" aria-  
describedby="emailHelp">  
  
<div id="emailHelp" class="form-text">We'll never share your email with  
anyone else.</div>  
  
</div>  
  
<div class="mb-3">  
  
<label for="exampleInputPassword1" class="form-label">Password</label>  
  
<input type="password" class="form-control"  
id="exampleInputPassword1">  
  
</div>  
  
<div class="mb-3">  
  
<label for="cexampleInputPassword1" class="form-label">Confirm  
Password</label>  
  
<input type="password" class="form-control"  
id="cexampleInputPassword1">  
  
</div>  
  
<button type="submit" class="btn btn-primary">Create Account</button>  
  
</form>  
  
</div>  
  
<div class="modal-footer">  
  
<button type="button" class="btn btn-secondary" data-bs-  
dismiss="modal">Close</button>  
  
</div>
```

```
</div>
</div>
</div>

<div id="carouselExampleCaptions" class="carousel slide carousel-fade">
  <div class="carousel-indicators">
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0" class="active"
      aria-current="true" aria-label="Slide 1"></button>
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1"
      aria-label="Slide 2"></button>
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="2"
      aria-label="Slide 3"></button>
  </div>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption d-none d-md-block">
        <h2>Welcome to CodeKnights</h2>
        <p>Technology News, Development and Trends</p>
        <button class="btn btn-danger">Technology</button>
        <button class="btn btn-primary">Web Development</button>
        <button class="btn btn-success">Tech Fun</button>
      </div>
    </div>
    <div class="carousel-item">
```

```


<div class="carousel-caption d-none d-md-block">
  <h2>The Best Coding Blog</h2>
  <p>Technology News, Development and Trends</p>
  <button class="btn btn-danger">Technology</button>
  <button class="btn btn-primary">Web Development</button>
  <button class="btn btn-success">Tech Fun</button>
</div>
</div>

<div class="carousel-item">
  
  <div class="carousel-caption d-none d-md-block">
    <h2>Award Winnig Blog</h2>
    <p>Technology News, Development and Trends</p>
    <button class="btn btn-danger">Technology</button>
    <button class="btn btn-primary">Web Development</button>
    <button class="btn btn-success">Tech Fun</button>
  </div>
</div>
</div>

<button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleCaptions"
  data-bs-slide="prev">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Previous</span>
</button>

<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleCaptions"
  data-bs-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
```

```
<span class="visually-hidden">Next</span>
</button>
</div>

<div class="container my-4">
  <div class="row mb-2">
    <div class="col-md-6">
      <div
        class="row g-0 border rounded overflow-hidden flex-md-row mb-4 shadow-sm h-md-250 position-relative">
        <div class="col p-4 d-flex flex-column position-static">
          <strong class="d-inline-block mb-2 text-primary">World</strong>
          <h3 class="mb-0">Global Conferences</h3>
          <div class="mb-1 text-body-secondary">Nov 12</div>
          <p class="card-text mb-auto">This is a wider card with supporting text below as a natural lead-in to additional content.</p>
          <a href="#" class="icon-link gap-1 icon-link-hover stretched-link">
            Continue reading
            <svg class="bi">
              <use xlink:href="#chevron-right"></use>
            </svg>
          </a>
        </div>
        <div class="col-auto d-none d-lg-block">
          
        </div>
      </div>
    <div class="col-md-6">
```

```
<div
    class="row g-0 border rounded overflow-hidden flex-md-row mb-4 shadow-sm
h-md-250 position-relative">

    <div class="col p-4 d-flex flex-column position-static">

        <strong class="d-inline-block mb-2 text-success">Design</strong>

        <h3 class="mb-0">Learn Designing</h3>

        <div class="mb-1 text-body-secondary">Nov 11</div>

        <p class="mb-auto">This is a wider card with supporting text below as a
natural lead-in to

            additional content.</p>

        <a href="#" class="icon-link gap-1 icon-link-hover stretched-link">
            Continue reading
            <svg class="bi">
                <use xlink:href="#chevron-right"></use>
            </svg>
        </a>
    </div>

    <div class="col-auto d-none d-lg-block">
        
    </div>
    </div>
</div>

<div class="row mb-2">
    <div class="col-md-6">
        <div
            class="row g-0 border rounded overflow-hidden flex-md-row mb-4 shadow-sm
h-md-250 position-relative">

            <div class="col p-4 d-flex flex-column position-static">
```

```
<strong class="d-inline-block mb-2 text-danger">FrontEnd</strong>
<h3 class="mb-0">Bootstrap Templates</h3>
<div class="mb-1 text-body-secondary">Nov 12</div>
<p class="card-text mb-auto">This is a wider card with supporting text below
as a natural
lead-in to additional content.</p>
<a href="#" class="icon-link gap-1 icon-link-hover stretched-link">
    Continue reading
    <svg class="bi">
        <use xlink:href="#chevron-right"></use>
    </svg>
</a>
</div>
<div class="col-auto d-none d-lg-block">
    
</div>
</div>
<div class="col-md-6">
    <div
        class="row g-0 border rounded overflow-hidden flex-md-row mb-4 shadow-sm
h-md-250 position-relative">
        <div class="col p-4 d-flex flex-column position-static">
            <strong class="d-inline-block mb-2 text-warning">Python</strong>
            <h3 class="mb-0">Learn Python</h3>
            <div class="mb-1 text-body-secondary">Nov 11</div>
            <p class="mb-auto">This is a wider card with supporting text below as a
natural lead-in to
            additional content.</p>
        <a href="#" class="icon-link gap-1 icon-link-hover stretched-link">
```

[Continue reading](#)

```
<svg class="bi">
  <use xlink:href="#chevron-right"></use>
</svg>
</a>
</div>
<div class="col-auto d-none d-lg-block">
  
</div>
</div>
</div>
</div>
</div>
```

```
<footer class="container">
```

```
  <p class="float-end"><a href="#">Back to top</a></p>
  <p>© 2017–2023 CodeKnights, Inc. · <a href="#">Privacy</a> · <a
    href="#">Terms</a></p>
</footer>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j68SIy3Te4Bkz"
  crossorigin="anonymous"></script>
</body>
```

```
</html>
```

about.html file:

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>CodeKnights - Heaven for Programmers</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUalbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">
</head>

<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">CodeKnights</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="/ind.html">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link active" href="/about.html">About</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
</body>
```

```
</li>

<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-
    toggle="dropdown"
        aria-expanded="false">
        Topics
    </a>
    <ul class="dropdown-menu">
        <li><a class="dropdown-item" href="/technology.html">Technology</a></li>
        <li><a class="dropdown-item" href="#">Web Development</a></li>
        <li>
            <hr class="dropdown-divider">
        </li>
        <li>
            <a class="dropdown-item" href="#">Support</a>
            <a class="dropdown-item" href="#">Write for us</a>
        </li>
    </ul>
</li>

<li class="nav-item">
    <a class="nav-link" href="/contact.html">Contact Us</a>
</li>
</ul>

<form class="d-flex" role="search">
    <input class="form-control me-2" type="search" placeholder="Search" aria-
    label="Search">
    <button class="btn btn-outline-success" type="submit">Search</button>
</form>
</div>
</div>
```

```
</nav>
```

```
<div class="container">
  <div class="container my-4">
    <div class="row featurette d-flex justify-content-center align-items-center">
      <div class="col-md-7">
        <h2 class="featurette-heading fw-normal lh-1">It all started with CodeKnights idea. <span
          class="text-body-secondary">It'll blow your mind.</span></h2>
        <p class="lead">Some great placeholder content for the first featurette here. Imagine some exciting
          prose here.</p>
      </div>
      <div class="col-md-5">
        
      </div>
    </div>
  </div>

<div class="container my-4">
  <div class="row featurette d-flex justify-content-center align-items-center">
    <div class="col-md-7 order-md-2">
      <h2 class="featurette-heading fw-normal lh-1">We started CodeKnights in 1977. <span
        class="text-body-secondary">It'll blow your mind.</span></h2>
      <p class="lead">Some great placeholder content for the first featurette here. Imagine some exciting
        prose here.</p>
    </div>
  </div>
```

```
<div class="col-md-5 order-md-1">
  
</div>
</div>
</div>

<div class="container my-4">
  <div class="row featurette d-flex justify-content-center align-items-center">
    <div class="col-md-7">
      <h2 class="featurette-heading fw-normal lh-1">First featurette heading. <span
          class="text-body-secondary">It'll blow your mind.</span></h2>
      <p class="lead">Some great placeholder content for the first featurette here.
      Imagine some exciting
        prose here.</p>
    </div>
    <div class="col-md-5">
      
    </div>
  </div>
</div>

<div class="container">
  <p class="float-end"><a href="#">Back to top</a></p>
  <p>© 2017–2023 CodeKnights, Inc. · <a href="#">Privacy</a> · <a
      href="#">Terms</a></p>
</div>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
       integrity="sha384-
geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"
       crossorigin="anonymous"></script>
</body>

</html>
```

technology.html file:

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>CodeKnights - Heaven for Programmers</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
        rel="stylesheet"
        integrity="sha384-
9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"
        crossorigin="anonymous">
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">CodeKnights</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
             data-bs-target="#navbarSupportedContent" aria-
             controls="navbarSupportedContent" aria-expanded="false"
             aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
```

```
</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav me-auto mb-2 mb-lg-0">
    <li class="nav-item">
      <a class="nav-link" aria-current="page" href="/">Home</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="/about.html">About</a>
    </li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle active" href="#" role="button" data-bs-
        toggle="dropdown"
        aria-expanded="false">
        Topics
      </a>
      <ul class="dropdown-menu">
        <li><a class="dropdown-item"
          href="/technology.html">Technology</a></li>
        <li><a class="dropdown-item" href="#">Web Development</a></li>
        <li>
          <hr class="dropdown-divider">
        </li>
        <li>
          <a class="dropdown-item" href="#">Support</a>
          <a class="dropdown-item" href="#">Write for us</a>
        </li>
      </ul>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="/contact.html">Contact Us</a>
    </li>
```

```
</ul>

<form class="d-flex" role="search">
  <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
  <button class="btn btn-outline-light" type="submit">Search</button>
</form>

<div class="mx-2">
  <button class="btn btn-danger" data-bs-toggle="modal" data-bs-target="#loginModal">Login</button>
  <button class="btn btn-danger" data-bs-toggle="modal" data-bs-target="#signupModal">SignUp</button>
</div>
</div>
</div>

</nav>

<div class="container my-4 ">
  <h1 class="text-center">Emerging Technologies</h1>
  
</div>
<div class="container">
  <div class="container my-4">
    <div class="row featurette d-flex justify-content-center align-items-center">
      <div class="col-md-7">
        <h2 class="featurette-heading fw-normal lh-1">It all started with CodeKnights idea. <span
          class="text-body-secondary">It'll blow your mind.</span></h2>
        <p class="lead">Some great placeholder content for the first featurette here. Imagine some exciting
          prose here.</p>
      
```

```
</div>

<div class="col-md-5">
  
</div>
</div>
</div>

<div class="container my-4">
  <div class="row featurette d-flex justify-content-center align-items-center">
    <div class="col-md-7 order-md-2">
      <h2 class="featurette-heading fw-normal lh-1">We started CodeKnights in 1977.<br/>
<span
      class="text-body-secondary">It'll blow your mind.</span></h2>
      <p class="lead">Some great placeholder content for the first featurette here.<br/>
Imagine some exciting
      prose here.</p>
    </div>
    <div class="col-md-5 order-md-1">
      
    </div>
  </div>
</div>

<div class="container my-4">
  <div class="row featurette d-flex justify-content-center align-items-center">
    <div class="col-md-7">
      <h2 class="featurette-heading fw-normal lh-1">First featurette heading. <span
          class="text-body-secondary">It'll blow your mind.</span></h2>
```

```

<p class="lead">Some great placeholder content for the first featurette here.  

Imagine some exciting  

    prose here.</p>  

</div>  

<div class="col-md-5">  

</div>  

</div>  

</div>  

</div>  

</body>  

</html>

```

contact.html file:

```

<!doctype html>  

<html lang="en">  

<head>  

    <meta charset="utf-8">  

    <meta name="viewport" content="width=device-width, initial-scale=1">  

    <title>CodeKnights - Heaven for Programmers</title>  

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"  

        rel="stylesheet"  

        integrity="sha384-  

        9ndCyUalbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"  

        crossorigin="anonymous">  

</head>  

<body>  

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

```

```
<div class="container-fluid">
  <a class="navbar-brand" href="#">CodeKnights</a>
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
    data-bs-target="#navbarSupportedContent" aria-
    controls="navbarSupportedContent" aria-expanded="false"
    aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
      <li class="nav-item">
        <a class="nav-link" aria-current="page" href="/">Home</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="/about.html">About</a>
      </li>
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-
        toggle="dropdown"
          aria-expanded="false">
          Topics
        </a>
        <ul class="dropdown-menu">
          <li><a class="dropdown-item"
            href="/technology.html">Technology</a></li>
          <li><a class="dropdown-item" href="#">Web Development</a></li>
          <li>
            <hr class="dropdown-divider">
          </li>
          <li>
            <a class="dropdown-item" href="#">Support</a>
          </li>
        </ul>
      </li>
    </ul>
  </div>
</div>
```

```
<a class="dropdown-item" href="#">Write for us</a>
</li>
</ul>
</li>
<li class="nav-item">
    <a class="nav-link active" href="/contact.html">Contact Us</a>
</li>
</ul>
<form class="d-flex" role="search">
    <input class="form-control me-2" type="search" placeholder="Search" aria-
label="Search">
        <button class="btn btn-outline-success" type="submit">Search</button>
    </form>
</div>
</div>
</nav>

<div class="container my-4">
    <h1>Contact Us</h1>
    <form class="row g-3">
        <div class="col-md-6">
            <label for="inputEmail4" class="form-label">Email</label>
            <input type="email" class="form-control" id="inputEmail4">
        </div>
        <div class="col-md-6">
            <label for="inputPassword4" class="form-label">Password</label>
            <input type="password" class="form-control" id="inputPassword4">
        </div>
        <div class="col-12">
            <label for="inputAddress" class="form-label">Address</label>
```

```
<input type="text" class="form-control" id="inputAddress" placeholder="1234  
Main St">  
</div>  
    <label for="inputCity" class="form-label">City</label>  
    <input type="text" class="form-control" id="inputCity">  
</div>  
    <label for="inputState" class="form-label">State</label>  
    <select id="inputState" class="form-select">  
        <option selected>Choose</option>  
        <option>Andhra Pradesh</option>  
        <option>Arunachal Pradesh</option>  
        <option>Assam</option>  
        <option>Bihar</option>  
        <option>Chhattisgarh</option>  
        <option>Goa</option>  
        <option>Gujarat</option>  
        <option>Haryana</option>  
        <option>Himachal Pradesh</option>  
        <option>Jammu and Kashmir</option>  
        <option>Jharkhand</option>  
        <option>Karnataka</option>  
        <option>Kerala</option>  
        <option>Madhya Pradesh</option>  
        <option>Maharashtra</option>  
        <option>Manipur</option>  
        <option>Meghalaya</option>  
        <option>Mizoram</option>  
        <option>Nagaland</option>  
        <option>Odisha</option>
```

```
<option>Punjab</option>
<option>Rajasthan</option>
<option>Sikkim</option>
<option>Tamil Nadu</option>
<option>Telangana</option>
<option>Tripura</option>
<option>West Bengal</option>
</select>
</div>
<div class="col-12">
<div class="form-check">
<input class="form-check-input" type="checkbox" id="gridCheck">
<label class="form-check-label" for="gridCheck">
    Check me out
</label>
</div>
</div>
<div class="col-12">
    <button type="submit" class="btn btn-primary">Submit</button>
</div>
</form>
</div>

<footer class="container">
    <p class="float-end"><a href="#">Back to top</a></p>
    <p>© 2017–2023 CodeKnights, Inc. · <a href="#">Privacy</a> · <a href="#">Terms</a></p>
</footer>

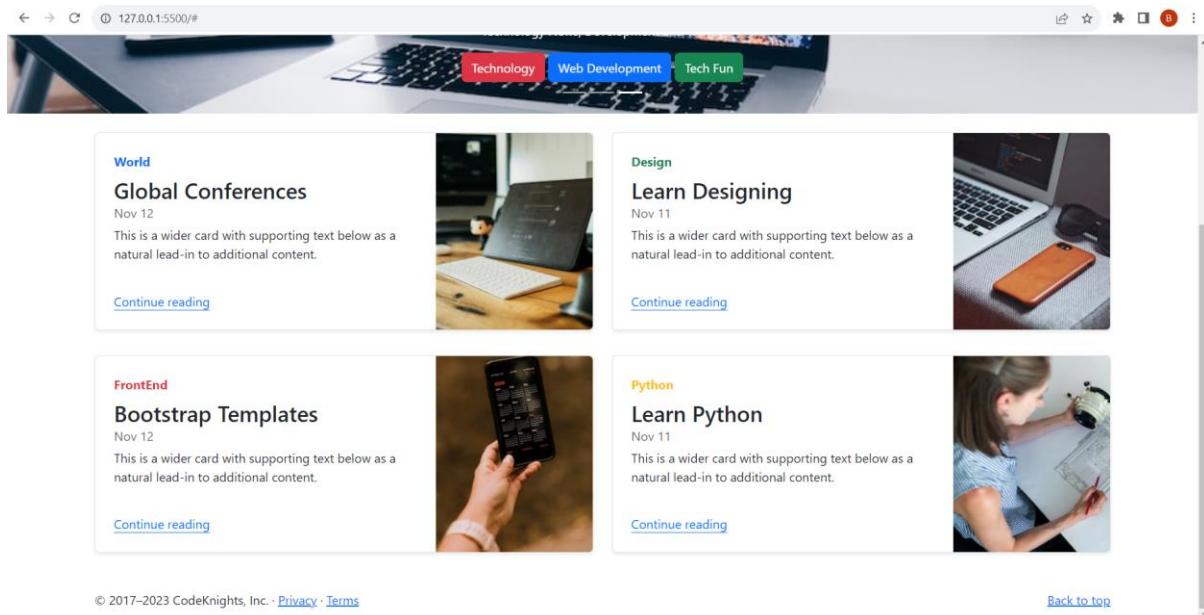
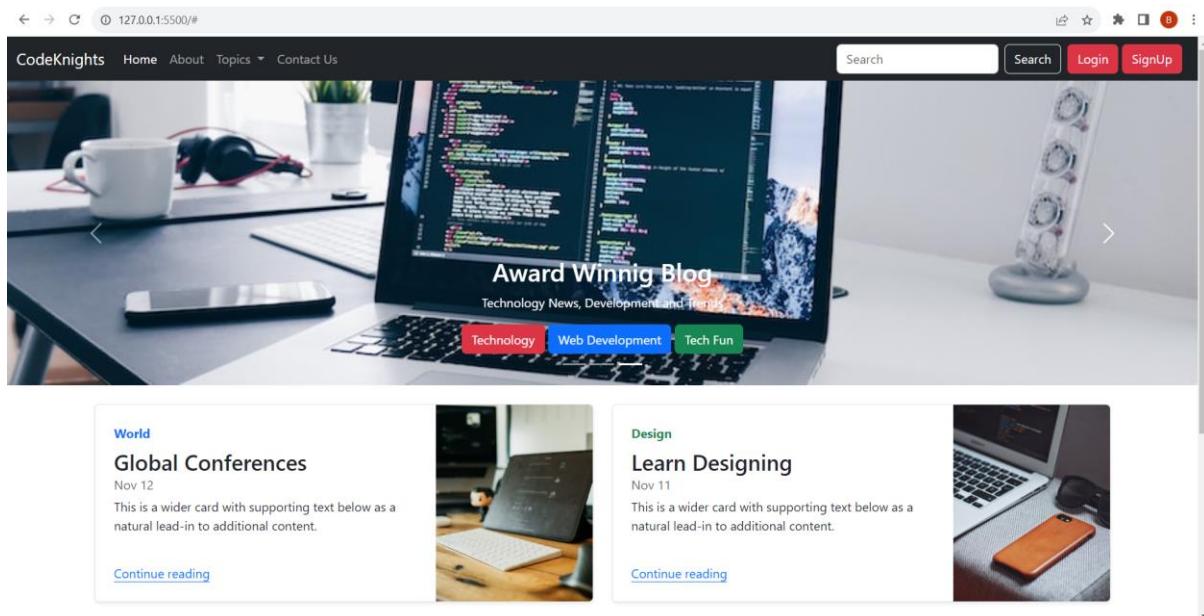
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
```

```

integrity="sha384-
geWF76RCwLtnZ8qwWowPQNgul3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"
crossorigin="anonymous">></script>
</body>
</html>

```

## Screenshots:



← → ⌂ 127.0.0.1:5500/about.html

CodeKnights Home About Topics ▾ Contact Us

Search

**It all started with CodeKnights idea. It'll blow your mind.**

Some great placeholder content for the first featurette here. Imagine some exciting prose here.



**We started CodeKnights in 1977. It'll blow your mind.**

Some great placeholder content for the first featurette here. Imagine some exciting prose here.

← → ⌂ 127.0.0.1:5500/about.html

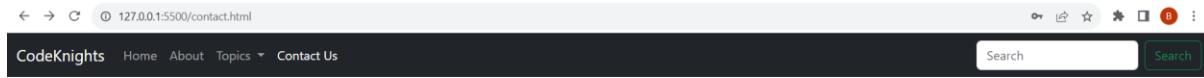
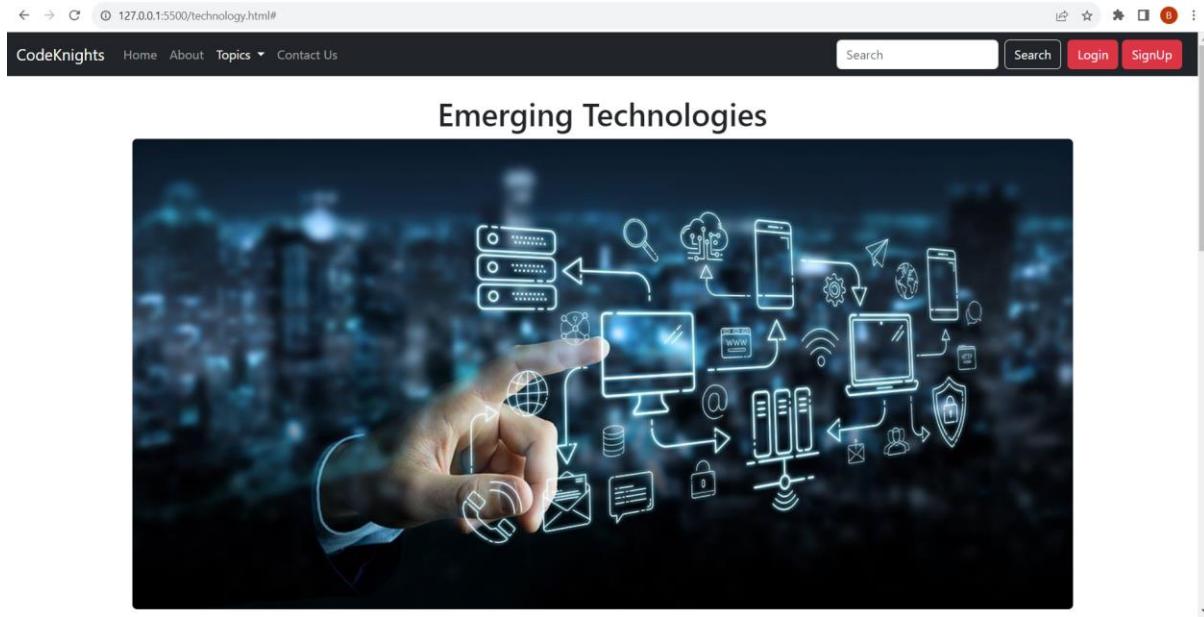
**We started CodeKnights in 1977. It'll blow your mind.**

Some great placeholder content for the first featurette here. Imagine some exciting prose here.

**First featurette heading. It'll blow your mind.**

Some great placeholder content for the first featurette here. Imagine some exciting prose here.





**Conclusion:** In this assignment we have learned Bootstrap and developed a web page using Bootstrap framework. We use following elements to develop a web page :- Navbar, Carousel, Breadcrumb, Jumbotron, Grid system, Forms, Button.

## Assignment No. – 4A

**Aim:** WAP in JS to study conditional Statements, Loops and Functions

**LO Mapped:** LO - 4

### Theory:

Code:

```
// Conditional Statements

function checkNumber(number) {
    if (number > 0) {
        return "Positive";
    } else if (number < 0) {
        return "Negative";
    } else {
        return "Zero";
    }
}
```

```
// Loops

function printNumbers(limit) {
    console.log("Numbers from 1 to " + limit + ":");
    for (let i = 1; i <= limit; i++) {
        console.log(i);
    }
}
```

```
// Functions

function calculateFactorial(number) {
    if (number === 0 || number === 1) {
```

```

        return 1;
    }
    else if(number<0){
        return " --> Given number is negative. So, we don't calculate its factorial";
    }
    return number * calculateFactorial(number - 1);
}

const numToCheck = 5;
console.log(numToCheck + " is " + checkNumber(numToCheck));
const numbersToPrint = 10;
printNumbers(numbersToPrint);
const factorialNumber = 5;
console.log("Factorial of " + factorialNumber + " is " + calculateFactorial(factorialNumber));

```

Output:

```

PS C:\Users\DELL\Desktop\practice> node "c:\Users\DELL\Desktop\practice\assign-4A.js"
5 is Positive
Numbers from 1 to 10:
1
2
3
4
5
6
7
8
9
10
Factorial of 5 is 120
PS C:\Users\DELL\Desktop\practice>

```

### Conclusion:

In this assignment we learn about conditional Statements, Loops and Functions and implement it in program.

## **Assignment No. - 4B**

**Aim:** WAP on inheritance, Iterators and Generators

**LO Mapped:** LO – 4

### **Theory:**

Code:

```
class Engine {  
    constructor(type) {  
        this.type = type;  
    }  
}
```

```
start() {  
    console.log(` ${this.type} engine started. `);  
}  
}
```

```
class V6Engine extends Engine {  
    constructor() {  
        super("FORD");  
    }  
}
```

```
class V8Engine extends Engine {  
    constructor() {  
        super("BMW");  
    }  
}
```

```
class V12Engine extends Engine {  
    constructor() {  
        super("TOYOTA");  
    }  
}  
  
class EngineIterator {  
    constructor(engines) {  
        this.engines = engines;  
        this.currentIndex = 0;  
    }  
  
    *next() {  
        while (this.currentIndex < this.engines.length) {  
            yield this.engines[this.currentIndex];  
            this.currentIndex++;  
        }  
    }  
}  
  
const v6Engine = new V6Engine();  
const v8Engine = new V8Engine();  
const v12Engine = new V12Engine();  
  
const engines = [v6Engine, v8Engine, v12Engine];  
  
const engineIterator = new EngineIterator(engines);  
const iterator = engineIterator.next();  
  
console.log("Using iterator with generator:");
```

```
for (const engine of iterator) {  
    engine.start();  
}
```

Output:

```
PS C:\Users\DELL\Desktop\practice> node "c:\Users\DELL\Desktop\practice\assign-4B.js"  
Using iterator with generator:  
FORD engine started.  
BMW engine started.  
TOYOTA engine started.  
PS C:\Users\DELL\Desktop\practice>
```

### Conclusion:

In this assignment we learn about inheritance, Iterators and Generators. We understood this concepts in JavaScript and implemented in a program.

## Assignment 5a and 5b

**Aim:** - a) Write a JavaScript program to study arrow functions, DOM Manipulation and CSS Manipulation.

b) Write a JavaScript program and implement the concept of Promise and fetch and asynchronous JavaScript.

**LO Mapped:** - LO-4

**Theory:** -

**A)**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Arrow Function, DOM, and CSS Manipulation</title>
    <style>
        .box {
            width: 200px;
            height: 200px;
            background-color: #3498db;
            color: #fff;
            text-align: center;
            line-height: 200px;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <div class="box" id="colorBox">Click me!</div>
    <script>

        const colorBox = document.getElementById('colorBox');

        const changeColor = () => {

            const randomColor = '#' + (Math.random().toString(16) + '000000').slice(2, 8);

            colorBox.style.backgroundColor = randomColor;
            colorBox.textContent = randomColor;

            // Log the color change
            console.log(`Background color changed to ${randomColor}`);
        };
    </script>

```

```
    colorBox.addEventListener('click', changeColor);
  </script>
</body>
</html>
```

**OUTPUT: -**



**B)**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Promise and Fetch Example</title>
</head>
<body>
  <h1>Fetching Data</h1>
  <div id="data-container"></div>

<script>

  function fetchData() {
    const url = 'https://jsonplaceholder.typicode.com/posts/1';

    return fetch(url)
      .then((response) => {
        if (!response.ok) {
```

```

        throw new Error('Network response was not ok');
    }
    return response.json();
});
}
}

async function handleData() {
try {
    const response = await fetchData(); // Wait for the Promise to resolve
    const dataContainer = document.getElementById('data-container');
    dataContainer.innerHTML = `<p>Title: ${response.title}</p><p>Body:  
${response.body}</p>`;
} catch (error) {
    console.error('Error:', error);
    const dataContainer = document.getElementById('data-container');
    dataContainer.innerHTML = '<p>An error occurred while fetching the data.</p>';
}
}

handleData();
</script>
</body>
</html>

```

#### **OUTPUT: -**



**Conclusion:** - We thoroughly studied about arrow functions and CSS and DOM manipulation and also implemented the concept of promises, fetch and asynchronous JavaScript.

## **ASSIGNMENT - 6**

**Aim:** WAP to implement the concept of forms and events.

**LO Mapped:** LO5

### **Theory:**

In web development, forms are a fundamental way to collect and interact with user input. React, a popular JavaScript library for building user interfaces, provides a convenient way to manage form elements and their events. In this theoretical overview, we'll discuss how to create a registration form in React, incorporating various form elements and handling events such as onSubmit, onClick, and onKeyDown.

React and Forms

React Components

React organizes user interfaces into components. A component is a reusable piece of UI, often representing a part of the user interface. Components can be composed of other components, making it easy to build complex applications.

#### 1. Click Event:

- This event occurs when a user clicks an element, such as a button or a link.
- You can handle click events in React using the onClick attribute on elements like buttons or links. For example:

```
<button onClick={handleClick}>Click me</button>
```

#### 2. Change Event:

- Change events are commonly used with form elements like textboxes and checkboxes.
- You can handle change events using the onChange attribute. For example:

```
<input type="text" onChange={handleChange} />
```

Now, let's discuss textbox and textarea in React:

- Textbox (Input):

- In React, you can create a textbox (input field) using the `<input>` element.
- To capture user input from a textbox, you typically use the `value` and `onChange` attributes.

```
<input type="text" value={this.state.textValue} onChange={this.handleChange}>/>
```

In this example, `this.state.textValue` is used to control the value of the textbox, and `this.handleChange` is the event handler for changes in the textbox.

- Textarea:

- A textarea in React is created using the `<textarea>` element.
- Similar to a textbox, you can control the content of a textarea using the `value` and `onChange` attributes.

```
<textarea value={this.state.textareaValue} onChange={this.handleTextareaChange}></textarea>
```

In this case, `this.state.textareaValue` stores the content of the textarea, and `this.handleTextareaChange` is the event handler for changes in the textarea.

You'll need to define the corresponding event handlers (`handleChange` and `handleTextareaChange` in this case) in your component to handle user input and update the component's state accordingly.

Remember that in React, the component's state should be used to manage and reflect the current value of form elements like textboxes and textareas, and event handlers help update this state when user interactions occur.

Code:

RegistrationForm.js

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

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

const RegistrationForm = () => {
  const [formData, setFormData] = useState({
    name: '',
    email: '',
    message: '',
    gender: 'male',
    subscribe: false
  });

  const handleChange = (e) => {
    const { name, value, type, checked } = e.target;

    setFormData(prevState => ({
      ...prevState,
      [name]: type === 'checkbox' ? checked : value
    }));
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    console.log('Form submitted:', formData);
    // Add further processing logic here
  };

  const handleReset = () => {
    setFormData({
      name: '',
      email: '',
      message: '',
      gender: 'male',
      subscribe: false
    });
  };
}
```

```
    subscribe: false
  });
};

return (
  <form onSubmit={handleSubmit}>
    <div>
      <label htmlFor="name">Name:</label>
      <input
        type="text"
        id="name"
        name="name"
        value={formData.name}
        onChange={handleChange}
        required
      />
    </div>
    <div>
      <label htmlFor="email">Email:</label>
      <input
        type="email"
        id="email"
        name="email"
        value={formData.email}
        onChange={handleChange}
        required
      />
    </div>
    <div>
      <label htmlFor="message">Message:</label>
      <textarea
        id="message"
        name="message"

```

```
        checked={formData.gender === 'female'}
        onChange={handleChange}
      />
      <label htmlFor="female">Female</label>
    </div>
    <div>
      <label htmlFor="subscribe">Subscribe to newsletter:</label>
      <input
        type="checkbox"
        id="subscribe"
        name="subscribe"
        checked={formData.subscribe}
        onChange={handleChange}
      />
    </div>
    <button type="submit">Submit</button>
    <button type="button" onClick={handleReset}>Reset</button>
  </form>
);
};

export default RegistrationForm;
```

## RegistrationForm.css

```
form {  
    max-width: 400px;  
    margin: 0 auto;  
    padding: 20px;  
    border: 2px solid #3498db;  
    border-radius: 10px;  
    background-color: #ecf0f1;  
}  
  
div {  
    margin-bottom: 15px;  
}  
  
label {  
    display: block;  
    font-weight: bold;  
    margin-bottom: 5px;  
    color: #2c3e50;  
}  
  
input,  
textarea {  
    width: calc(100% - 16px);  
    padding: 10px;  
    border: 2px solid #3498db;  
    border-radius: 5px;  
    box-sizing: border-box;  
    font-size: 16px;  
}  
  
input:focus,  
textarea:focus {  
    outline: none;  
}
```

```
padding: 4px;
border: 2px solid #3498db;
border-radius: 5px;
box-sizing: border-box;
font-size: 16px;
}

input:focus,
textarea:focus {
  outline: none;
  border-color: #2980b9;
}

button {
  background-color: #3498db;
  color: #fff;
  border: none;
  padding: 10px 20px;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #2980b9;
}

button[type="reset"] {
  background-color: #e74c3c;
}

button[type="reset"]:hover {
  background-color: #c0392b;
}
```

## Output:

The image shows a web-based form with a light gray background and a prominent blue border. Inside the form, there are three input fields: one for 'Name', one for 'Email', and one for 'Message'. Each field has a blue rectangular outline. Below these fields is a 'Gender' section. It contains two radio button options: 'Male' and 'Female'. The 'Female' option is currently selected, indicated by a blue circle. At the bottom of the form, there is a small, partially visible button.

**Gender:**

Male

Female

**Subscribe to newsletter:**

**Submit**   **Reset**

---

### Conclusion:

We studied that In React, an event is a user interaction or system generated action that can trigger specific behaviour in a React component and created a registration form which consisted of textbox, text area, selection input, check box, radio button, submit button and reset button handling onSubmit, onClick and keyDown events.

## Assignment 7

**Aim:** a. WAP to implement ReactJS Router and Animation.

b. WAP to implement the concept of React Hooks.

**LO Mapped:** LO-5

**Theory:**

**Code: a)**

npm install react-router-dom

```
// Home.js
import React from 'react';

const Home = () => {
  return (
    <div>
      <h1>Home Page</h1>
      <p>Welcome to the Home Page!</p>
    </div>
  );
};

export default Home;
```

```
// About.js

import React from 'react';

const About = () => {

  return (
    <div>
      <h1>About Page</h1>
      <p>About our awesome website!</p>
    </div>
  );
};

export default About;
```

```
// Contact.js

import React from 'react';

const Contact = () => {

  return (
    <div>
      <h1>Contact Page</h1>
      <p>Contact us for more information!</p>
    </div>
  );
};

export default Contact;
```

```
// App.js

import React from 'react';

import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom'; // Updated
import

import Home from './Home';
import About from './About';
import Contact from './Contact';

const App = () => {

  return (
    <Router>
      <div>
        <nav>
          <ul>
            <li>
              <Link to="/">Home</Link>
            </li>
            <li>
              <Link to="/about">About</Link>
            </li>
            <li>
              <Link to="/contact">Contact</Link>
            </li>
          </ul>
        </nav>
      </div>
    </Router>
  )
}

export default App;
```

```
<Route path="/contact" element={<Contact />} />

</Routes>

</div>

</Router>

);

};

export default App;

// index.js

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

ReactDOM.render(<App />, document.getElementById('root'));
```



## Home Page

Welcome to the Home Page!



## About Page

About our awesome website!



**b)**

```
// App.js

import React, { useState } from 'react';
import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';

import Home from './Home';
import About from './About';
import Contact from './Contact';

const App = () => {
  const [isLoggedIn, setIsLoggedIn] = useState(false);
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");

  const handleLogin = () => {
    if (username === 'user' && password === 'password') {
      setIsLoggedIn(true);
    }
  };

  const handleLogout = () => {
    setIsLoggedIn(false);
  };

  return (
    <Router>
      <div>
        <nav>
          <ul>
```

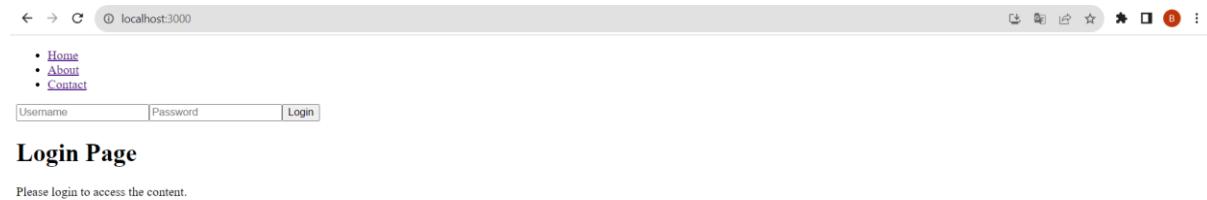
```
<li>
  <Link to="/">Home</Link>
</li>
<li>
  <Link to="/about">About</Link>
</li>
<li>
  <Link to="/contact">Contact</Link>
</li>
</ul>
{isLoggedIn ? (
  <button onClick={handleLogout}>Logout</button>
) : (
  <div>
    <input
      type="text"
      placeholder="Username"
      value={username}
      onChange={(e) => setUsername(e.target.value)}>
    />
    <input
      type="password"
      placeholder="Password"
      value={password}
      onChange={(e) => setPassword(e.target.value)}>
    />
    <button onClick={handleLogin}>Login</button>
  </div>
)}
```

```
</nav>

<Routes>
  <Route
    path="/"
    element={isLoggedin ? <Home /> : <Login />}
  />
  <Route path="/about" element={<About />} />
  <Route path="/contact" element={<Contact />} />
</Routes>
</div>
</Router>
);
};

const Login = () => {
  return (
    <div>
      <h1>Login Page</h1>
      <p>Please login to access the content.</p>
    </div>
  );
};

export default App;
```





## Login Page

Please login to access the content.



## Home Page

Welcome to the Home Page!

## Conclusion:

Routing in React involves managing and rendering different components based on the URL, typically using libraries like React Router to define routes and navigate between them. Hooks are functions that allow you to "hook into" or use state and other React features in functional components. Hooks include useState for managing component state, useEffect for side effects, and many more, enabling more flexibility and reusability in functional components.

## Assignment – 8

**Aim:** Assignment on REPL (Commandline).

**LO Mapped:** LO-6

**Theory:**

1. Program

```
# Define a function to perform calculations

def calculate(expression):

    try:

        result = eval(expression)

        return result

    except Exception:

        return "Invalid input. Please enter a valid arithmetic expression."

# Start the REPL loop

print("Welcome to the Simple Calculator REPL!")

while True:

    user_input = input("Enter your arithmetic expression (e.g., 5 + 3) or type 'exit' to quit: ")

    if user_input.lower() == 'exit':

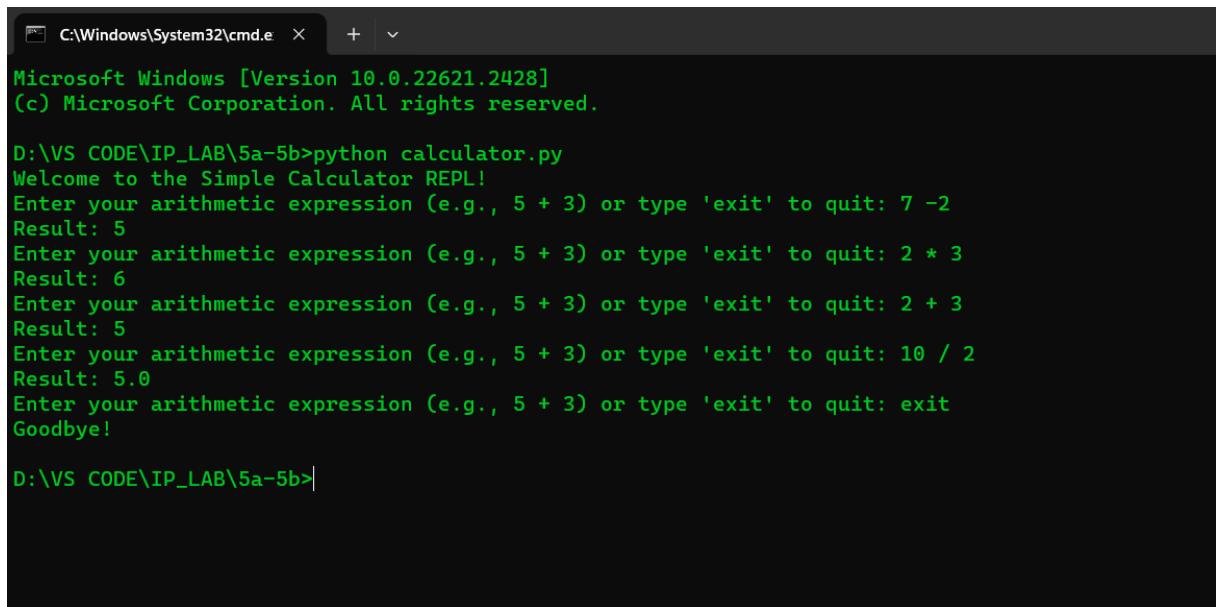
        print("Goodbye!")

        break

    result = calculate(user_input)

    print("Result:", result)
```

## Screenshot:



```
C:\Windows\System32\cmd.e + 
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

D:\VS CODE\IP_LAB\5a-5b>python calculator.py
Welcome to the Simple Calculator REPL!
Enter your arithmetic expression (e.g., 5 + 3) or type 'exit' to quit: 7 -2
Result: 5
Enter your arithmetic expression (e.g., 5 + 3) or type 'exit' to quit: 2 * 3
Result: 6
Enter your arithmetic expression (e.g., 5 + 3) or type 'exit' to quit: 2 + 3
Result: 5
Enter your arithmetic expression (e.g., 5 + 3) or type 'exit' to quit: 10 / 2
Result: 5.0
Enter your arithmetic expression (e.g., 5 + 3) or type 'exit' to quit: exit
Goodbye!

D:\VS CODE\IP_LAB\5a-5b>
```

## Conclusion:

In this assignment, we successfully implemented a basic calculator in a REPL environment using Python. The program enables users to perform arithmetic operations and receive results interactively, with robust error handling for invalid inputs. This assignment not only reinforces foundational concepts of programming, such as user input, calculations, and control flow, but also highlights the importance of a user-friendly interface and error reporting. It serves as a practical exercise in building a simple yet effective command-line tool for performing calculations.

**Name: Bhuvan Sawant**

**Roll no.: 110**

## **Assignment - 9**

**Aim:** WAP to implement Refs in React.

### **Theory:**

In React, refs are a way to access and interact with DOM elements or React components directly. They provide a means to reference a specific element or component within your application, allowing you to read values, trigger actions, or perform other operations that would be difficult to achieve using React's standard data flow and state management mechanisms. Refs can be helpful in scenarios such as handling form input focus, triggering animations, integrating with third-party libraries, or accessing underlying DOM APIs.

Use cases for React refs:

#### **1. Accessing DOM Elements:**

Refs are commonly used to interact with DOM elements directly. For instance, you can access input values, set focus, or trigger animations.

#### **2. Managing Third-Party Libraries:**

When integrating React with third-party libraries that require direct DOM manipulation, refs are essential. For example, if you're using a charting library or a video player, you can use refs to interact with their APIs.

### **3. Animating Elements:**

You can use refs to trigger animations by changing CSS classes or properties. This is useful when you need precise control over animations.

### **4. Scrolling and Measuring Elements:**

You can use refs to scroll to specific elements on the page or measure their dimensions. This is helpful for building features like smooth scrolling or lazy-loading content.

### **5. Integration with Non-React Code:**

If you need to integrate React with non-React code, such as legacy JavaScript or libraries like D3.js, you can use refs to bridge the gap between React components and external code.

### **6. Forms and Input Validation:**

Refs can be useful for form handling and input validation.

You can access input elements directly to check and manipulate their values or apply custom validation logic.

#### **Creating Refs:**

You can create a ref using the **React.createRef()** method (for class components) or the **createRef** hook (for functional components).

#### **Accessing Refs:**

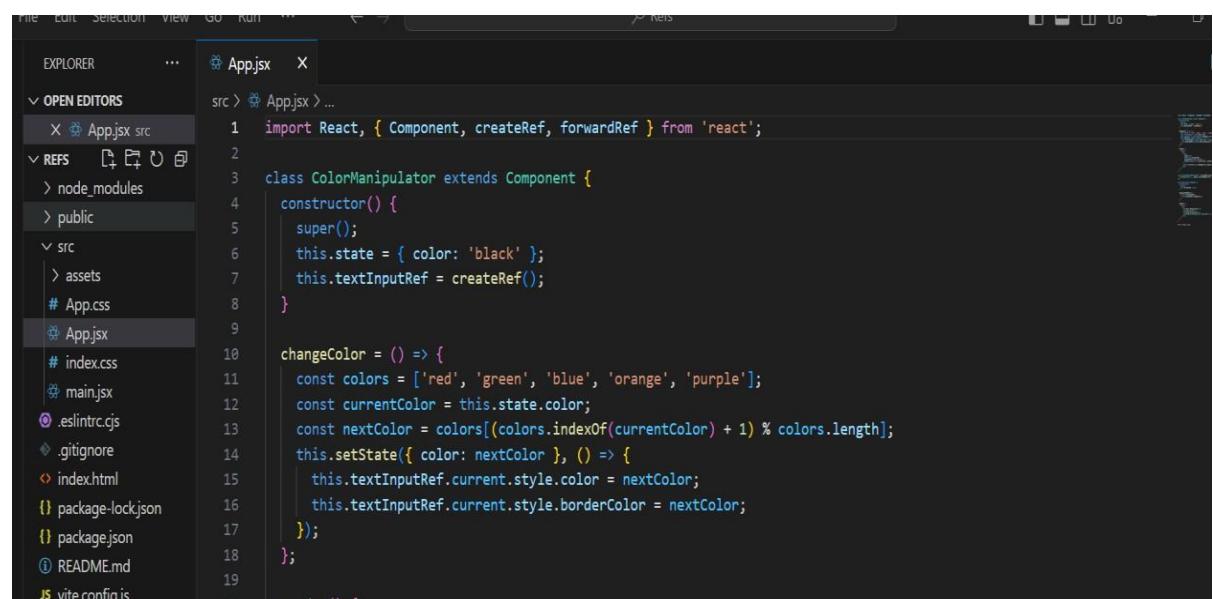
You can access the DOM element or React component associated with a ref by using the **current** property of the ref object. This property holds the reference to the underlying element or component. You typically access the ref inside lifecycle methods (for class components) or within the functional component itself.

## Forwarding Refs:

Sometimes, you may need to pass a ref from a parent component to a child component. This can be achieved using the "forwarding refs" technique. React provides the **forwardRef** function for this purpose. It allows you to pass a ref down the component hierarchy and access the child component's ref from a parent component.

## Callback Refs:

Callback refs are another way to work with refs, especially in cases where you need to perform additional operations when a ref is set. Instead of using the **createRef** or **useRef** functions, you define a callback function that will be called when the ref is attached to an element or component.



The screenshot shows a code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, etc.
- Explorer:** Shows the project structure:
  - OPEN EDITORS: App.jsx
  - REFS: node\_modules, public
  - src
    - assets
    - # App.css
    - App.jsx
    - # index.css
    - main.jsx
    - .eslintrc.cjs
    - .gitignore
    - index.html
    - package-lock.json
    - package.json
    - README.md
    - vite.config.js
- Code Editor:** The App.jsx file is open, showing the following code:

```
1 import React, { Component, createRef, forwardRef } from 'react';
2
3 class ColorManipulator extends Component {
4   constructor() {
5     super();
6     this.state = { color: 'black' };
7     this.textInputRef = createRef();
8   }
9
10 changeColor = () => {
11   const colors = ['red', 'green', 'blue', 'orange', 'purple'];
12   const currentColor = this.state.color;
13   const nextColor = colors[(colors.indexOf(currentColor) + 1) % colors.length];
14   this.setState({ color: nextColor }, () => {
15     this.textInputRef.current.style.color = nextColor;
16     this.textInputRef.current.style.borderColor = nextColor;
17   });
18 }
19 }
```

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists files and folders: node\_modules, public, src (with assets, App.css, App.jsx, index.css, main.jsx, .eslintrc.js, .gitignore, index.html), package-lock.json, package.json, README.md, and vite.config.js. The App.jsx file is selected in the sidebar. The main editor area shows the following code:

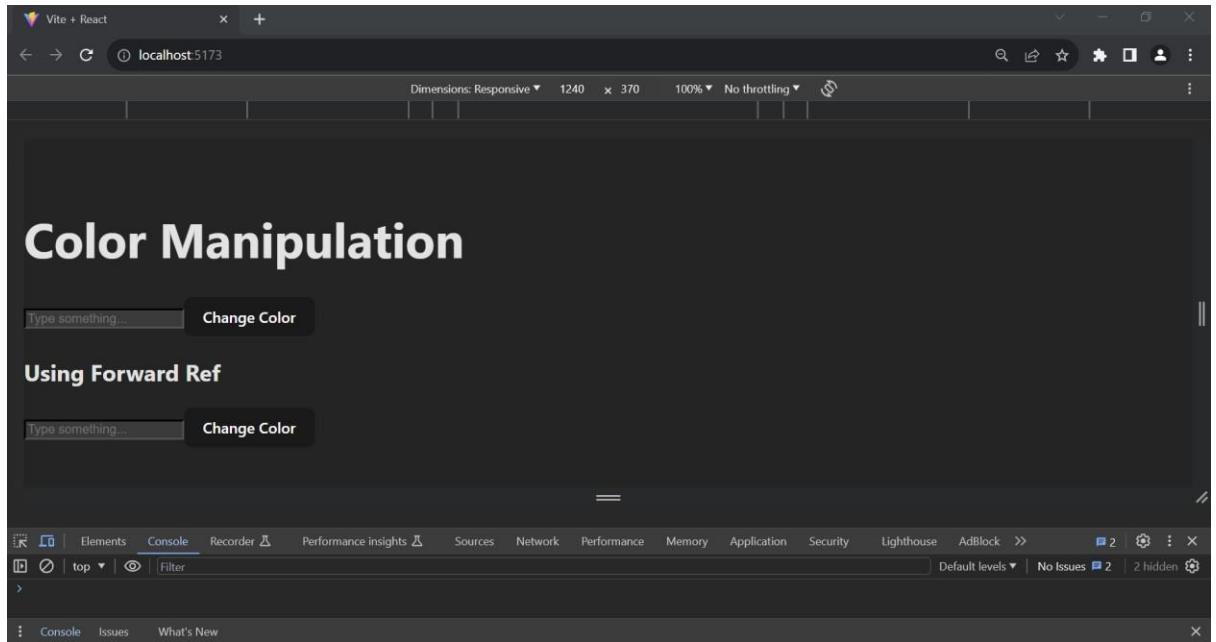
```
20  render() {
21    return (
22      <div>
23        <input
24          type="text"
25          ref={this.textInputRef}
26          placeholder="Type something..."
27          style={{ color: this.state.color, borderColor: this.state.color }}
28        />
29        <button onClick={this.changeColor}>Change Color</button>
30      </div>
31    );
32  }
33
34  const ForwardedColorManipulator = forwardRef((props, ref) => (
35    <ColorManipulator {...props} forwardedRef={ref} />
36  ));
37
38  class App extends Component {
39    constructor() {
40      super();
41      this.callbackRef = null;
42    }
43  }
```

The screenshot shows the same code editor interface. The Explorer sidebar remains the same. The main editor area now includes the following additional code at the bottom:

```
43  }

44  componentDidMount() {
45    if (this.callbackRef) {
46      this.callbackRef.changeColor();
47    }
48  }

49
50  render() {
51    return (
52      <div>
53        <h1>Color Manipulation</h1>
54        <ColorManipulator />
55        <h2>Using Forward Ref</h2>
56        <ForwardedColorManipulator ref={(ref) => (this.callbackRef = ref)} />
57      </div>
58    );
59  }
60
61  }
62
63  export default App;
64
```



## Conclusion:

We successfully demonstrated how React refs can be used to create, access, and manipulate DOM elements and components, showcasing their versatility within a React application.

## ASSIGNMENT – 10

Aim: Create a web application that performs CRUD operations (database connectivity).

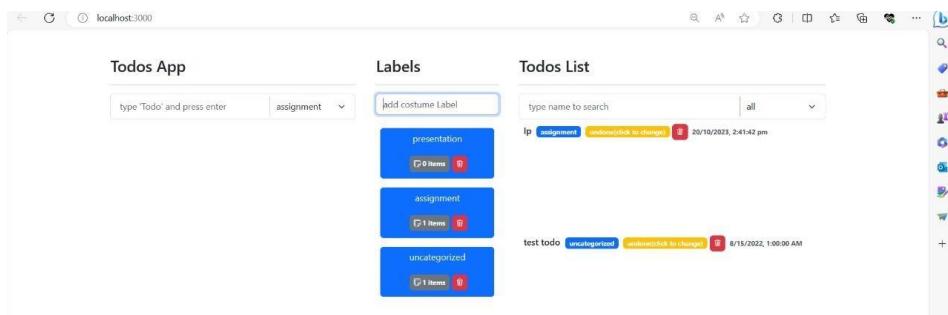
**LO:** LO-6: - Construct Backend Applications using Node.js/Express.

### Theory:

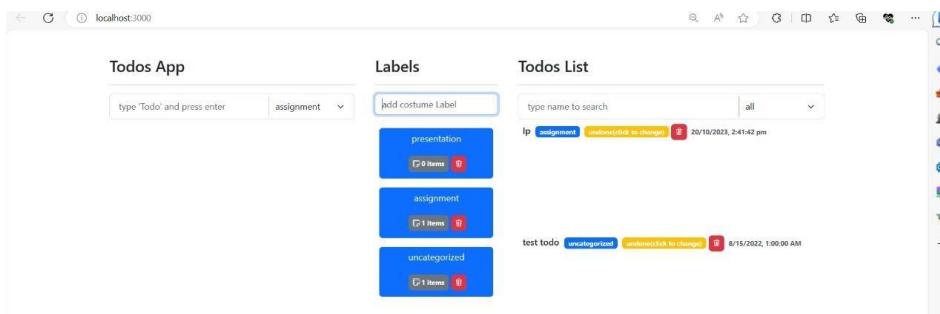
CRUD is the acronym for CREATE, READ, UPDATE and DELETE. These terms describe the four essential operations for creating and managing persistent data elements, mainly in relational and NoSQL databases.

Express is a node js web application framework that provides broad features for building web and mobile applications. It is used to build a single page, multipage, and hybrid web application. It's a layer built on the top of the Node js that helps manage servers and routes.

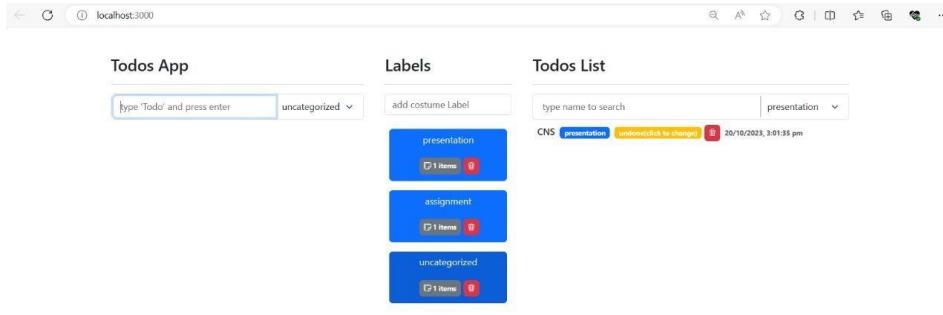
### 1. Creating a To-Do App that can perform all the CRUD operations.



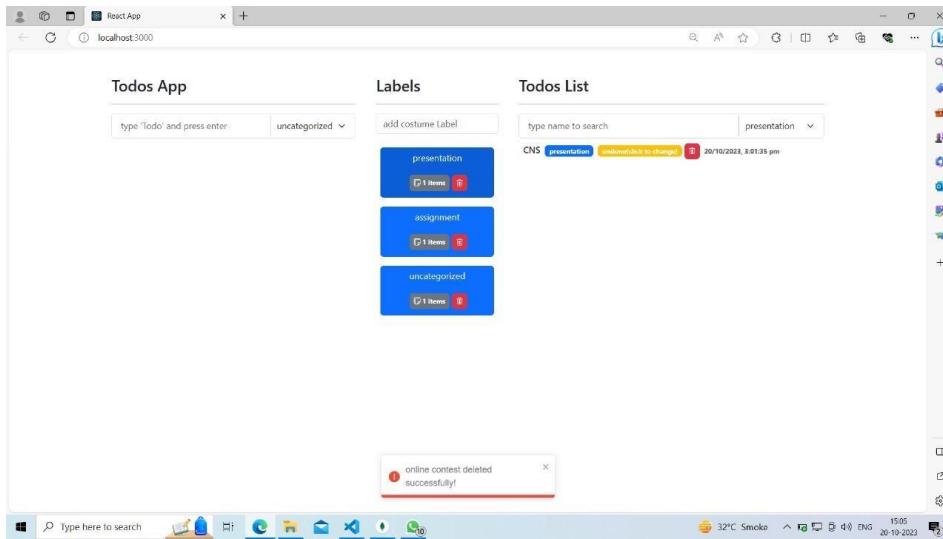
### 2 Adding Labels into the application.



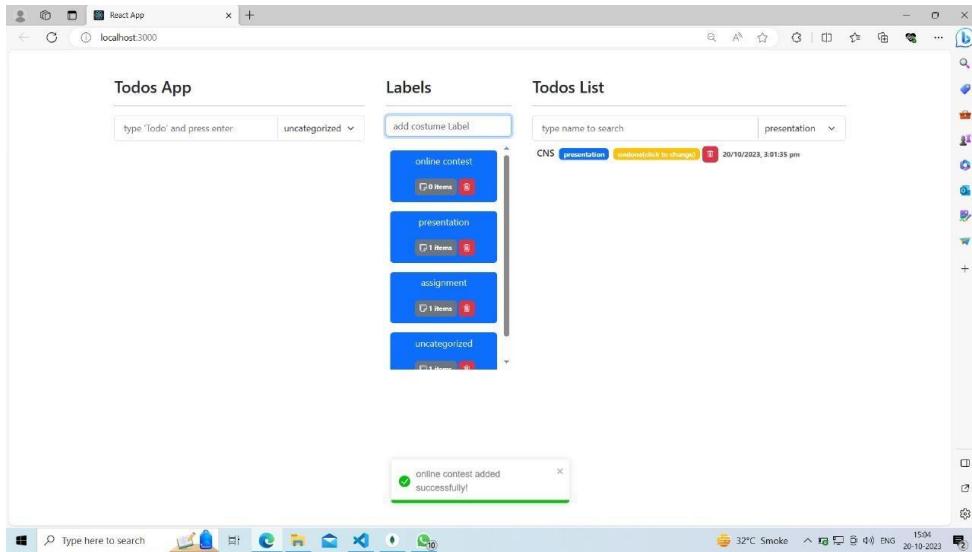
### 3 Adding a list in the To Do Application



### 4 The Online Contest was Deleted Successfully by using this operation.



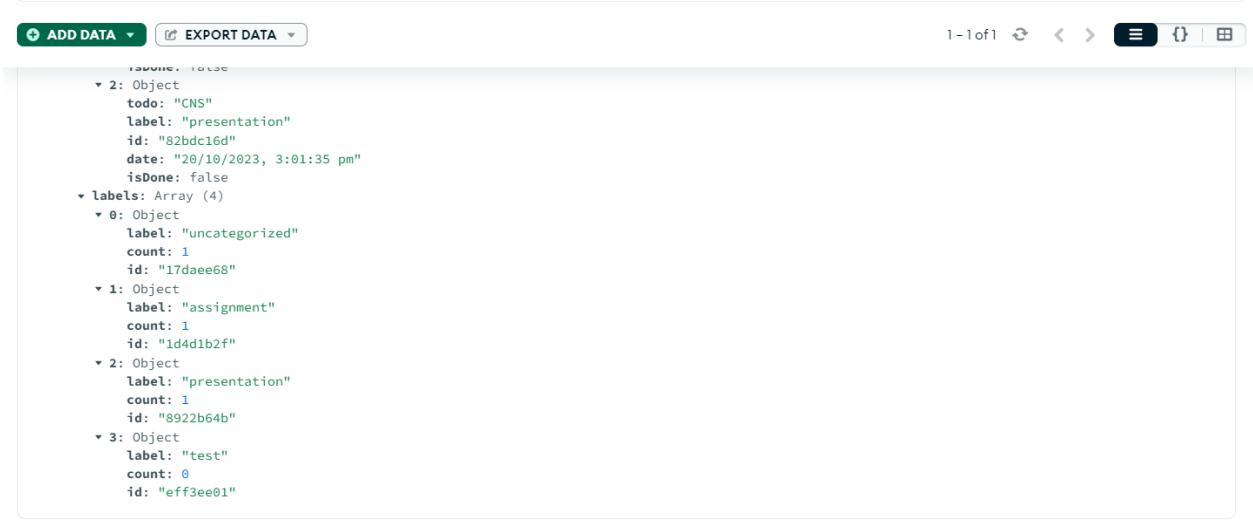
## 5 Adding an Online Contest by making an operation.



## 6 Connectivity of Database

The screenshot shows a MongoDB document in a database interface. The document structure is as follows:

```
_id: ObjectId('6532508f420aa3e5e37fd346')
todos: Array (3)
  0: Object
    todo: "test todo"
    label: "uncategorized"
    id: "618e39ca"
    date: "8/15/2022, 1:00:00 AM"
    isDone: false
  1: Object
    todo: "Ip"
    label: "assignment"
    id: "01fb7112"
    date: "20/10/2023, 2:41:42 pm"
    isDone: false
  2: Object
    todo: "CNS"
    label: "presentation"
    id: "82bdc16d"
    date: "20/10/2023, 3:01:35 pm"
    isDone: false
labels: Array (4)
  0: Object
    label: "uncategorized"
  count: 1
```



The screenshot shows a MongoDB interface with a list of documents. The first document has the following structure:

```
isDone: false
  2: Object
    todo: "CNS"
    label: "presentation"
    id: "82bdc16d"
    date: "20/18/2023, 3:01:35 pm"
    isDone: false
  labels: Array (4)
    0: Object
      label: "uncategorized"
      count: 1
      id: "17daee68"
    1: Object
      label: "assignment"
      count: 1
      id: "1d4d1b2f"
    2: Object
      label: "presentation"
      count: 1
      id: "8922b64b"
    3: Object
      label: "test"
      count: 0
      id: "eff3ee01"
```

## Conclusion:

We displayed the use of CRUD operations in an Express App.