A Lab Manual

On

FULL STACK WEB DEVELOPMENT LAB

(III- B. Tech. – II– Semester)

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(DATA SCIENCE)

By

Ms. V.V.BHAVANI
(Asst Professor, Dept. of CSE(DS)

CMR INSTITUTE OF TECHNOLOGY

Kandlakoya(V), Medchal Road, Hyderabad – 501 401
Ph. No. 08418-222042, 22106 Fax No. 08418-222106

(2023-24)

CONTENTS

CMR INSTITUTE OF TECHNOLOGY

VISION: To create world class technocrats for societal needs

MISSION: Impart global quality technical education for a better future by providing appropriate learning environment through continuous improvement and customization

QUALITY POLICY: Strive for global excellence in academics and research to the satisfaction of students and stakeholders

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING:

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

Vision: To be a model for academic excellence and research in the field of computer science and engineering with a special focus on applications of Data Science that leads to innovative skills and moral values for the betterment of global society with professional concern.

Mission: Impart quality education through state-of-art curriculum by providing conducive learning & research environment for continuous improvement and professional advancement**.**

I. PROGRAMME EDUCATIONAL OBJECTIVES (PEO's)

PEO1: Graduate will be capable of practicing principles of computer science & engineering, mathematics and scientific investigation to solve the problems that are appropriate to the discipline.[PO's: 1,2,3,4,5,7,8,9,10,11 and 12] [PSO's: 1and 2]

PEO2: Graduate will profess in Data Science applications that lead to professional, career and research advancement. [PO's: 1,2,3,4,5,6,7,8,9,10 and 12] [PSO's: 1, 2 and 3]

PEO3: Graduate exhibits professional ethics, communication skills, teamwork and adapts to changing environments of engineering and technology by engaging in lifelong learning. [PO's: 1,2,3,4,5,6,7,8,9,10,11 and 12] [PSO's: 2 and 3]

II. PROGRAMME OUTCOMES (PO's)

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. **[PEO's: 1,2 and 3]**

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. **[PEO's: 1,2 and 3]**

3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations**. [PEO's: 1,2 and 3]**

4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. **[PEO's: 1,2 and 3]**

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. **[PEO's: 1,2 and 3]**

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. **[PEO's: 2 and 3]**

7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. **[PEO's: 1,2 and 3]**

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. **[PEO's: 1,2 and 3]**

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. **[PEO's: 1,2 and 3]**

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. **[PEO's: 1,2 and 3]**

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. **[PEO's: 1 and 3]**

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. **[PEO's: 1,2 and 3]**

2. Syllabus

FULL STACK WEB DEVELOPMENT LAB

	L	T	P	C
III-B.Tech.-II-Sem.				
Subject Code: 20-CS-PC-326	-	-	3	1.5

Course Outcomes: Upon completion of the course, the student will be able to

COs	Upon completion of course the students will be able to	PO4	PO5	PO14
CO1	illustrate implementation procedure of full stack web development	3	3	3
CO2	demonstrate HTML5, CSS5 scripting languages and Github	3	3	3
CO3	make use of scripting languages in web development	3	3	3
CO4	develop web applications using AJAX	3	3	3
CO5	build real time applications using full stack web development	3	3	3

List of Experiments

Week	Title/Experiment
1	Write code in HTML5 to develop simple webpage.
2	Write CSS & HTML5 Code to show Dropdown Menu.
3	Design Single Page Application with different menu items.
4	Write a program in CSS to show your city with building and moving cars.
5	Write a program to validate web form using javascript.
6	Write jquery code to show website slider.
7	Write a program in javascript to create a user login system.
8	Write a program in javascript to create a user registration system.
9	Write a program to display user details using HTML, CSS & AJAX.
10	Demonstrate version control in Git and Github.

Micro-Projects: Student must submit a report on one of the following Micro–Projects before commencement of second internal examination.

1. Develop Project MyNote - A HTML5 App
2. Develop a Bookstore application by using HTML5, CSS, jquery in Github
3. Develop a shopping cart application by using HTML5, CSS, jquery in Github
4. Develop an e-learning system using HTML5, CSS, jquery in Github
5. Build a personal portfolio webpage using HTML5, CSS, jquery.
6. Develop google.com Search result page using HTML5, CSS, jquery & Ajax
7. Develop a webpage to display solar system using HTML5, CSS, jquery & Ajax
8. Build Tajmahal using CSS.
9. Build a Real-Time Markdown Editor with Node.js
10. Develop an User model covering, Registration, Email verification(send an email), Login (with remember me)

Reference: 1. Full Stack Web Development Lab Manual, Department of CSE, CMRIT, Hyd.

3. Student Entry Behavior or Pre-requisites

1. Students should have basic knowledge on HTML and CSS
2. Students should have basic knowledge on C and Java programming.
3. Student should have knowledge on oops and software engineering concepts

These prerequisites are taken by the students during the first two years. However during the initial sessions the topics are reviewed.

4. Course Outcomes

Course Outcome	Course Outcome Statements
CO - 1	illustrate implementation procedure of full stack web development
CO – 2	demonstrate HTML5, CSS5 scripting languages and Github
CO – 3	make use of scripting languages in web development
CO – 4	develop web applications using AJAX
CO – 5	build real time applications using full stack web development

5. Mapping of Course with PEOs-POs

(Only Ticking)

Program Educational Objectives (PEOs)

Sl. No.	PEOs Name	Program Education Objective Statements
1	PEO - 1	Graduate will be capable of practicing principles of computer science & engineering, mathematics and scientific investigation to solve the problems that are appropriate to the discipline.[PO's: 1,2,3,4,5,7,8,9,10,11 and 12] [PSO's: 1 and 2]
2	PEO – 2	Graduate will profess in Data Science applications that lead to professional, career and research advancement. [PO's: 1,2,3,4,5,6,7,8,9,10 and 12] [PSO's: 1, 2 and 3]
3	PEO – 3	Graduate exhibits professional ethics, communication skills, teamwork and adapts to changing environments of engineering and technology by engaging in lifelong learning. [PO's: 1,2,3,4,5,6,7,8,9,10,11 and 12] [PSO's: 2 and 3]

Program Outcomes (POs)

PO Name	Graduate Attributes	PO Statements
PO1	Engineering knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. [PEO's: 1,2 and 3] [PSO's: 1,2 and 3]
PO 2	Problem analysis	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. [PEO's: 1,2 and 3] [PSO's: 1,2 and 3]
PO 3	Design/ development of solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. [PEO's: 1,2 and 3] [PSO's: 1,2 and 3]
PO 4	Conduct investigations of complex problems	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. [PEO's: 1,2 and 3] [PSO's: 1,2 and 3]
PO 5	Modern tool usage	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. [PEO's: 1,2 and 3] [PSO's: 1,2 and 3]

PO 6	The engineer and society	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. [PEO's: 2 and 3]
PO 7	Environment and sustainability	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. [PEO's: 1,2 and 3]
PO 8	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. [PEO's: 1,2 and 3] [PSO's: 2 and 3]
PO 9	Individual and team work	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. [PEO's: 1,2 and 3] [PSO's: 3]
PO 10	Communication	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. [PEO's: 1,2 and 3] [PSO's: 2 and 3]
PO 11	Project management and finance	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. [PEO's: 1 and 3] [PSO's: 2 and 3]
PO 12	Life-long learning	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. [PEO's: 1,2 and 3] [PSO's: 1,2 and 3]

6. Mapping Of Course Outcomes With POs

No	Course Outcomes	Po$_1$	Po$_2$	Po$_3$	Po$_4$	Po$_5$	Po$_6$	Po$_7$	Po$_8$	Po$_9$	Po$_{10}$	Po$_{11}$	Po$_{12}$	Avg
1	CO - 1				3	3	3		3					
2	CO – 2				3	3	3		3					
3	CO – 3				3	3	3		3					
4	CO – 4				3	3	3		3					
5	CO – 5				3	3	3		3					
	Avg				3	3	3		3					

7. Direct Course Assessment

(As mentioned in following table of 10 parameters, of which consider only the parameters required for this courses)

No	Description	Targeted Performance	Actual Performance	Remarks	Course Attainment
1	Internal Marks(25)	80% of Students(182 Students) should Secure 60% of Internal Marks i.e., 15 Marks			
2	External Marks(50)	80% of Students(182 Students) should Secure 70% of External Marks i.e., 35 Marks			
3	Clearing of Subject	A minimum of 95% of Students(216 Students) should clear this course in first attempt			
4	Getting First Class	90% of Students(205 Students) should Secure I Class Marks i.e., 45 Marks in my course			
5	Distinction	80% of Students (182 Students) should secure First Class With Distinction i.e., 53 Marks in my course			
6	Outstanding Performance	60% of Students (137 Students) should secure 80% and above Marks. i.e., 60 Marks in my course			

8. Indirect Course Assessment

(As mentioned-strong (3), moderate (2), weak (1) & no comment (0))

Mission Statement of CSE(DS)

- **Impart fundamentals through state of art technologies for research and career in Computer Science & Engineering.**
- **Create value-based, socially committed professionals for anticipating and satisfying fast changing societal requirements.**
- **Foster continuous self learning abilities through regular interaction with various stake holders for holistic development.**

Correlation of Mission Elements with Mission Statement of CSE(Data Science) Department related to the Course (only Ticking given by faculty)

No	Mission Elements	Strong	Moderate	Weak	No Comment
M-1	Impart Fundamentals	√			
M-2	State Of Art Technologies	√			
M-3	Research & Career Development	√			
M-4	Value based Socially Committed Professional	√			
M-5	Anticipating & Satisfying Industry Trends		√		
M-6	Changing Societal Requirements			√	
M-7	Foster Continuous Learning	√			
M-8	Self Learning Abilities	√			
M-9	Interaction with stakeholders	√			
M-10	Holistic Development		√		

Indirect Course Assessment through Student Satisfaction Survey

(Note for *: Parameters used for course teaching like
a: Classroom teaching b: Simulations c:labs d: Mini_Projects
e: Major Projects f: Conferences g: professional activities
h: Technical Clubs i: Guest Lectures j: Workshops k: Technical Fests l:Tutorials
m:NPTLs n:Digital Library o: Industrial Visits p: software Tools
q: Internship/training r:Technical Seminars
s: NSS t: NSS u: sports etc.

No	Question Based on PEO/ PO/PSO/CO	Parameters (a /b /c…/)*	Strong (3)	Moderate (2)	Weak (1)	No comment (0)
1	Did the course impart fundamentals through interactive learning and contribute to core competence?					
2	Did the course provide the required knowledge to foster continuous learning?					
3	Whether the syllabus content anticipates & satisfies the industry and societal needs?					
4	Whether the course focuses on value based education to be a socially committed professional?					
5	Rate the role of the facilitator in mentoring and promoting the self learning abilities to excel academically and professionally?					
6	Rate the methodology adopted and techniques used in teaching learning processes?					
7	Rate the course in applying sciences & engineering fundamentals in providing research based conclusions with the help of modern tools?					
8	Did the course have any scope to design, develop and test a system or component?					
9	Rate the scope of this course in addressing cultural, legal, health, environment and safety issues?					
10	Scope of applying management fundamentals to demonstrate effective technical project presentations & report writing?					
	Total					
	Average					
Total Average			2.52			

9. Overall Course Assessment

(80% Direct + 20% Indirect, if any)

No	Assessment Type	Weightage	Attainment Level
1	Direct-Assignment, Quiz, Subjective, University Exams, Results, Bench Marks	0.8	
2	Indirect-Surveys-Questionnaire	0.2	
	Overall		

FSWD LAB Course Attainment level:

10. Pi diagrams, Bar charts, Histograms

(For representing previous results, if any)

FSWD Pass % for Last 4 Academic Years	Appeared	Passed	Pass%

11. Lesson/Course Plan

Week No.	Name of the Program	Week	Text Books	**Mode of Assessment**
1	Write code in HTML5 to develop simple webpage	1	R1	By observations, lab records, viva-voice
2	Write CSS &amp; HTML5 Code to show Dropdown Menu.	2	R4	By observations, lab records, viva
3	Design Single Page Application with different menu items	3	R4	By observations, lab records, viva
4	Write a program in CSS to show your city with building and moving cars.	4	R4	By observations, lab records, viva
5	Write a program to validate web form using javascript	5	R4	By observations, lab records, viva
6	Write jquery code to show website slider	6	R2	By observations, lab records, viva
7	Write a program in javascript to create a user login system.	7	R2	By observations, lab records, viva
8	Write a program in javascript to create a user registration system.	8	R1	By observations, lab records, viva
9	Write a program to display user details using HTML, CSS & AJAX.	9	R1	By observations, lab records, viva
10	Demonstrate version control in Git and Github	10	R1	By observations ,lab records, viva

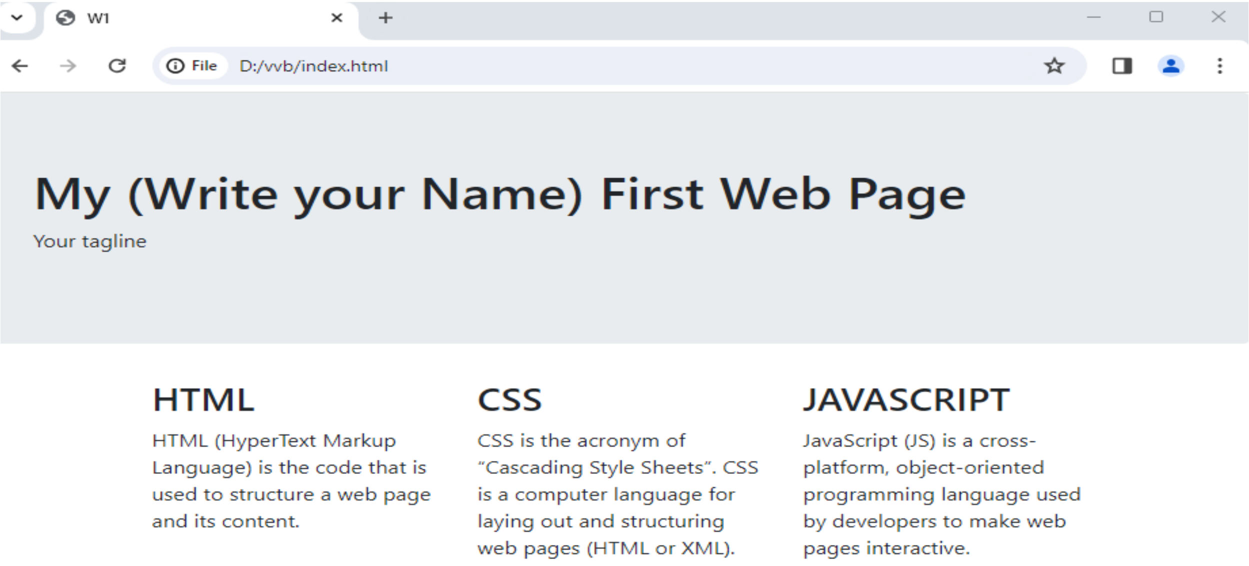
EXPERIMENT: 1. Write code in HTML5 to develop simple webpage.

SOURCE CODE:

```
 <!DOCTYPE html>
<html lang="en">
<head>
 <title>W1</title>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">
 <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.1/dist/jquery.slim.min.js"></script>
 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
 <div class="jumbotron text-left">
  <h1>My (Write your Name) First Web Page</h1>
  <p>Your tagline</p>
 </div>
 <div class="container">
  <div class="row">
   <div class="col-sm-4">
    <h3>HTML</h3>
    <p>HTML (HyperText Markup Language) is the code that is used to structure a web page and
its content. </p>
   </div>
```

```
      <div class="col-sm-4">
        <h3>CSS</h3>
       <p>CSS is the acronym of “Cascading Style Sheets”. CSS is a computer language for laying out and structuring web pages (HTML or XML). </p>
      </div>
      <div class="col-sm-4">
        <h3>JAVASCRIPT</h3>
        <p>JavaScript (JS) is a cross-platform, object-oriented programming language used by developers to make web pages interactive. </p>
      </div>
    </div>
  </div>
</body>
</html>
```

OUTPUT:



My (Write your Name) First Web Page

Your tagline

HTML

HTML (HyperText Markup Language) is the code that is used to structure a web page and its content.

CSS

CSS is the acronym of “Cascading Style Sheets”. CSS is a computer language for laying out and structuring web pages (HTML or XML).

JAVASCRIPT

JavaScript (JS) is a cross-platform, object-oriented programming language used by developers to make web pages interactive.

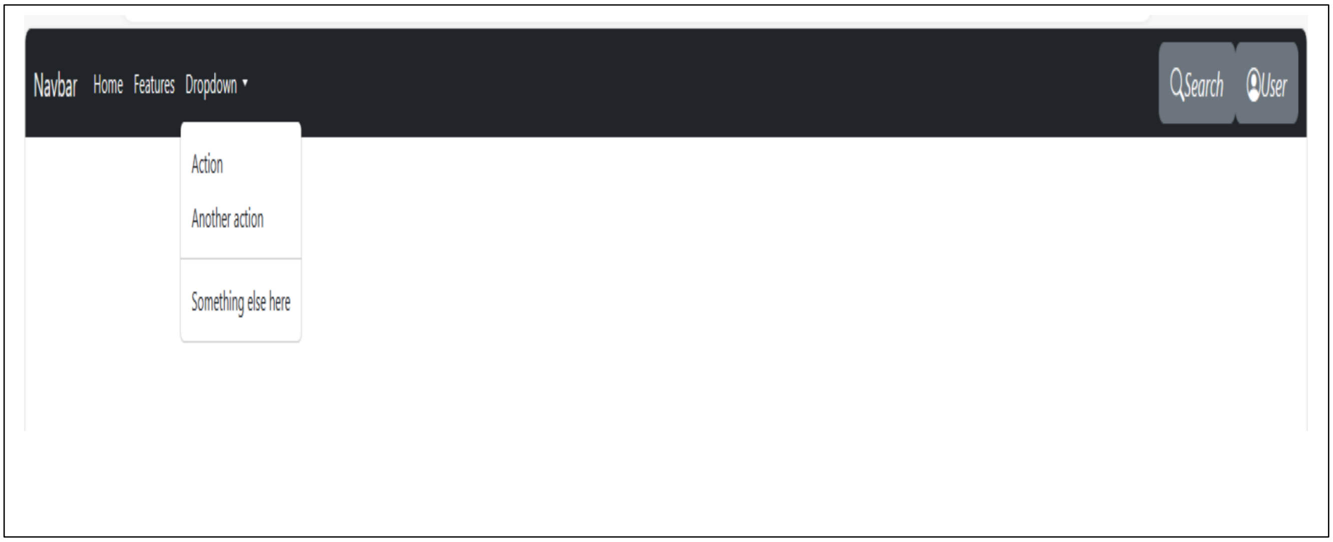
EXPERIMENT 2: Write CSS5 & HTML5 Code to show Dropdown Menu.

SOURCE CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Navbar with Dropdown Example</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.3/font/bootstrap-
icons.css">
  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.min.js"></script>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"
>
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" href="#">Home</a>
        </li>
        <li class="nav-item ">
          <a class="nav-link active" href="#">Features</a>
        </li>
        <li class="nav-item dropdown ">
          <a class="nav-link active dropdown-toggle" href="#" id="navbarDropdown" role="button" data-
bs-toggle="dropdown" aria-expanded="false">
            Dropdown
          </a>
          <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="#">Action</a></li>
            <li><a class="dropdown-item" href="#">Another action</a></li>
            <li><hr class="dropdown-divider"></li>
            <li><a class="dropdown-item" href="#">Something else here</a></li>
          </ul>
        </li>
      </ul>

    </div>
    <div class="nav navbar-nav ">
        <a href="#" class="btn btn-secondary btn-lg"><i class="bi bi-search">Search</i></a>
        <a href="#" class="btn btn-secondary btn-lg"><i class="bi bi-person-circle">User</i></a>
    </div>
  </div>
</nav>
</body>
</html>
```

OUTPUT:

-



EXPERIMENT 3: Design Single Page Application with different menu items.

SOURCE CODE:

- npm install -g @angular/cli
- ng new my-app
- cd my-app
- ng serve –open

Opens your browser to http://localhost:4200/

app.component.html

```
<app-navbar></app-navbar>
<router-outlet></router-outlet>
```

home.component.html

```
<p>home works!</p>
<h1>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Rerum, ad? Neque ducimus
repellendus enim veniam vel magnam perspiciatis fugit nesciunt?</h1>
<h2>Lorem ipsum dolor sit amet consectetur adipisicing elit. In, aperiam.</h2>
```

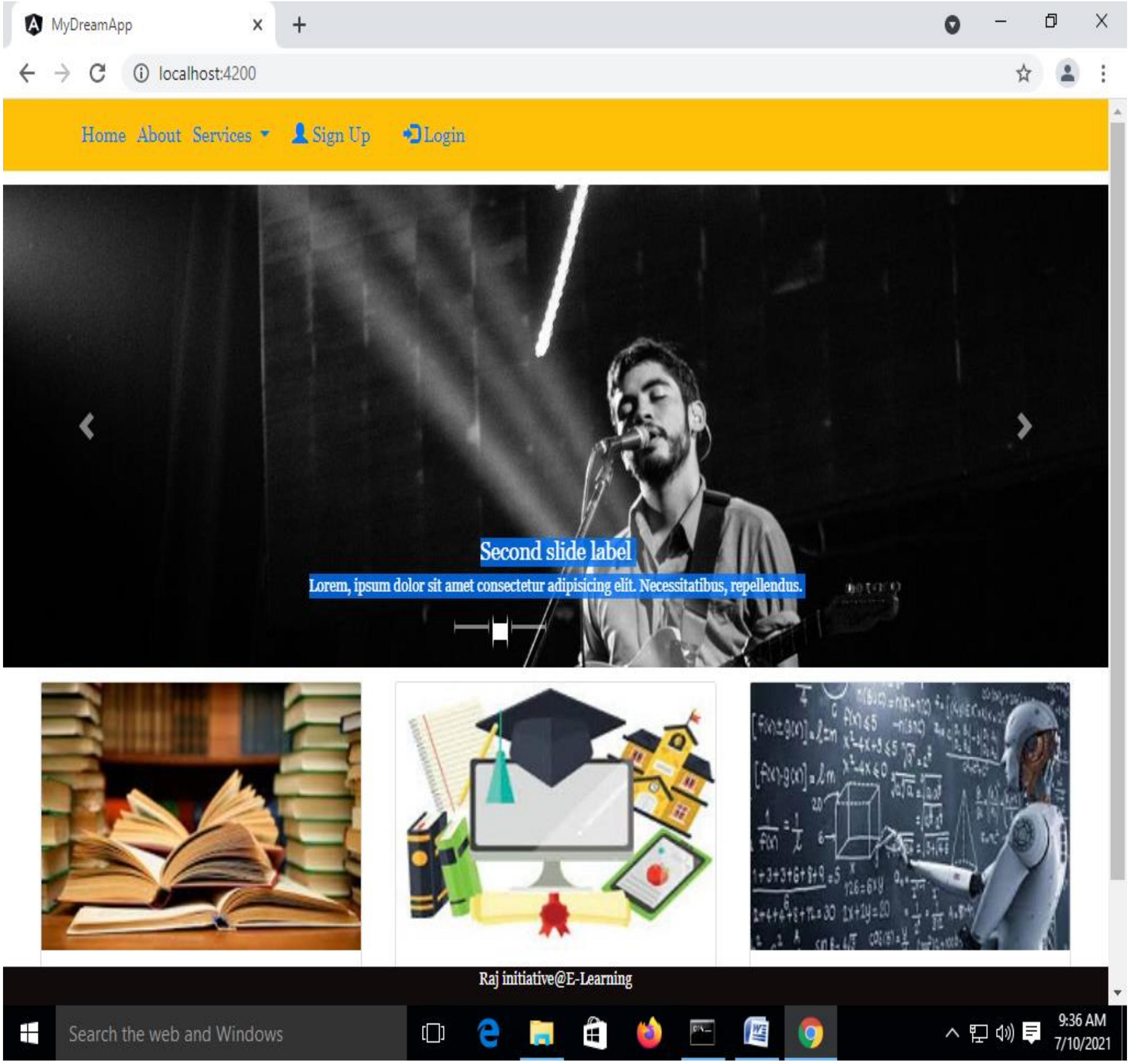
about.component.html

```
<h1>about works!</h1>
<p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Consectetur ab natus aperiam, saepe
accusantium voluptas dolo</p>
```

app-routing.module.ts

```
const routes: Routes = [
    { path: 'Home', component: HomeComponent },
    { path: 'About', component: AboutComponent },
];
```

OUTPUT:



Second slide label

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Necessitatibus, repellendus.

Raj initiative@E-Learning

EXPERIMENT 4: Write a program in CSS to show your city with building and moving cars.

SOURCE CODE:

Index.html

```
<!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>Moving Cars</title>
    <link rel="stylesheet" href="style.css">
</head>

<body>
    <div class="container">
        <div class="road"></div>
        <div class="road-sideview"></div>
        <div class="moving-car">
            <img src="bmw.png" alt="moving-car">
        </div>

        <div class="car-wheel">
            <img src="wheel.png" alt="moving car wheel" class="car-back-wheel">

            <img src="wheel.png" alt="moving car wheel" class="car-front-wheel">
        </div>
    </div>
</body>
</html>
```

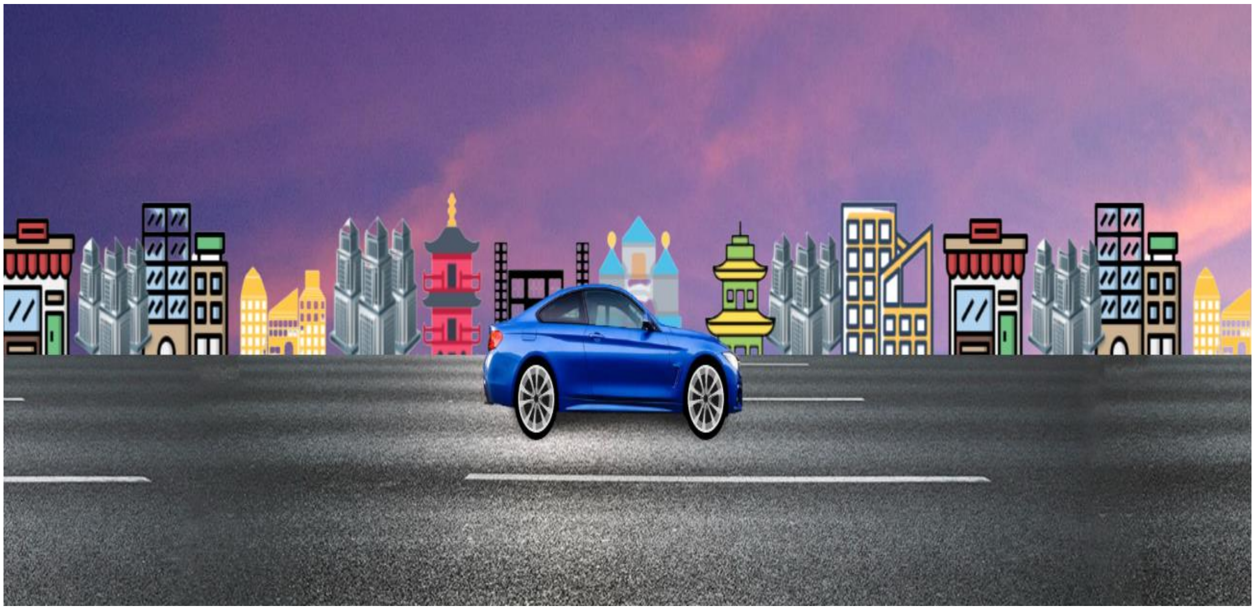
style.css

```
* {
    margin: 0;
    padding: 0;
}
.container {
    height: 50vh;
    width: 100%;
    background-image: url(sky1.jpg);
    background-size: cover;
    background-position: center;
```

```
  position: relative;
  overflow-x: hidden;
}
.road {
  height: 200px;
  width: 300%;
  display: block;
  background-image: url(road.jpg);
  position: absolute;
  bottom: 0;
  left: 0;
  right: 0;
  z-index: 1;
  background-repeat: repeat-x;
  animation: road 5s linear infinite;
}
@keyframes road {
  100% {
    transform: translateX(-3400px)
  }
}
.road-sideview {
  height: 130px;
  width: 1000%;
  background-image: url(city.png);
  position: absolute;
  bottom: 200px;
  left: 0;
  right: 0;
  display: block;
  z-index: 1;
  background-repeat: repeat-x;
  animation: road-sideview 5s linear infinite;
}

@keyframes road-sideview {
  100% {
    transform: translateX(-1400px);
  }
}
.moving-car {
  width: 500px;
  left: 50%;
  bottom: 50px;
  transform: translateX(-50%);
  position: absolute;
  z-index: 2;
```

```
}
.moving-car img {
   width: 90%;
   animation: moving-car 1s linear infinite;
}
@keyframes moving-car {
   100% {
      transform: translateY(-1px);
   }
   50% {
      transform: translateY(1px);
   }
   0% {
      transform: translateY(-1px);
   }
}
.car-wheel {
   left: 50%;
   bottom: 198px;
   transform: translateX(-50%);
   position: absolute;
   z-index: 2;
}
.car-wheel img {
   width: 65px;
   height: 65px;
   animation: car-wheel .5s linear infinite;
}
@keyframes car-wheel {
   100% {
      transform: rotate(360deg);
   }
}
.car-back-wheel {
   left: -170px;
   position: absolute;
}
.car-front-wheel {
   left: 80px;
   position: absolute;
}
```

OUTPUT:



EXPERIMENT 5: Write a program to validate web form using JavaScript.

Source Code:

Validation.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Register</title>
    <style>
        body {
            text-align: center;
            font-family: sans-serif;
        }
        h1 {
            font-size: 20px;
        }
        table tr td {
            padding-top: 6px;
            padding-bottom: 6px;
        }
        fieldset {
            width: 500px;
            text-align: center
        }
    </style>

<script>
```

```
function validate() {

    var fn = frm.fname.value;
    for (x in fn) {
        ch = fn.charCodeAt(x);
        if (ch < 65 || ch > 90 && ch < 97 || ch > 122) {
            alert("Invalid firstname");
            return false;
        }
    }
    var ln = frm.lname.value;
    for (y in ln) {
        ch = ln.charCodeAt(y);
        if (ch < 65 || ch > 90 && ch < 97 || ch > 122) {
            alert("Invalid lastname");
            return false;
        }
    }
    var phn = frm.phone.value;
    var lenp = phn.length;
    if (lenp !== 10) {
        alert("Phone no should be exactly 10 digits");
        return false;
    }

    var pwd1 = frm.pwd.value;
```

```
        var pwdl = pwd1.length;
        if (pwdl % 2 === 1) {
            alert("Password should contain even number of characters");
            return false;
        }
        if (pwdl > 8) {
            alert("Password should not exceed 8 digits");
            return false;
        }

        var reg = /^\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*$/;
        var mail = frm.mailid.value;
        if (reg.test(mail)) {
            alert("Valid email");
        }
        else {
            alert("Invalid email");
            return false;
        }
        return true;
    }
</script>
    <body>
        <div id="container"></div>
        <div id="header">
            <h1>Registration</h1>
```

```
</div>
<div id="content">
 <center>
 <form name="frm" method="POST" action="success.html" onSubmit="return
validate()">
<fieldset align="center">
<table align="center">
   <tr>
 <td>First Name: </td>
<td><input type="text" name="fname" value="" size="50" required /></td>
</tr>
 <tr>
 <td>Last Name: </td>
 <td> <input type="text" name="lname" value="" size="50" required /></td>
 </tr>
 <tr>
<td>Phone No: </td>
 <td> <input type="text" name="phone" value="" size="50" required /></td>
 </tr>
 <tr>
<td>Mail id:</td>
 <td><input type="email" name="mailid" value="" size="50" required /></td>
 </tr>
 <tr>
 <td>Gender:</td>
<td>Male: <input type="radio" name="gender" value="male">
```

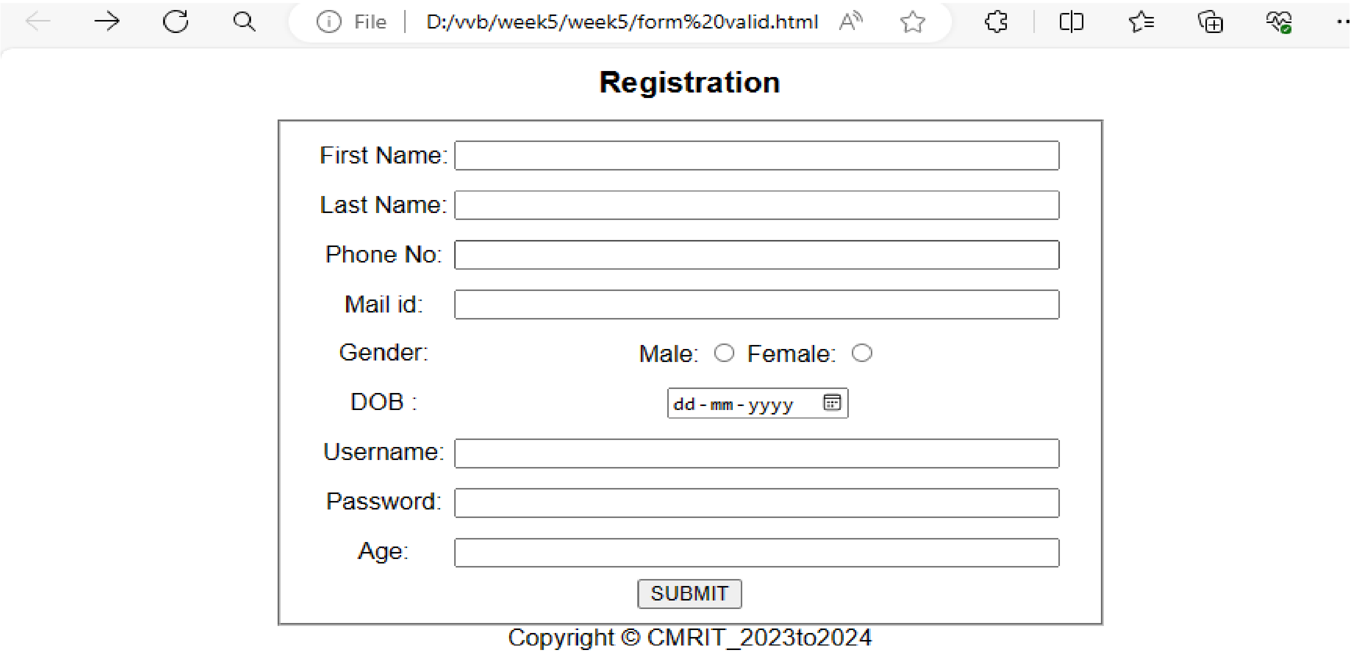
```
 Female: <input type="radio" name="gender" value="female"></td>
 </tr>
 <tr>
 <td>DOB :</td>
 <td><input type="date" name="dob" size="50" required /></td>
</tr>
 <tr>
 <td>Username:</td>
<td><input type="text" name="uname" value="" size="50" required /></td>
 </tr>
<tr>
<td>Password:</td>
 <td><input type="password" name="pwd" value="" size="50" required /></td>
</tr>
<tr>
<td>Age:</td>
<td><input type="text" name="age" value="" size="50" required /></td>
</tr>
</table>
<input type="submit" value="SUBMIT" name="submit" />
 </fieldset>
</form>
</div>
</center>
</body>
<div id="footer">
```

```
    Copyright © CMRIT_2023to2024
</div>
</div>
</html>
```

Success.html

```
<!DOCTYPE html>
<html>
<body>
    <h1>Registration completed</h1>
</body>
</html>
```

Output:



Registration

First Name:

Last Name:

Phone No:

Mail id:

Gender: Male: ○ Female: ○

DOB : dd - mm - yyyy

Username:

Password:

Age:

SUBMIT

Copyright © CMRIT_2023to2024

EXPERIMENT 6: Write jquery code to show website slider.

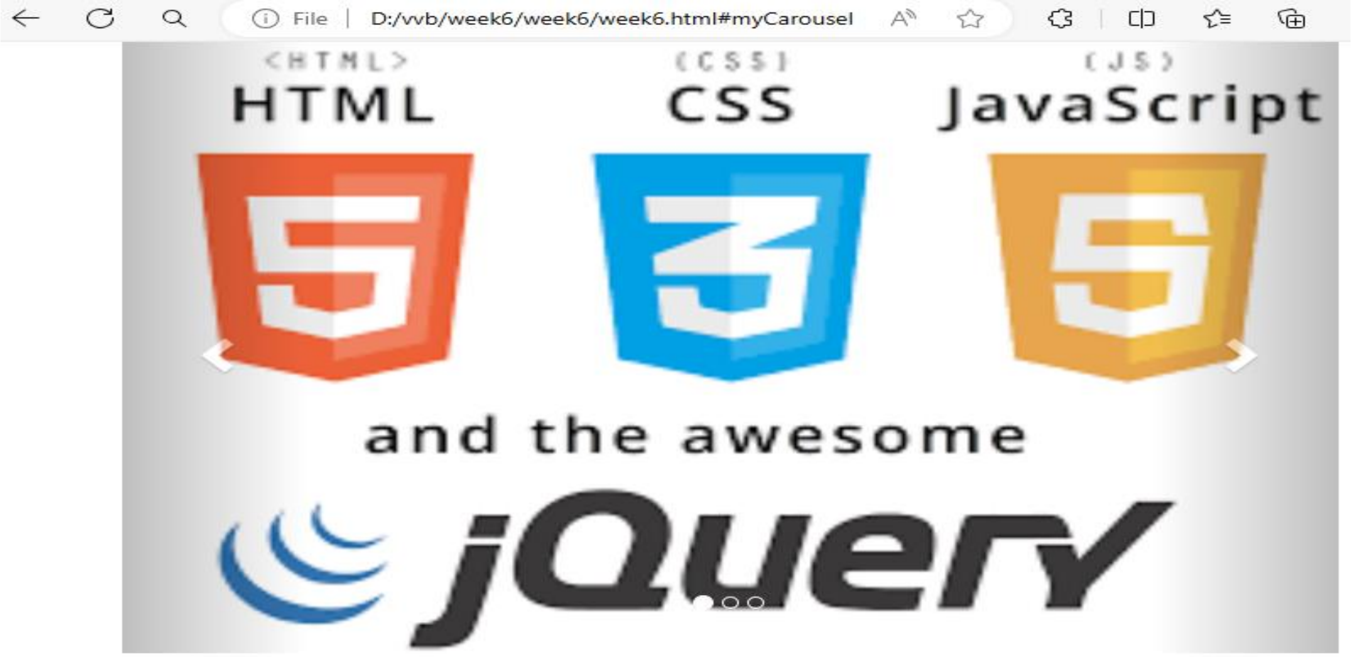
Source Code

Corousel.html:

```
<!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>JS Slider</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<div id="myCarousel" class="carousel slide">

<ol class="carousel-indicators">

    <!--The carousel-indicators list provides small circles or dots at the bottom of the carousel to indicate the currently active slide-->

<li class="item1 active"></li>

<li class="item2"></li>

<li class="item3"></li>

</ol>

<div class="carousel-inner" role="listbox">

<div class="item active">
```

```
<img src="5.jpg" alt="image1" width="50%" height="50%">
</div>
<div class="item">
<img src="6.jpg" alt="image2" width="100%" height="100%">
</div>
<div class="item">
<img src="7.jpg" alt="image3" width="100%" height="100%">
</div>
</div>
<a class="left carousel-control" href="#myCarousel" role="button">
<span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
<span class="sr-only">Prev</span>
</a>
<a class="right carousel-control" href="#myCarousel" role="button">
<span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
<span class="sr-only">Next</span>
</a>
</div>
</div>
<script>
$(document).ready(function(){
$("#myCarousel").carousel();
$(".item1").click(function(){
$("#myCarousel").carousel(0);
})
$(".item2").click(function(){

$("#myCarousel").carousel(1);

})

$(".item3").click(function(){

$("#myCarousel").carousel(2);

})

$(".left").click(function(){

$("#myCarousel").carousel("prev");

})

$(".right").click(function(){

$("#myCarousel").carousel("next");

})});

</script>

</body>

</html>
```

OUTPUT:



EXPERIMENT 7: Write a program in javascript to create a user login system.

Source Code:

Login.pug:

```
doctypehtml
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 Login
style
include ./my.css
body
div(class='container')
include ./index.pug
h1 Login form
br
form(action="/Login" method="post" align="center")
label(for="username") username
input(type="text" name="username")
br
br
label(for="password") password
input(type="password" name="password")
br
br
input(type="submit" name="Login" value="Login")
```

Style.css

```
            h1{
color:blue;
text-align:center;
}
a:link,a:visited{
background-color: brown;
color:white;
padding:14px 25px;
text-align: center;
display: inline-block;
}
a:hover,a:active{
background-color: chartreuse;
}
.header {
padding: 10px;
text-align: center;
background: #1abc9c;
color: white;
font-size: 30px;
}
.container {
border-radius: 5px;
background-color: #f2f2f2;
padding: 10px;
}
input{
width: 20%;
padding: 12px;
border: 1px solid #ccc;
margin-top: 6px;
margin-bottom: 16px;
resize: vertical;
}
input[type=submit] {
background-color: #04AA6D;
color: white;
padding: 12px 20px;
border: none;
cursor: pointer;
}
input[type=submit]:hover {
background-color: #45a049;
}
label{
color: blue;
font-size: 22px;
padding: 8px;
text-align:left;
}
```

App.js

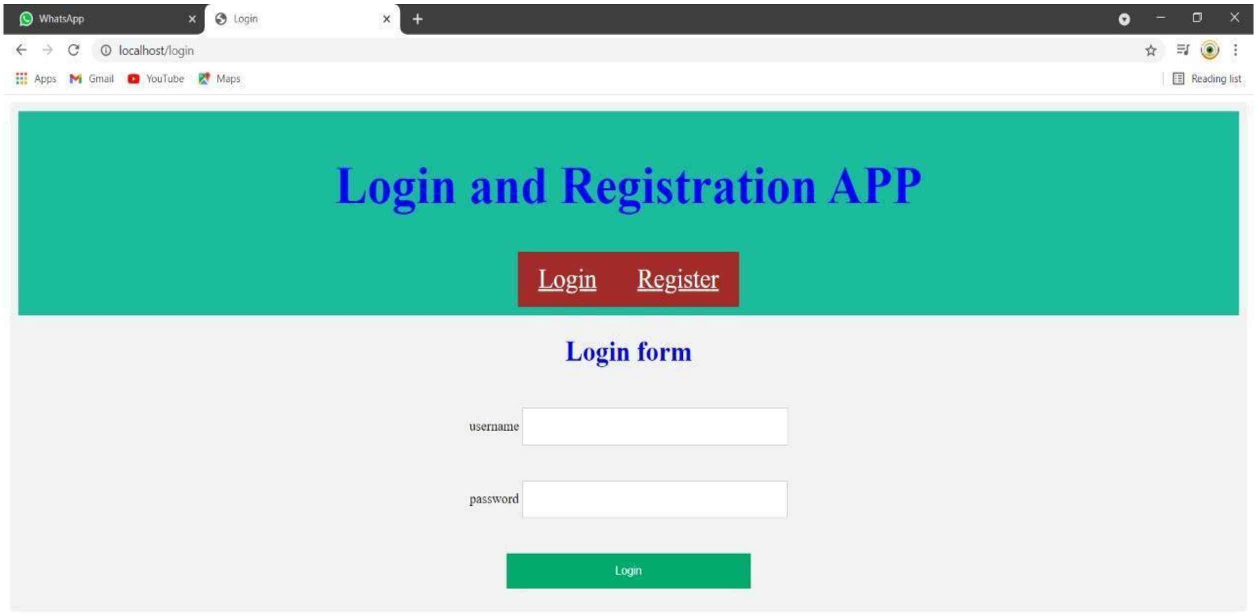
```
Const  express=require('express');
const bodyparser=require("body-parser")
const bcrypt=require("bcrypt");
const user=require('./models/user');
const mongoose = require('mongoose');
const expressValidator = require("express-validator");
const {check, validationResult} = require('express-validator/check')
const app = express();
const port = process.env.PORT || 80
mongoose.connect("mongodb://localhost:27017/user",{userNewUrlP
arser : true});
```

```
app.set('view engine', 'pug');
app.use(bodyparser.json());
app.use(bodyparser.urlencoded({extended:true}));
//handling get request
app.get('/',function(req,res){
res.render('index')
})
app.get('/Login',function(req,res){
res.render('Login')
})
//handling post request
app.post('/Login',function(req,res){

user.findOne({username:req.body.username},function(err,docs){
if(err)
{
console.log(err)
}
else

{
if(docs.username==req.body.username)
{
bcrypt.compare(req.body.password,docs.password,function(err,data)
{
if(err)
{
console.log(err);
}
if(data)
{
console.log(data);
res.send("Welcome");
}
else
{
res.send("invalid password");
}
});

}
else
{
//res.send("invalid username or password")
res.redirect("Register");
}
}
})
})
app.listen(port,() => {console.log(`app is listening on http://localhost:${port}`)})
```

Models/user.js

```
 const  mongoose=require('mongoose');
  const Schema=mongoose.Schema;

const userSchema=new Schema(
{
username : {type:String},
password : {type:String},
age : {type:Number},
mobile : {type:Number}
}
);
module.exports=mongoose.model("user",userSchema);
```

OUTPUT:



Login and Registration APP

Login form

EXPERIMENT 8: Write a program in javascript to create a user registration system.

Source Code:

Register.pug

```
doctypehtml
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 Register
style
include ./my.css
body
div(class='container')
include ./index.pug
h1 Registration form
ul(id="errors")
Br
form(action="/Register" method="post" align="center")
label(for="username") username
input(type="text" name="username")
Br
Br
label(for="password") password
input(type="password" name="password")
Br
label(for="cpassword") Confirm password
input(type="password" name="cpassword")
Br
Br
label(for="age") user age
input(type="text" name="age")
Br
Br
label(for="mobile") user mobile
input(type="text" name="mobile")
Br
         label(for="email") user email
       input(type="text" name="email")
          Br
         Br
input(type="submit" name="Register" value="Register")
```

App.js

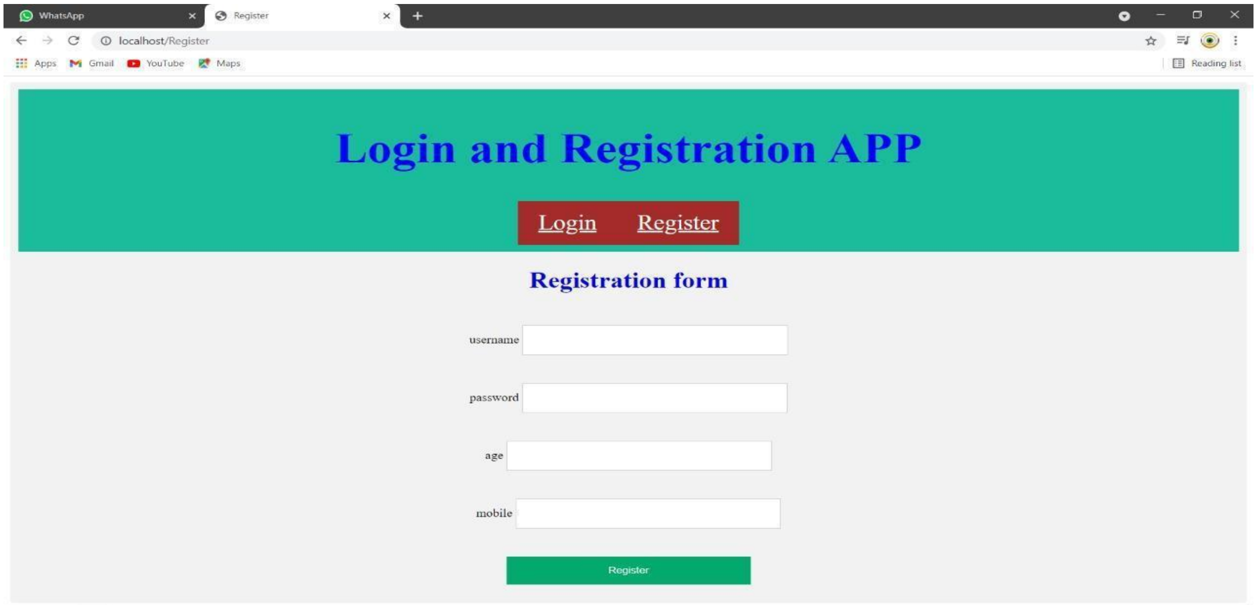
```
Const express=require('express’);
const bodyparser=require("body-parser") const bcrypt=require("bcrypt");
const user=require('./models/user'); const mongoose = require('mongoose');
const expressValidator = require("express-validator");
const {check, validationResult} = require('express-validator/check') const app = express();
Const port = process.env.PORT || 80
mongoose.connect("mongodb://localhost:27017/user",{userNewUrlParser : true}); app.set('view engine',
'pug')

app.use(bodyparser.json()); app.use(bodyparser.urlencoded({extended:true}));
//handling get request app.get('/',function(req,res){ res.render('index')
})
app.get('/Register',  function(req, res){
res.render('Register')
})
//handling post request
app.post('/Register', [
check('username').not().isEmpty().isLength({min:5}).withMessage('User  name  must  be  5  characters'),
check('password').not().isEmpty().isLength({min:6}).withMessage('Password name must be 6
characters'),
check('mobile').not().isEmpty().isInt().isLength({min:10}).withMessage('mobile number must be
number and 10 characters'),
check('cpassword').custom((value,{req})  =>  (value  ===  req.body.password)).withMessage("Confirm
password not match with your password"),
check('email').not().isEmpty().isEmail().normalizeEmail().withMessage("Enetr proper email"),
],
function(req,res){
const errors= validationResult(req);
 if(!errors.isEmpty())
{
return res.status(422).jsonp(errors.array());
}
else{
//console.log(req.body.username)
const newUser=new user();
newUser.username=req.body.username;
var salt=bcrypt.genSaltSync(10);
varhash=bcrypt.hashSync(req.body.password,salt);
newUser.password=hash;
newUser.age=req.body.age;
 newUser.mobile=req.body.mobile;
 newUser.save(function(err,result){
if(err){
console.log(err);
}
else{
```

```
console.log(result);
res.redirect("Login");

}

})

}
})

})

app. listen(port,() => {console.log(`app is listening on http://localhost:${port}`)})
```

Model/users.js

```
Const mongoose=require('mongoose');
const Schema=mongoose. Schema;
const user Schema=new Schema(
{
username : {type: String},
password : {type: String},
age : {type: Number},
mobile : {type: Number}
}
);
module.exports=mongoose. model("user",userSchema);
```

OUTPUT:



Login and Registration APP

Registration form

EXPERIMENT 9: Write a program to display user details using HTML, CSS &AJAX.

Source Code:

Ajax.pug:

```
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 Login
      style
         include./my.css
  script(src="http://code.jquery.com/jquery-3.1.0.min.js")
  script(src="/magic.js")
         include ./index.pug
  body
      div(class='container')
   form(method="post" id="change" align="center") input(type='text', placeholder='search user',
      name='name') input(type="submit", value="Search")
      br
      h1  User  Details  p
      !{name}
```

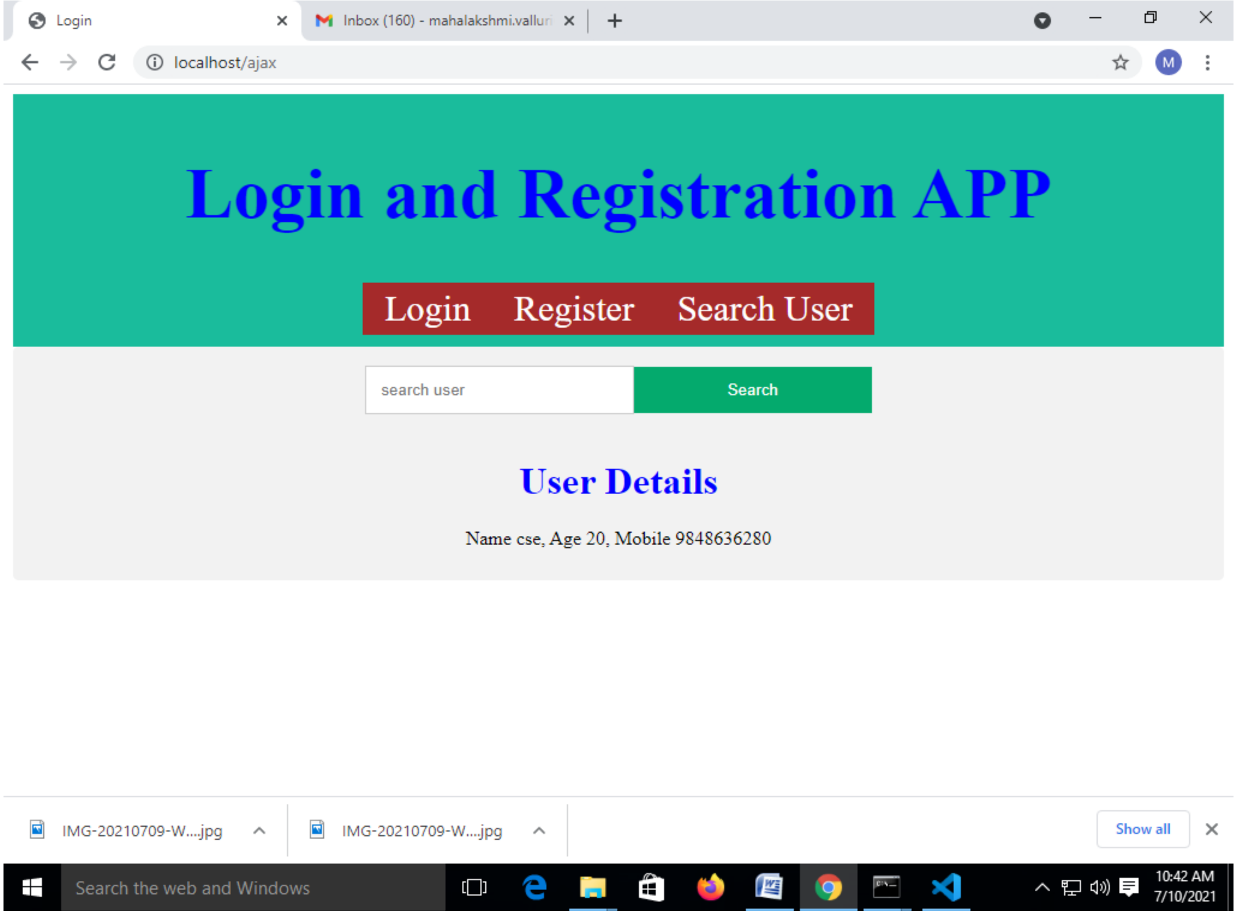
/public/magic.js

```
      $(document).ready(function(){
    $("form#change").on('submit',function(e)
   {
   e.preventDefault();
      var data = $('input[name=name]').val();
      $.ajax({
      type: 'post',
      url: '/ajax',
      data:data,
          dataType: 'text'
          })
          .done(function(data){
          $('h1').html(data.name);
          });
          });
          });
```

App.js

```
    const express = require('express');
    const bodyparser=require("body-parser")
    const bcrypt=require("bcrypt");
    const
    user=require('./models/user');
    const
    mongoose=require('mongoose');
    const expressValidator = require("express-validator");
    const {check, validationResult} = require('express-validator/check')
    const app = express();
    const port = process.env.PORT || 80
    mongoose.connect("mongodb://localhost:27017/user",{userNewUrlParser : true});
    app.set('view engine', 'pug');
     app.use(bodyparser.json());
        app.use(bodyparser.urlencoded({exte
                nded:true});
    app.get('/ajax', function(req, res){
    res.render('ajax', {title: 'An Ajax Search', name: "Search user!"});
    });


    app.post('/ajax', function(req, res){
  user.findOne({username:req.body.name},function(err,docs)
  {
  if(err)
    {
    console.log(err)


    }
    else
    {


                res.render('ajax', {title: 'An Ajax search', name: "Name "+docs.username+", Age
    "+docs.age+", Mobile "+docs.mobile });


       }


          });

       });


app.listen(port,() => {console.log(`app is listening on http://localhost:${port}`)})
```

OUTPUT:



Login and Registration APP

User Details

Name cse, Age 20, Mobile 9848636280

EXPERIMENT 10: Demonstrate version control in Git and Github.

Source Code:

```
$ git config --global user.name "cmr"
$ git config --global user.email cmr@example.com
```

Index.html

```
<h1>welcome to my web page</h1>
```

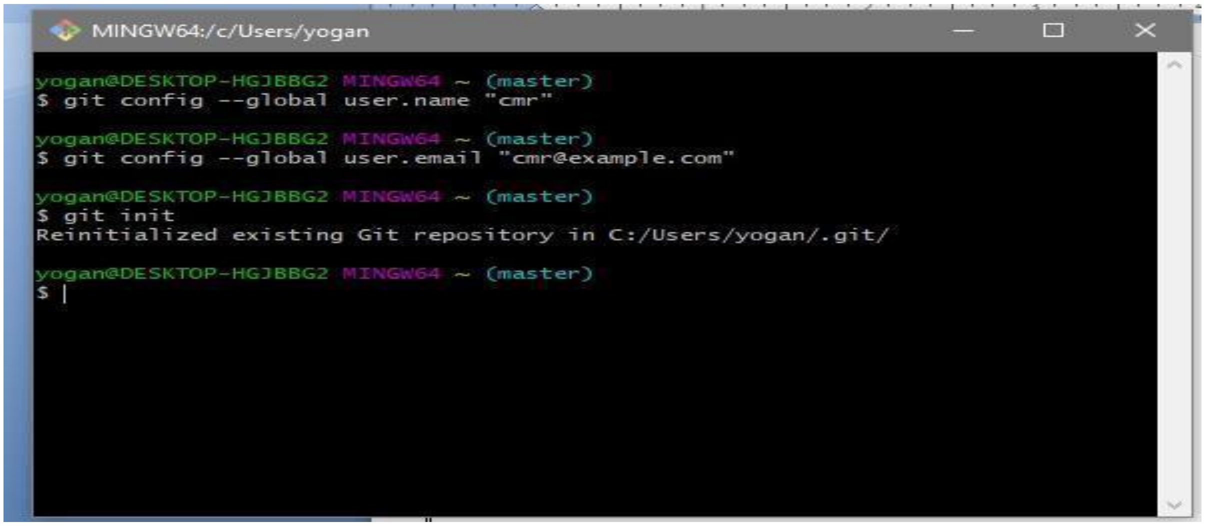
```
git init git add .
git commit -m "Hello world"
```

Index.html

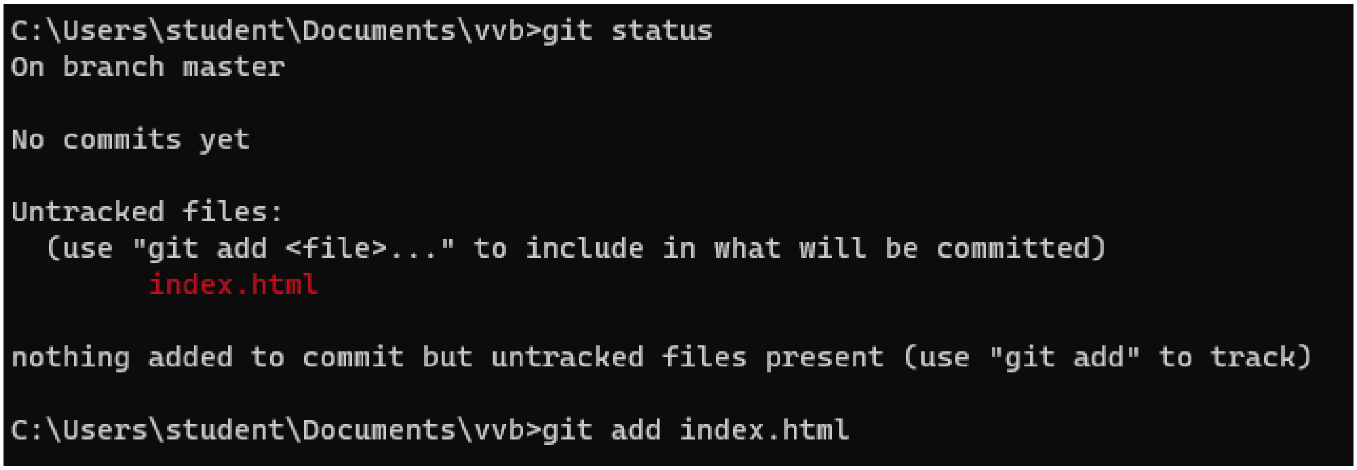
```
<h1>welcome to my web page</h1>
<p>My first website</p>
```

```
git add .
git commit -m "paragraph added"
git log git staus
git branch -M main
git remote add origin url
git push –u origin main
```

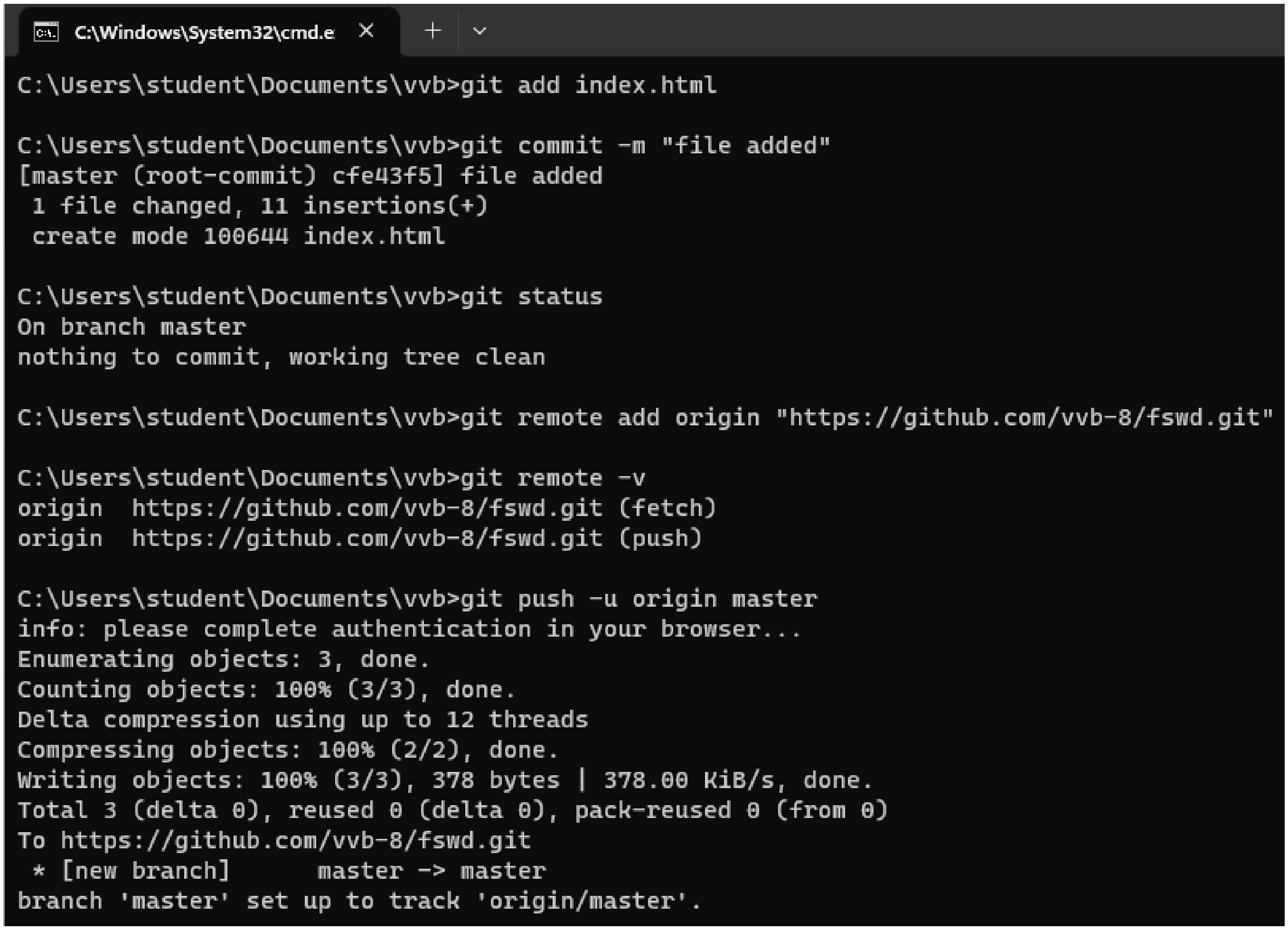
OUTPUT:



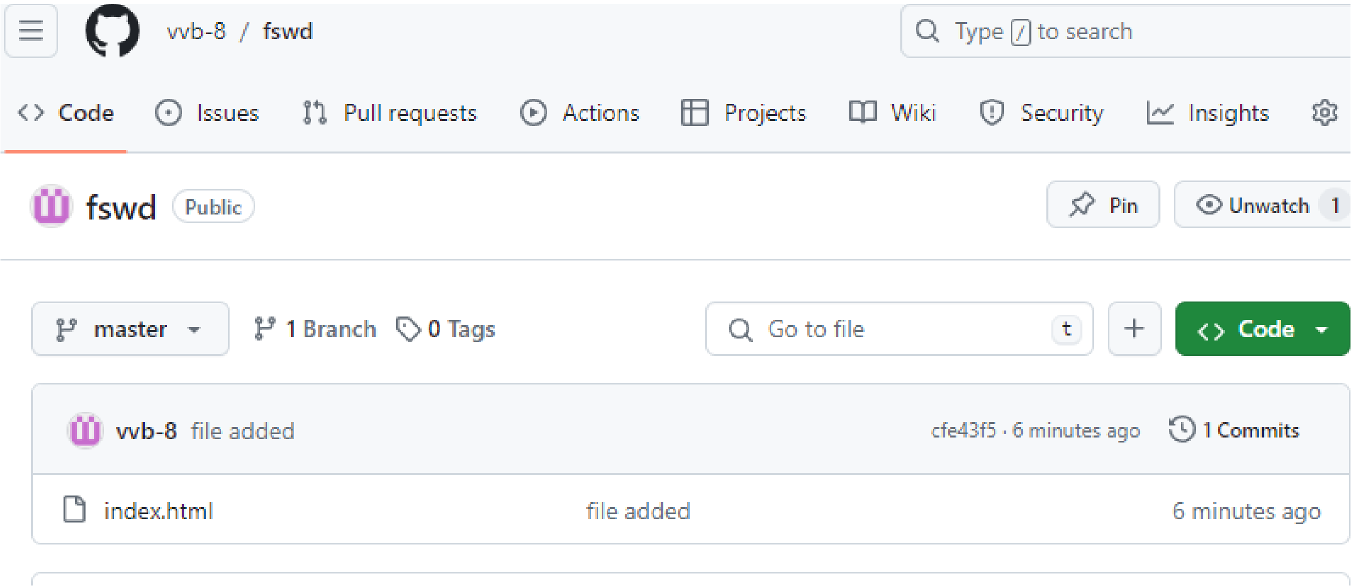
```
yogan@DESKTOP-HGJBBG2 MINGW64 ~ (master)
$ git config --global user.name "cmr"

yogan@DESKTOP-HGJBBG2 MINGW64 ~ (master)
$ git config --global user.email "cmr@example.com"

yogan@DESKTOP-HGJBBG2 MINGW64 ~ (master)
$ git init
Reinitialized existing Git repository in C:/Users/yogan/.git/

yogan@DESKTOP-HGJBBG2 MINGW64 ~ (master)
$ |
```



```
C:\Users\student\Documents\vvb>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\student\Documents\vvb>git add index.html
```



```
C:\Users\student\Documents\vvb>git add index.html

C:\Users\student\Documents\vvb>git commit -m "file added"
[master (root-commit) cfe43f5] file added
 1 file changed, 11 insertions(+)
 create mode 100644 index.html

C:\Users\student\Documents\vvb>git status
On branch master
nothing to commit, working tree clean

C:\Users\student\Documents\vvb>git remote add origin "https://github.com/vvb-8/fswd.git"

C:\Users\student\Documents\vvb>git remote -v
origin  https://github.com/vvb-8/fswd.git (fetch)
origin  https://github.com/vvb-8/fswd.git (push)

C:\Users\student\Documents\vvb>git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 378 bytes | 378.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/vvb-8/fswd.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```



vvb-8 file added cfe43f5 · 6 minutes ago 1 Commits

index.html	file added	6 minutes ago