FSD – LAB-EXTERNAL

Week 1:

1a) Create a Web Page using HTML which contains a Heading, Image and 2 hyperlinks. Each hyperlink opens a new page in the same web browser. New page contains "Go Back" link that takes you to the main page.

Webpage.html:

```
<!DOCTYPE html>
<html>
   <head>
      <style>
      #header
      {
         height: 100px;
         padding: 0px;
         border: 0px;
         text-align: center;
         background-color: beige;
      }
      #img
      {
         float: left;
      }
      #menu
      {
         height: 600px;
         padding: 0px;
         border: 0px;
      }
      #footer
      {
         color: rgb(226, 13, 13);
         text-align: center;
      }
   </style>
   </head>
   <body>
   <div id ="header"><div id="img">
      <img src="gnits_logo.jpeg" alt="Image not Found" width="100"
height="80">
   </div>
```

```
        <h1>GNITS</h1>
    </div>

    <div id="menu">
        <a href="IT.html">IT</a>
        <a href="CSE.html">CSE</a>
    </div>
    <div id="footer">
        <p>CONTACT: xxxxxxxx</p>
    </div>
    </body>
</html>
```

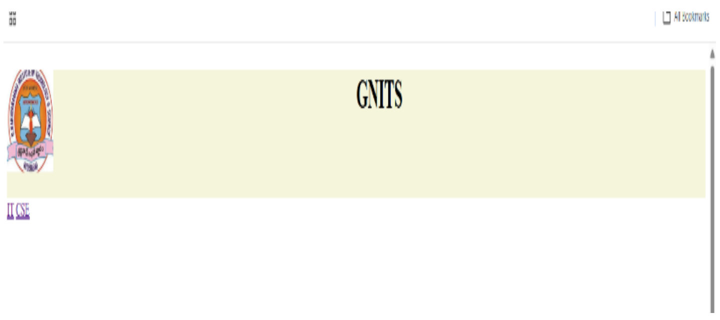
IT.html:

```
<!DOCTYPE html>
<html>
    <head>
        <h1>IT Department</h1>
    </head>
    <body>You're at the IT Page<br>
        <p>Information Technology (IT) encompasses the use of computer
systems, software, networks, and related technologies to manage, process, store,
and exchange information.</p>
        <br>
        <a href="webpage.html">Go back</a>
    </body>
</html>
```

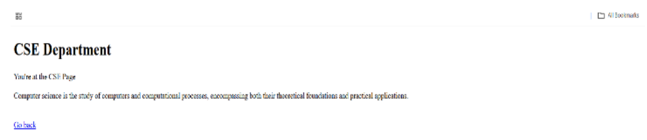
CSE.html

```
<!DOCTYPE html>
<html>
    <head>
        <h1>CSE Department</h1>
    </head>
    <body>You're at the CSE Page<br>
        <p>Computer science is the study of computers and computational
processes, encompassing both their theoretical foundations and practical
applications. </p>
        <br>
        <a href="webpage.html">Go back</a>
    </body>

</html>
```

OUTPUT:


GNITS

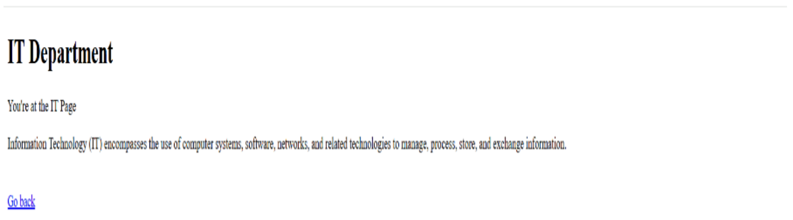
IT CSE


CSE Department

You're at the CSE Page

Computer science is the study of computers and computational processes, encompassing both their theoretical foundations and practical applications.

Go back


IT Department

You're at the IT Page

Information Technology (IT) encompasses the use of computer systems, software, networks, and related technologies to manage, process, store, and exchange information.

Go back

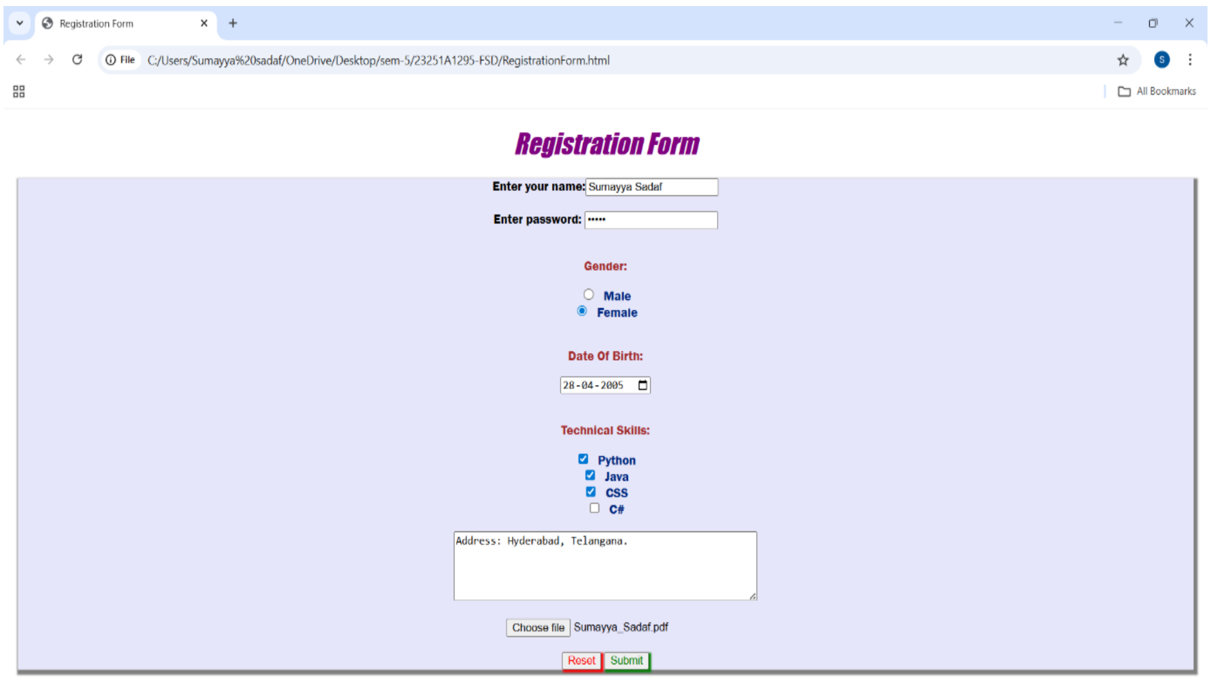
1b) Write a HTML program to create a Registration form, which contains User Name, Password, Date of Birth, Gender, Mail-id, Contact number, Address and submit button.

RegistrationForm.html:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Registration Form</title>
        <style>
            #reg{
                font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-
serif;
                margin: 10px;
                text-align: center;
                box-shadow: 2px 3px 2px 3px grey;
                background-color: lavender;
            }
            label{
                margin-left: 5px;
                color: rgb(0, 38, 128);
            }
            h1{
                font-style: italic;
                font-family: Impact, Haettenschweiler, 'Arial Narrow Bold', sans-
serif;
                font-weight: 100;
                color: purple;
```

```
          }
          #b1{
             color: red;
             box-shadow: 2px 2px 2px 1px;
          }
          #b2{
             color:green;
             box-shadow: 2px 2px 2px 1px;
          }
          p{
             color: brown;
          }
       </style>
       </head><body>
       <h1 ><center>Registration Form</center> </h1>
       <form id="reg">
          Enter your name:<input type="textbox" ><br><br>
          Enter password:    <input type="password"><password></password>
<br><br>
          <p>Gender:</p><input type="radio" id="male" name="Gender"
value="Male">
          <label for="male">Male</label><br>
          <input type="radio" id="female" name="Gender" value="Female">
          <label for="female">Female</label><br><br>
          <p>Date Of Birth:</p><input type="date" ><br><br>
          <p>Technical Skills:</p><input type="checkbox" id="1" name="1"
value="Python">
          <label for="1"> Python</label><br>
          <input type="checkbox" id="2" name="2" value="Java">
          <label for="2"> Java</label><br>
          <input type="checkbox" id="3" name="3" value="CSS">
          <label for="3"> CSS</label><br>
          <input type="checkbox" id="4" name="4" value="C#">
          <label for="4"> C#</label><br><br>
          <textarea rows="5" cols="50" >Address:</textarea><br><br>
          <input type="file" ><br><br>
          <button id="b1">Reset</button>
          <button id="b2">Submit</button>
       </form>
       </body>
</html>
```

OUTPUT:



Registration Form

C:/Users/Sumayya%20sadaf/OneDrive/Desktop/sem-5/23251A1295-FSD/RegistrationForm.html

All Bookmarks

Registration Form

Enter your name: Sumayya Sadaf

Enter password: •••••

Gender:

○ Male
● Female

Date Of Birth:

28-04-2005

Technical Skills:

☑ Python
☑ Java
☑ CSS
☐ C#

Address: Hyderabad, Telangana.

Choose file Sumayya_Sadaf.pdf

Reset Submit

Week 2:

2a) Create a web page to demonstrate Position Property in CSS.

position.html:

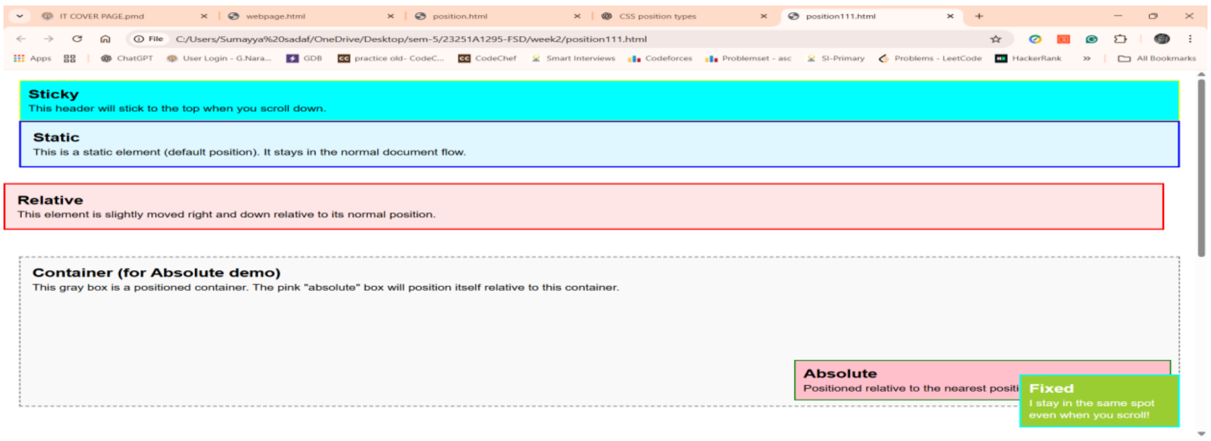
```
<!DOCTYPE html>
<html>
<head>
   <style>
      body {
         font-family: Arial, sans-serif;
         margin: 0;
         padding: 20px;
         line-height: 1.4;
      }
      h1 {
         margin: 0;
         font-size: 22px;
      }
      .section {
         margin-bottom: 40px;
         padding: 15px;
```

```
}
.stat {
   position: static;
   border: 3px solid blue;
   background-color: #e0f7ff;
}
.rel {
   position: relative;
   border: 3px solid red;
   background-color: #ffe6e6;
   right: 20px; /* small visible offset */
   bottom: 10px;
}
.fix {
   position: fixed;
   border: 3px solid aqua;
   right: 20px;
   bottom: 20px;
   background-color: yellowgreen;
   padding: 10px;
   width: 180px;
   color: white;
}
.container {
   position: relative;
   height: 250px;
   border: 2px dashed gray;
   background-color: #f9f9f9;
   margin-top: 20px;
}
.abs {
   position: absolute;
   right: 10px;
   bottom: 10px;
   border: 2px solid green;
   background-color: pink;
   padding: 10px;
}
.sticky {
   position: sticky;
   top: 0;
   padding: 10px;
   background-color: aqua;
```

```
            border: 2px solid yellow;
        }
    </style>
</head>
<body>
    <div class="sticky">
        <h1>Sticky</h1>
        This header will stick to the top when you scroll down.
    </div>

    <div class="section stat">
        <h1>Static</h1>
        This is a static element (default position). It stays in the normal document
flow.
    </div>

    <div class="section rel">
        <h1>Relative</h1>
        This element is slightly moved right and down relative to its normal
position.
    </div>

    <div class="section container">
        <h1>Container (for Absolute demo)</h1>
        This gray box is a positioned container.
        The pink "absolute" box will position itself relative to this container.
        <div class="abs">
            <h1>Absolute</h1>
            Positioned relative to the nearest positioned ancestor (gray box).
        </div>
    </div>
    <div class="section" style="height:600px;">
        <h1>Scroll Down</h1>
        Keep scrolling to see the fixed box stay in the same place and the sticky
box stay at top.
        <p>
            Artificial intelligence (AI) is the capability of computational systems to
perform tasks typically associated with human intelligence,
such as learning, reasoning, problem-solving, perception, and decision-
making...
        </p>
    </div>
    <div class="fix">
```

```
        <h1>Fixed</h1>
        I stay in the same spot even when you scroll!
    </div>
</body>
</html>
```

OUTPUT:



Sticky
This header will stick to the top when you scroll down.

Static
This is a static element (default position). It stays in the normal document flow.

Relative
This element is slightly moved right and down relative to its normal position.

Container (for Absolute demo)
This gray box is a positioned container. The pink "absolute" box will position itself relative to this container.

Absolute
Positioned relative to the nearest positi

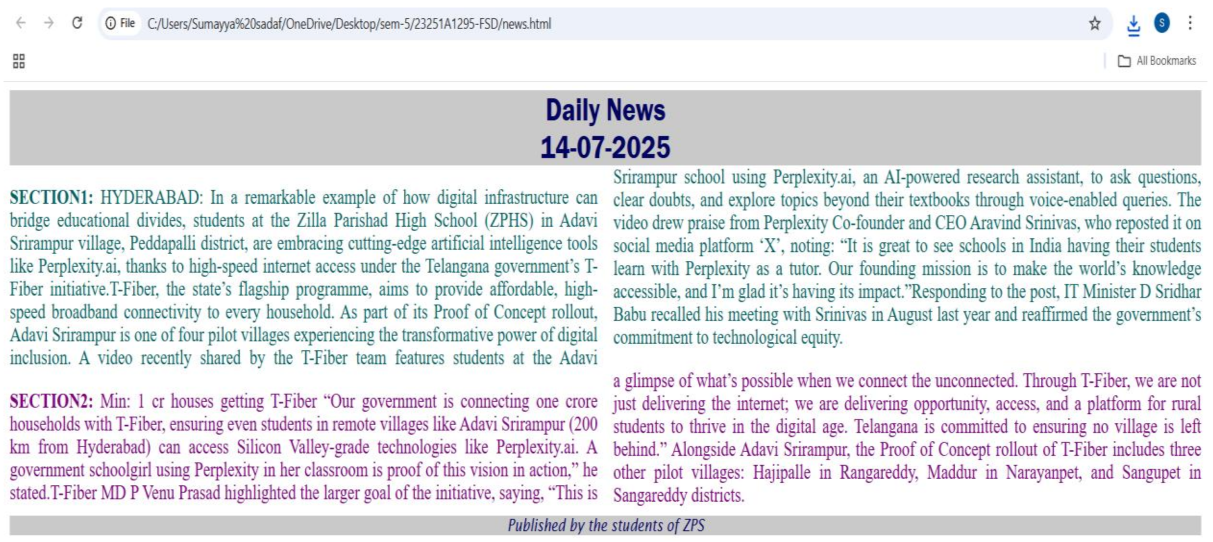
Fixed
I stay in the same spot
even when you scroll!

2b) Create a Newspaper Style Design to print minimum 2 articles using HTML and CSS.

news.html:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <title>News Article</title>
    <style>
        .article{
            column-count: 2;
            column-gap: 20px;
            text-align: justify;
            font-size: 20px;
        }
        header{
            font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
            color: rgb(3, 3, 97);
            font-size: 32px;
            background-color:  rgb(201, 201, 201);
        }
        footer{
            font-style: italic;
            font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande',
'Lucida Sans Unicode', Geneva, Verdana, sans-serif;
            color: rgb(3, 3, 97);
            background-color:rgb(201, 201, 201);
        }
    </style>
</head>
<body>
    <header><center><b>Daily News </b><br> 14-07-2025</center></header>
    <div>
        <section class="article">
<p style="color: rgb(3, 98, 98)"><b>SECTION1:</b>"YOUR
CONTENT"</p>
        </section>
        <section class="article">
<p style="color:purple"><b>SECTION2:</b>"YOUR CONTENT"</p>
        </section>
    </div>
    <footer><center>Published by the students of ZPS</center></footer>
</body>
</html>
```

OUTPUT:



C:/Users/Sumayya%20sadaf/OneDrive/Desktop/sem-5/23251A1295-FSD/news.html

All Bookmarks

Daily News
14-07-2025

SECTION1: HYDERABAD: In a remarkable example of how digital infrastructure can bridge educational divides, students at the Zilla Parishad High School (ZPHS) in Adavi Srirampur village, Peddapalli district, are embracing cutting-edge artificial intelligence tools like Perplexity.ai, thanks to high-speed internet access under the Telangana government's T-Fiber initiative.T-Fiber, the state's flagship programme, aims to provide affordable, high-speed broadband connectivity to every household. As part of its Proof of Concept rollout, Adavi Srirampur is one of four pilot villages experiencing the transformative power of digital inclusion. A video recently shared by the T-Fiber team features students at the Adavi Srirampur school using Perplexity.ai, an AI-powered research assistant, to ask questions, clear doubts, and explore topics beyond their textbooks through voice-enabled queries. The video drew praise from Perplexity Co-founder and CEO Aravind Srinivas, who reposted it on social media platform 'X', noting: "It is great to see schools in India having their students learn with Perplexity as a tutor. Our founding mission is to make the world's knowledge accessible, and I'm glad it's having its impact."Responding to the post, IT Minister D Sridhar Babu recalled his meeting with Srinivas in August last year and reaffirmed the government's commitment to technological equity.

SECTION2: Min: 1 cr houses getting T-Fiber "Our government is connecting one crore households with T-Fiber, ensuring even students in remote villages like Adavi Srirampur (200 km from Hyderabad) can access Silicon Valley-grade technologies like Perplexity.ai. A government schoolgirl using Perplexity in her classroom is proof of this vision in action," he stated.T-Fiber MD P Venu Prasad highlighted the larger goal of the initiative, saying, "This is a glimpse of what's possible when we connect the unconnected. Through T-Fiber, we are not just delivering the internet; we are delivering opportunity, access, and a platform for rural students to thrive in the digital age. Telangana is committed to ensuring no village is left behind." Alongside Adavi Srirampur, the Proof of Concept rollout of T-Fiber includes three other pilot villages: Hajipalle in Rangareddy, Maddur in Narayanpet, and Sangupet in Sangareddy districts.

Published by the students of ZPS

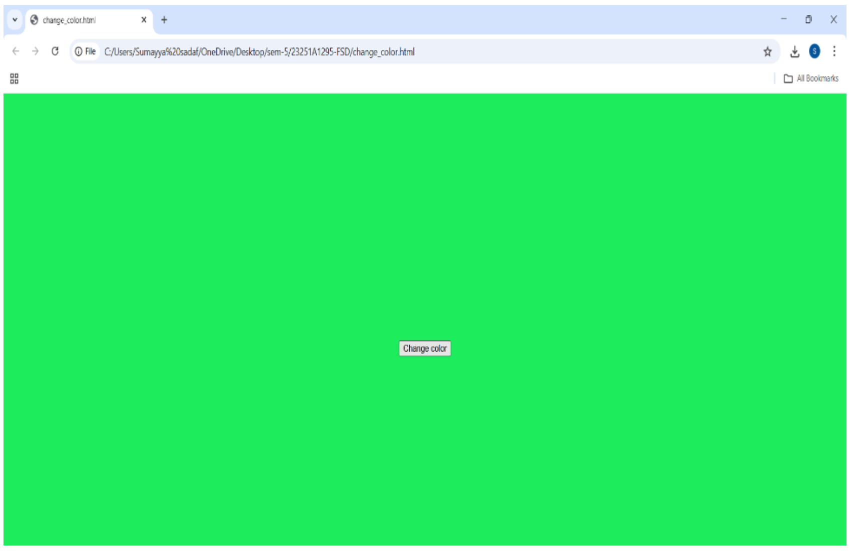
Week 3:

3a) Write a JavaScript program to change the background color after clicking "change color" button.

Change_color.html

```
<!DOCTYPE html>
<html><head>
   <style>
      body {
         margin: 0;
         height: 100vh;
         display: flex;
         justify-content: center;
         align-items: center;
      }
   </style>
   <script>
   function getRandomColor()
   {
      const letters='0123456789ABCDEF';
      let color='#';

      for(let i=0;i<6;i++){
         color+=letters[Math.floor(Math.random()*16)];
      }
      document.body.style.backgroundColor=color;
   }
   </script></head><body>
   <button onclick="getRandomColor()">Change color</button>
   </body>
</html>
```

OUTPUT:


change_color.html
File
C:/Users/Sumayya%20sadaf/OneDrive/Desktop/sem-5/23251A1295-FSD/change_color.html
All Bookmarks
Change color

3b) Write a JavaScript program to validate registration page using regular expression.

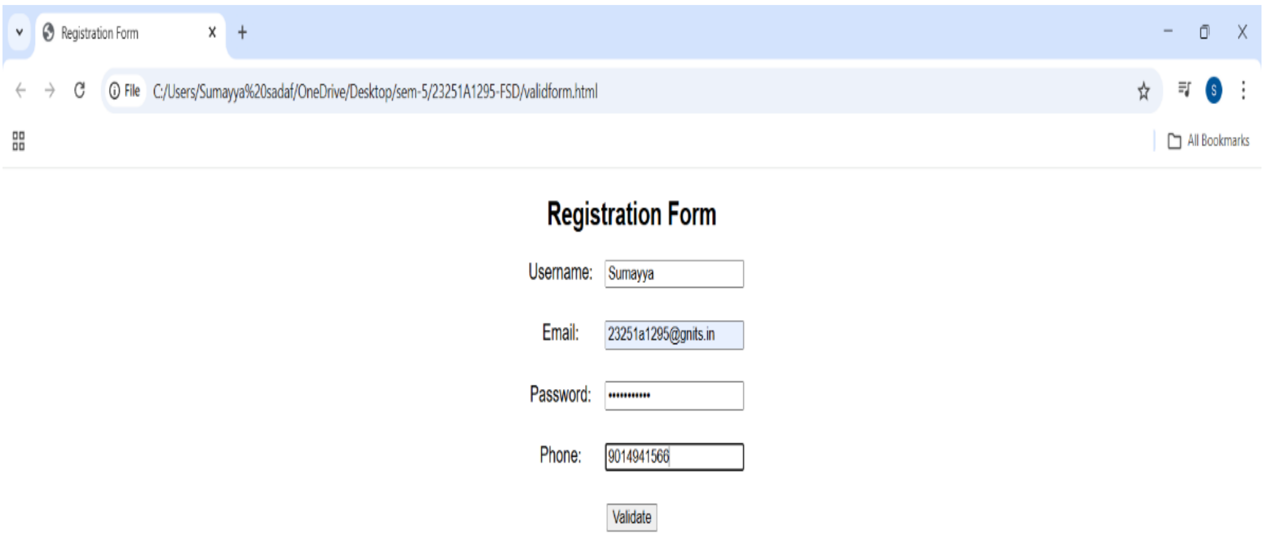
validform.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Registration Form</title>
   <style>
      body
      {
         text-align: center;
         font-family: Arial, Helvetica, sans-serif;
         margin: 20px;
      }
      label{
         display: inline-block;
         width: 100px;;
         margin-bottom: 8px;
      input[type="text"],input[type="password"]
      {
         padding:6px;
         width: 200px;
      }
      #errorMessage
      {
         color:red;
         margin-top: 15px;
      }
      #successMessages
      {
         color: green;
         margin-top: 15px;
      }
      }
   </style>
</head>
<body><centre>
   <h2>Registration Form</h2>
   <form id="registrationForm">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username"><br><br>
```

```
    <label for="email">Email:</label>
    <input type="text" id="email" name="email"><br><br>

    <label for="password">Password:</label>
    <input type="password" id="password" name="password"><br><br>

    <label for="phone">Phone:</label>
    <input type="text" id="phone" name="phone"><br><br>

    <button type="submit">Validate</button>
  </form>

  <script>
    const form = document.getElementById("registrationForm");

    form.addEventListener("submit", function(event) {
      event.preventDefault();

      const username = document.getElementById("username").value;
      const email = document.getElementById("email").value;
      const password = document.getElementById("password").value;
      const phone = document.getElementById("phone").value;

      const usernameRegex = /^[a-zA-Z0-9]{3,15}$/;
      const emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-
Z]{2,}$/;
      const passwordRegex = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{6,}$/;
      const phoneRegex = /^[0-9]{10}$/;

      if (!usernameRegex.test(username)) {
        alert("Invalid Username");
        return;
      }
      if (!emailRegex.test(email)) {
        alert("Invalid Email");
        return;
      }
      if (!passwordRegex.test(password)) {
        alert("Invalid Password (must have uppercase, lowercase, number)");
        return;
      }
      if (!phoneRegex.test(phone)) {
        alert("Invalid Phone (must be 10 digits)");
```

```
                return;
            }
            alert("All inputs are valid");
        });
    </script></centre>
</body>
</html>
```

OUTPUT:



Registration Form

C:/Users/Sumayya%20sadaf/OneDrive/Desktop/sem-5/23251A1295-FSD/validform.html

Registration Form

Username:	Sumayya
Email:	23251a1295@gnits.in
Password:	••••••••••
Phone:	9014941566

Week 4:

4a) Write a code to hide and show an element in a periodic interval without any action from the user using JQuery.

Toggle.html

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" type="text/css" media="screen" href="toggle.css">
    </head>
    <body>
        <div id='toggleElement'>This element will toggle visibility</div>
        <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
        <script src="toggle.js"></script>
    </body>
</html>
```

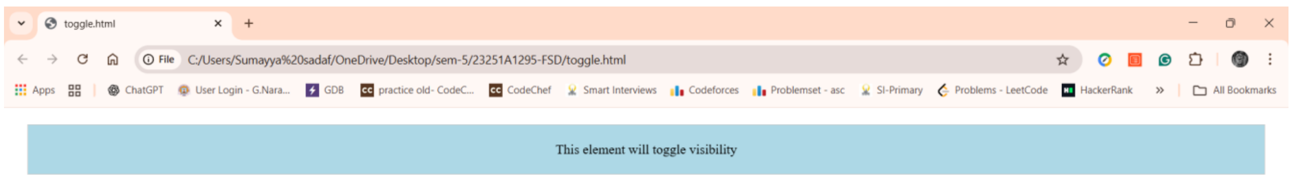
Toggle.css

```
#toggleElement{
    padding: 20px;
    background-color: lightblue;
    text-align: center;
    margin: 20px;
    border: 1px solid #ccc;
}
```

Toggle.js

```
$(document).ready(function(){
    var duration=1000;
    function toggleVisibility(){
        $('#toggleElement').toggle(1000);
    }
    setInterval(toggleVisibility,duration);
})
```

OUTPUT:



This element will toggle visibility

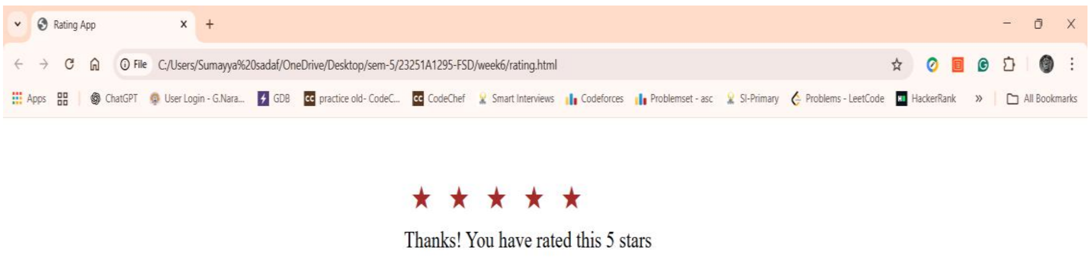
4b) Write a program to create and Build a star rating system using JQuery.

rating.html

```
<!DOCTYPE html>
<html>
   <head>
      <title>Rating App</title>
      <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
      <style>
         #container{
            width: 400px;
            height: 200px;
            margin: 50px auto;
         }
         ul{
            padding: 0px;
            margin: 0px;
         }
         li{
            display: inline-block;
            font-size: 35px;
            padding: 10px;
            color: #ccc;
         }
         #message{
            font-size: 25px;
         }
         .hovered-stars{
            color: yellow;
         }
         .clicked-stars{
            color: brown;
         }
      </style>
      <script>
         $(document).ready(function(){
            $("li").mouseover(function(){
               var current=$(this);
               $("li").each(function(index){
                  $(this).addClass("hovered-stars");
                  if(index==current.index()){
                     return false;
                  }
```

```
                });
            });
            $("li").mouseleave(function(){
                $("li").removeClass("hovered-stars");
            });
            $("li").click(function(){
                $("li").removeClass("clicked-stars");
                $(".hovered-stars").addClass("clicked-stars");
                $("#message").html("Thanks! You have rated this "+$(".clicked-
stars").length + " stars");
            });
        });

    </script>
  </head>
  <body>
    <div id="container">
      <ul>
        <li>&starf;</li>
        <li>&starf;</li>
        <li>&starf;</li>
        <li>&starf;</li>
        <li>&starf;</li>
      </ul>
      <div id="message">Please rate your experience!</div>
    </div>
  </body>
</html>
```

OUTPUT:


Rating App
C:/Users/Sumayya%20sadaf/OneDrive/Desktop/sem-5/23251A1295-FSD/week6/rating.html
★ ★ ★ ★ ★

```
Thanks! You have rated this 5 stars
```

Week 5:

5a) Write a program to demonstrate ReactJS Class and Instance.

Steps:

1. In VSCode -> select Command prompt->
2. **npm install -g create-react-app ->** 64 packages get installed
3. **npx create-react-app projectname**
4. After you see **Happy Hacking!** You have successfully created a project.
5. Make sure u are in the right path of your project. **cd projectname**
6. **npm start ->** to run the project
7. **Go to src> create your .js Component Files**
8. A component allows a single parent tag , so include all your tags inside 1 <div> tag

App.js:

```
import './App.css';
import HelloClass from './HelloClass';
import Hellofun from './Hellofun';
function App() {
  return (
    <div>
      <h1>WELCOME TO REACT!</h1>
      <HelloClass />
      <Hellofun />
    </div>
  );
}

export default App;
```

Hellofun.js:

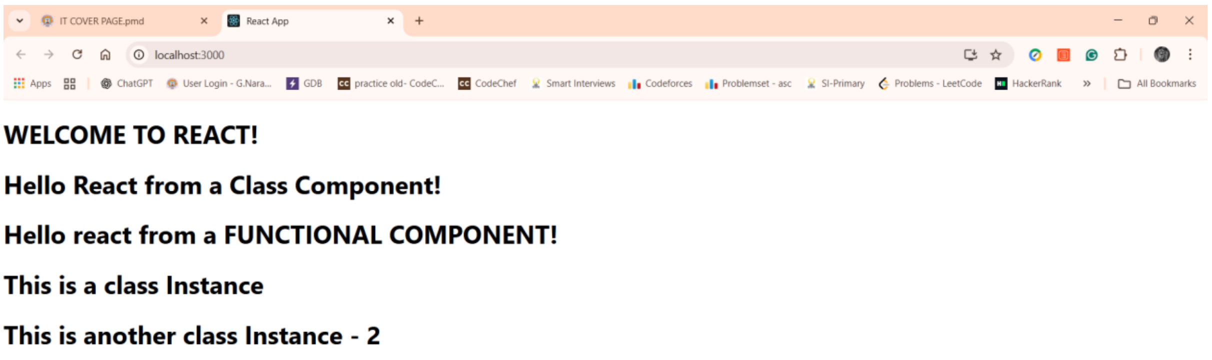
```
import React from 'react';
export default function Hellofun(){
    return (
        <div>
            <h1>Hello react from a FUNCTIONAL COMPONENT!</h1>
            <ClassInst/>
        </div>
    );
}
```

```
class ClassInst extends React.Component{
   render(){
      return(
         <div>
            <h1>This is a class Instance</h1>
            <ClassInst1/>
         </div>
      );
   }
}

class ClassInst1 extends React.Component{
   render(){
      return(
         <div>
            <h1>This is another class Instance - 2</h1>
         </div>
      );
   }
}
```

HelloClass.js

```
import React from 'react';
export default class HelloClass extends React.Component{
   render(){
      return(
         <div>
            <h1>Hello React from a Class Component!</h1>
         </div>
      );
   }
}
```

OUTPUT:



```
WELCOME TO REACT!
Hello React from a Class Component!
Hello react from a FUNCTIONAL COMPONENT!
This is a class Instance
This is another class Instance - 2
```

5b) Write a program to create a basic calculator to perform arithmetic operations using ReactJS.

Part1: Calculator.js:

```
import React,{useState} from 'react';
import './Calculate.css';
const Calculate=()=>{
   const [data,setData]=useState(" ");
   const getValue=(event)=>{
      console.log(event.target.value);
      setData(data.concat(event.target.value));
   }
   const Calculate=()=>{
      setData(eval(data).toString());
   }
   const back=()=>{
      setData(data.slice(0,-1));
   }
   const clear=()=>{
      setData("");
   }
   return(
      <>
      <div className='Container'>
         <div>
            <input placeholder='0' value={data}/>
         </div><br/>
         <button onClick={getValue} value="(">(</button>
         <button onClick={getValue} value=")">)</button>
         <button onClick={getValue} value="%">%</button>
         <button onClick={clear} value="AC">AC</button>

         <button onClick={getValue} value="7">7</button>
         <button onClick={getValue} value="8">8</button>
         <button onClick={getValue} value="9">9</button>
         <button onClick={getValue} value="*">*</button>

         <button onClick={getValue} value="4">4</button>
         <button onClick={getValue} value="5">5</button>
         <button onClick={getValue} value="6">6</button>
         <button onClick={getValue} value="-">-</button>

         <button onClick={getValue} value="1">1</button>
```

```
            <button onClick={getValue} value="2">2</button>
            <button onClick={getValue} value="3">3</button>
            <button onClick={getValue} value="+">+</button>

            <button onClick={getValue} value="0">0</button>
            <button onClick={back} value="Back">Back</button>
            <button onClick={Calculate} value="=">=</button>
            <button onClick={getValue} value="/">/</button>
        </div></>
    )
}
export default Calculate;
```

Calculate.css

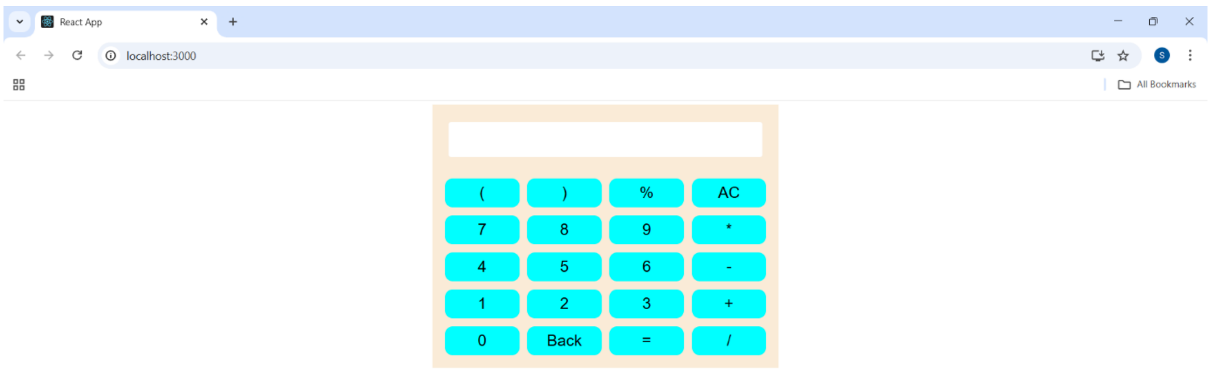
```
.Container{
    width: 420px;
    background-color: antiquewhite;
    margin: auto;
    text-align: center;
    padding: 11px;
    margin-top: 5px;
}

.Container input{
    width: 90%;
    padding: 11px;
    border: none;
    font-size: 20px;
    margin-top: 11px;
    border-radius: 3px;
}

.Container button{
    width: 95px;
    padding: 7px;
    margin: 5px;
    border: none;
    font-size: 20px;
    background-color: aqua;
    border-radius: 10px;
    cursor: pointer;
}
```

```
.Container equal{
    background-color: aquamarine;
    color: aliceblue;
}
```

App.js:

```
import './App.css';
import Calculate from './Calculator';
function App() {
  return (
    <div>
      <Calculate/>
    </div>
  );
}

export default App;
```

OUTPUT:



Part1: NumericInput.js:

npm install react-numeric-input

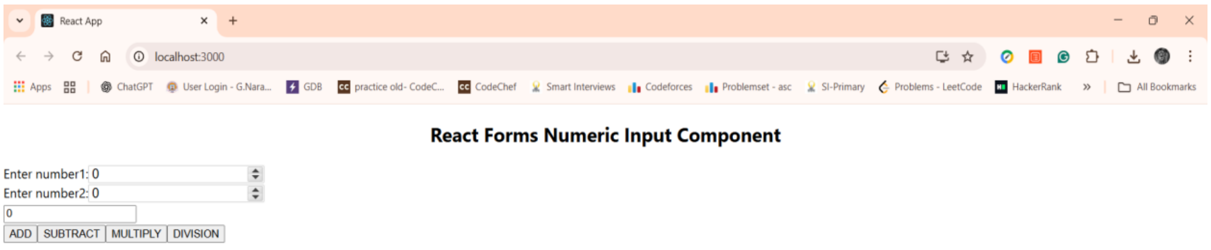
```
import React from "react";
import NumericInput from 'react-numeric-input';

class NumericInput1 extends React.Component{
    constructor(props){
        super(props)
        this.state={
            n1:props.value||0,
            n2:props.value||0,
            result:0
        };
    }
    add=()=>{
        this.setState({
            result:parseInt(this.state.n1)+parseInt(this.state.n2)
        });
    };

    sub=()=>{
        this.setState({
            result:parseInt(this.state.n1)-parseInt(this.state.n2)
        });
    };

    multiply=()=>{
        this.setState({
            result:parseInt(this.state.n1)*parseInt(this.state.n2)
        });
    };

    division=()=>{
        this.setState({
            result:parseInt(this.state.n1)/parseInt(this.state.n2)
        });
    };

    handleChange=(name,value)=>{
        this.setState({[name]:value||0});
    };
    myFormat(num){
        return num+"$";
    };
    render(){
        const {n1,n2,result}=this.state;
        return(
            <div className="NumericInput1">
                <center><h2>React Forms Numeric Input Component</h2></center>
                <form>
                    <label>Enter number1:</label>
```

```
                    <NumericInput name='n1' value={n1}
                    onChange={(value)=>this.handleChange('n1',value)}/><br></br>
                    <label>Enter number2:</label>
                    <NumericInput name='n2' value={n2}
                    onChange={(value)=>this.handleChange('n2',value)}/><br></br>
                    <input type="text" name="result" value={result}
readOnly/><br></br>
                    <button type="button" onClick={this.add}>ADD</button>
                    <button type="button" onClick={this.sub}>SUBTRACT</button>
                    <button type="button"
onClick={this.multiply}>MULTIPLY</button>
                    <button type="button"
onClick={this.division}>DIVISION</button>
                </form>
            </div>
        )
    }
}

export default NumericInput1;
```

App.js:

```
import './App.css';
import NumericInput1 from './NumericInput';
function App() {
  return (
    <div>
      <NumericInput1/>
    </div>
  );
}

export default App;
```

OUTPUT:



React Forms Numeric Input Component

Enter number1: 0
Enter number2: 0
0
ADD SUBTRACT MULTIPLY DIVISION

Week 6:

6a) Demonstrate simple event handling examples using ReactJS.

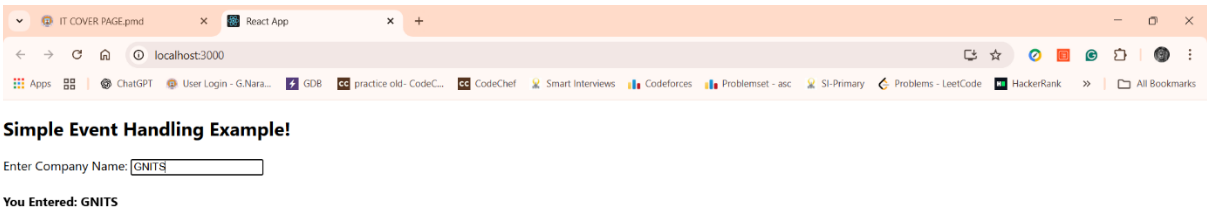
event.js:

```
import React from "react";
export default class EventHandling extends React.Component{
    constructor(props){
        super(props);
        this.state={companyName: " "};
    }
    changeText(events){
        this.setState({
            companyName:events.target.value
        })
    }
    render(){
        return(
            <div>
                <h2>Simple Event Handling Example!</h2>
                <label>Enter Company Name: </label>
                <input type="text" id="companyName"
onChange={this.changeText.bind(this)}/>
                <h4>You Entered: {this.state.companyName}</h4>
            </div>
        );
    }
}
```

App.js:

```
import './App.css';
import EventHandling from './event';
function App() {
  return (
    <div>
      <EventHandling/>
    </div>
  );
}

export default App;
```

OUTPUT:



Simple Event Handling Example!

Enter Company Name: GNITS

You Entered: GNITS

6b) Write a program to create a simple voting application system using ReactJS.

vote.js:

```
import React from "react";
import './vote.css';
class Vote extends React.Component{
    constructor(props){
        super(props)
            this.state={
                languages: [{ name:"Java", vote:0 },
                            { name:"Python", vote:0 },
                            { name:"Html", vote:0 },
                            { name:"CSS", vote:0 },
                            { name:"JQuery", vote:0 },
                ]
            }
    }
    vote(i){
        let newLanguageVote=this.state.languages;
        console.log("click on:"+newLanguageVote[i].name);
        console.log("click: "+i);
        newLanguageVote[i].vote++;
        this.setState({
        languages: newLanguageVote
        });
    }
    render(){
        return(
            <div>
                <h1>Vote your favourite language</h1>
                <div className="languages"> {
this.state.languages.map((language,i)=>
                        <div key={i} className="language">
                            <div className="voteCount">
                                {language.vote}
                            </div>
                            <div className="languageName">
                                {language.name}
                            </div>
                            <button onClick={this.vote.bind(this,i)}>Click
Here!</button>
                        </div>
                    )}
                    </div>
                </div>
            );
        }
}
export default Vote;
```

vote.css:

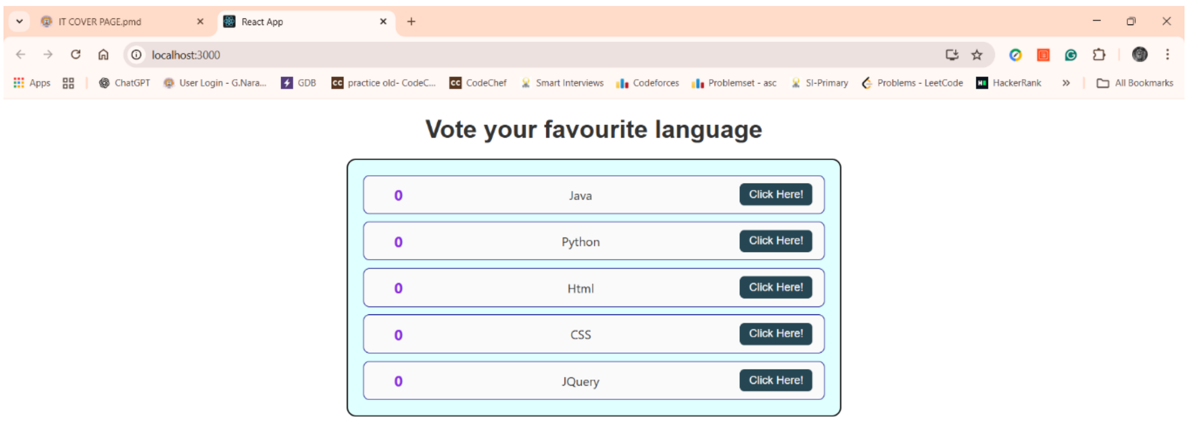
```
/* Heading centered */
h1 {
  text-align: center;
  margin-bottom: 20px;
  font-family: Arial, sans-serif;
  font-size: 20;
  color: #333;
}

/* Container box */
.languages {
  display: flex;
  flex-direction: column;   /* stack rows vertically */
  gap: 10px;
  padding: 20px;
  border: 2px solid black;
  border-radius: 12px;
  background: lightcyan;
  max-width: 600px;
  margin: 0 auto;           /* center horizontally */
}
```

```
/* Header row */
.languages::before {
  display: flex;
  justify-content: space-between;
  font-weight: bold;
  padding: 10px 15px;
  border-bottom: 2px solid #ccc;
  color: #444;
}

/* Each language row */
.language {
  display: flex;
  flex-direction: row;      /* align horizontally */
  justify-content: space-between;
  align-items: center;
  padding: 10px 15px;
  border: 1px solid navy;
  border-radius: 8px;
  background: #fafafa;
  transition: background 0.2s ease;
}

.language:hover {
  background: #f0f0f0;
}

/* Vote count */
.voteCount {
  font-size: 1.2rem;
  font-weight: bold;
  color: blueviolet;
  width: 60px;              /* fixed width for alignment */
  text-align: center;
}

/* Language name */
.languageName {
  flex: 1;                  /* take remaining space */
  font-size: 10;
  text-align: center;
  color: #333;
}

/* Button */
.language button {
  padding: 6px 12px;
  border: none;
  border-radius: 6px;
  background: #264653;
  color: #fff;
  cursor: pointer;
  font-size: 0.9rem;
  transition: background 0.3s ease;
}

.language button:hover {
  background: #2a9d8f;
}
```

App.js:

```
import './App.css';
import Vote from './vote';
function App() {
  return (
    <div>
      <Vote/>
    </div>
  );
}
export default App;
```

OUTPUT:



Vote your favourite language

```
0    Java      Click Here!
0    Python    Click Here!
0    Html      Click Here!
0    CSS       Click Here!
0    JQuery    Click Here!
```

Week 7:

7a) Create a webpage to display “Hello World” using SERVLET.

Steps:

(If you don’t find Dynamic Web Project then Window-> Perspective->Open Perspective ->Other.. -> Java EE->Open)

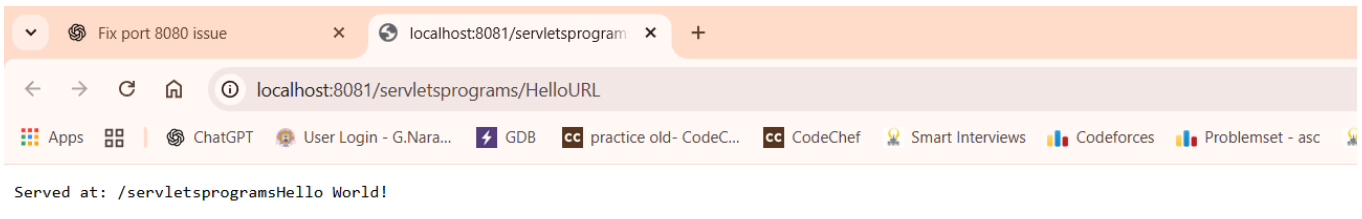
1. **File -> New -> Dynamic Web Project**
2. **Enter your projectname**
3. **Select “New Runtime” -> Apache -> Apache tomcat v9.0 ->Next**
4. **Browse for Apache Software Foundation\Tomcat 9.0-> Finish->Next**
5. **Next-> Select the checkbox “Generate web.xml” ->Finish.**
6. **Create a Servlet:**
7. **Right Click on Project -> New -> Servlet**
8. **Create Servlet -> class name: your servlet name ->Next**
9. **URL -> edit**
10. **Check only doGet Method -> Finish.**

HelloWorld.java:

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/HelloURL")
public class HelloWorld extends HttpServlet {
        private static final long serialVersionUID = 1L;
   public HelloWorld() {
      super();
      // TODO Auto-generated constructor stub
   }
        protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
                // TODO Auto-generated method stub
                response.getWriter().append("Served at:
").append(request.getContextPath());
```

```
                PrintWriter out= response.getWriter();
                out.println("Hello World!");
                }

}
```

OUTPUT:



```
Served at: /servletsprogramsHello World!
```

7b) Implement a web application using SERVLET, which takes a name as input and on submitting it, shows a hello page. It shows start time at the right top comer of the page and provides a logout button. On clicking logout button, it should show a logout page with Thank You message with the duration of usage (hint: Use session to store name and time).

SessionOut.java:

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/TimeOut")
public class SessionOut extends HttpServlet {
        private static final long serialVersionUID = 1L;

        protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();
```

```
        HttpSession session = request.getSession(false);
        if (session == null) {
            out.println("<html><body><h3>No active session. Please <a
href='first.html'>login again</a>.</h3></body></html>");
            return;
        }

        java.util.Date d2 = new java.util.Date();
        String un = (String) session.getAttribute("user");
        Long t1 = (Long) session.getAttribute("time");
        Long t2 = d2.getTime();
        session.invalidate();

        long durationSeconds = (t2 - t1) / 1000;

        out.println("<html><body>");
        out.println("<h3>Thank you, " + un + "!</h3>");
        out.println("<p>Session duration: " + durationSeconds + " seconds</p>");
        out.println("</body></html>");
    }
}
```

SessionIn.java:

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/TimeIn")
public class SessionIn extends HttpServlet {
    /**
         *
         */
        private static final long serialVersionUID = 1L;

        protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String un = request.getParameter("uname");
        java.util.Date date = new java.util.Date();

        HttpSession session = request.getSession();
        session.setAttribute("user", un);
        session.setAttribute("time", date.getTime());

        out.println("<html><body>");
        out.println("<p align='right'>Login Time: " + date.toString() + "</p>");
        out.println("<h3>Hello, " + un + "!</h3>");
        out.println("<form method='get' action='TimeOut'>");
        out.println("<input type='submit' value='Logout'/>");
        out.println("</form>");
        out.println("</body></html>");
    }
}
```

First.html:

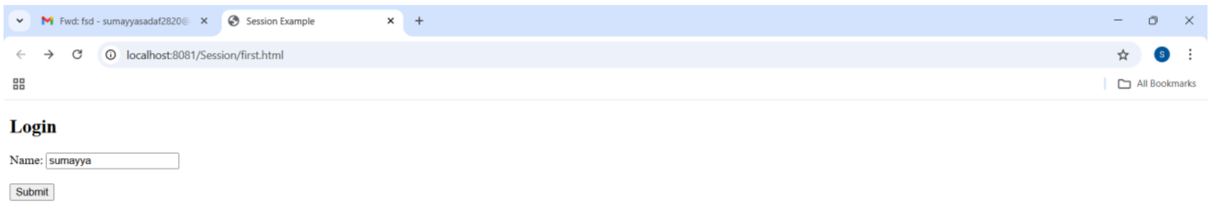
```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Session Example</title>
</head>
<body>
<h2>Login</h2>
<form action="TimeIn" method="get">
  <label for="name">Name:</label>
  <input type="text" name="uname" id="name" required/><br/><br/>
  <button type="submit">Submit</button>
</form>
</body>
</html>
```

If you get a 404 error, change a little in web.xml
Add the following highlighted text to this web-app tag:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns="http://xmlns.jcp.org/xml/ns/javaee"
```

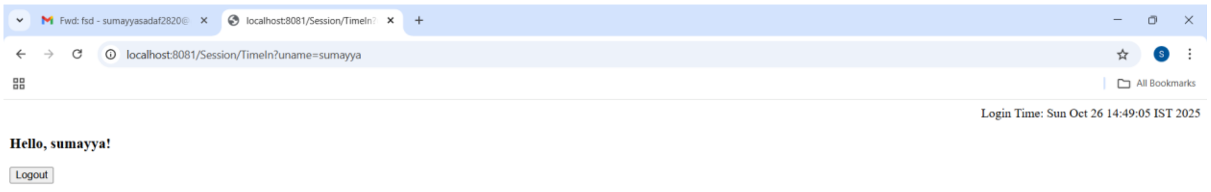
```
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                     http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
version="4.0"
id="WebApp_ID"
metadata-complete="false">
```

OUTPUT:


Fwd: fsd - sumayyasadaf2820@
Session Example
localhost:8081/Session/first.html
All Bookmarks
Submit

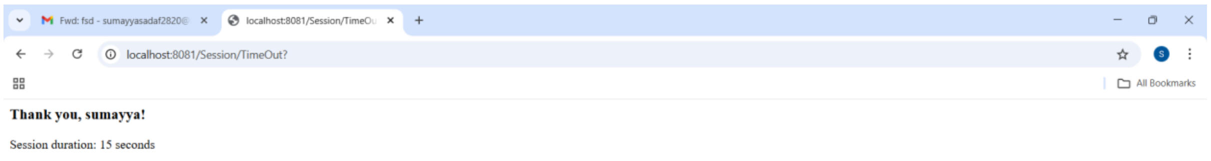
Login

Name: sumayya


Fwd: fsd - sumayyasadaf2820@
localhost:8081/Session/TimeIn?
localhost:8081/Session/TimeIn?uname=sumayya
All Bookmarks
Logout

Login Time: Sun Oct 26 14:49:05 IST 2025

Hello, sumayya!


Fwd: fsd - sumayyasadaf2820@
localhost:8081/Session/TimeOu
localhost:8081/Session/TimeOut?
All Bookmarks

Thank you, sumayya!

Session duration: 15 seconds

Week 8:

8a) Write a JSP program to find a factorial of the given number.

Steps:

1. **File->New->Create a dynamic web project**
2. **Right click on project-> Create JSP File**
3. **Right click on project-> Create HTML File**

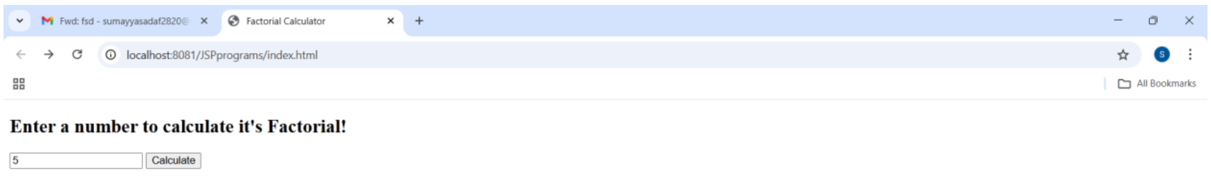
Factorial.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<%@ page import="javax.servlet.*, javax.servlet.http.*" %>

<%!
long factorial(long n) {
   if (n == 0)
      return 1;
   else
      return n * factorial(n - 1);
}
%>

<%
   String str = request.getParameter("num");
   long n = Long.parseLong(str);
   long result = factorial(n);
%>

<b>Factorial Value:</b> <%= result %>
```

Index.html:

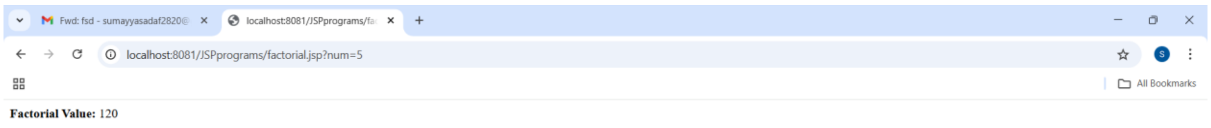
```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Factorial Calculator</title>
</head>
<body>
        <h2>Enter a number to calculate it's Factorial!</h2>
        <form action="factorial.jsp">
        <input type="text" name="num"/>
        <input type="submit" value="Calculate"/>
        </form>
</body>
</html>
```

OUTPUT:


Fwd: fsd - sumayyasadaf2820@ | Factorial Calculator
localhost:8081/JSPprograms/index.html

Enter a number to calculate it's Factorial!

5 Calculate


Fwd: fsd - sumayyasadaf2820@ | localhost:8081/JSPprograms/fa
localhost:8081/JSPprograms/factorial.jsp?num=5

Factorial Value: 120

8b) Create a user validation web application using JSP, where the user submits the login name and password to the server. The name and password are checked against the data already available in database and if the data matches, a successful login page is returned. Otherwise show a failure message to the user.

DatabaseConnect.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ page import="java.io.*,java.sql.*"%>
<%@ page import="javax.servlet.http.*,javax.servlet.*"%>
<%
    // Get username and password from form
    String user = request.getParameter("username");
    String password = request.getParameter("password");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Login Result</title>
</head>
```

```
<body>
<%
   if(user != null && password != null && !user.isEmpty() &&
!password.isEmpty()){
      Connection con = null;
      PreparedStatement ps = null;
      ResultSet rs = null;
      try{
         // Load MySQL driver
         Class.forName("com.mysql.cj.jdbc.Driver");
         con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/sumayya","root","r
oot");

         // Check if username and password match
         ps = con.prepareStatement("SELECT username FROM valid WHERE
username=? AND password=?");
         ps.setString(1, user);
         ps.setString(2, password);
         rs = ps.executeQuery();

         if(rs.next()){
            out.println("<h3>Hello " + user + "</h3>");
         } else {
            // Check if username exists
            ps = con.prepareStatement("SELECT username FROM valid WHERE
username=?");
            ps.setString(1, user);
            rs = ps.executeQuery();
            if(rs.next()){
               out.println("<h3>Invalid password</h3>");
            } else {
               out.println("<h3>Invalid Credentials</h3>");
            }
         }
      } catch(Exception e){
         out.println("<h3>Error: " + e.getMessage() + "</h3>");
      } finally {
         try { if(rs != null) rs.close(); } catch(Exception e){}
         try { if(ps != null) ps.close(); } catch(Exception e){}
         try { if(con != null) con.close(); } catch(Exception e){}
      }
   } else {
        out.println("<h3>Enter Value</h3>");
    }
%>
</body>
</html>
```

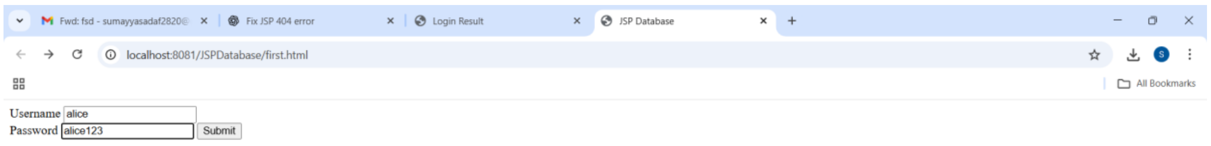
First,html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>JSP Database</title>
</head>
<body>
<form action="DatabaseConnect.jsp">
        <label id="name">Username</label>
        <input type="text" name="username"/><br/>
        <label id="pass">Password</label>
        <input type="text" name="password"/>
        <input type="submit" name="Sign In">
</form>
</body>
</html>
```

1. Copy paste the mysql-connector-j-8.1.0.jar file in WEB-INF-> lib
2. From cmd to enter into mysql: **mysql -u root -p**
3. Create a table valid in Database:

```
CREATE TABLE valid (
    username VARCHAR(50) PRIMARY KEY,
    password VARCHAR(50) NOT NULL
 );
```

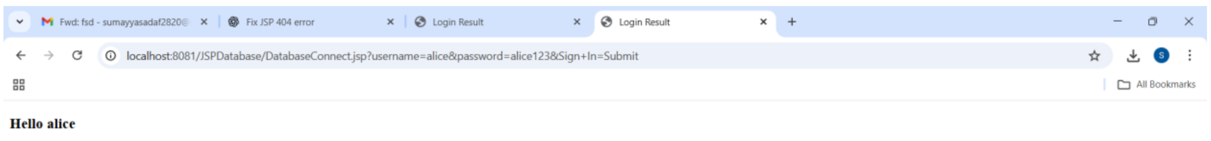
OUTPUT:


Fwd: fsd - sumayyasadaf2820@ | Fix JSP 404 error | Login Result | JSP Database

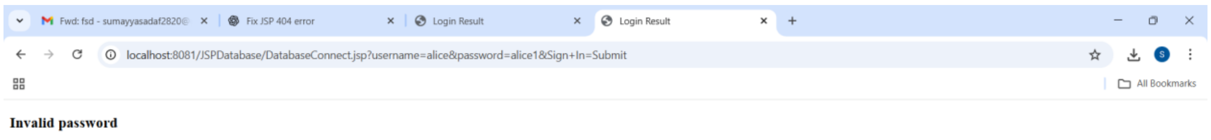
localhost:8081/JSPDatabase/first.html

All Bookmarks

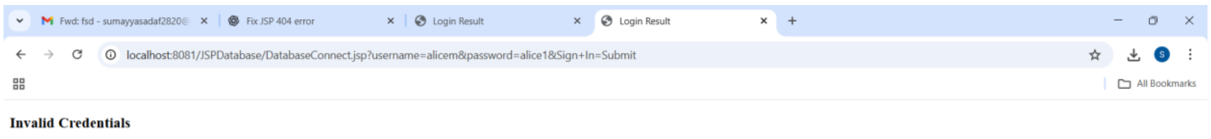
Username alice
Password alice123 Submit


localhost:8081/JSPDatabase/DatabaseConnect.jsp?username=alice&password=alice123&Sign+In=Submit

Hello alice


localhost:8081/JSPDatabase/DatabaseConnect.jsp?username=alice&password=alice1&Sign+In=Submit

Invalid password


localhost:8081/JSPDatabase/DatabaseConnect.jsp?username=alicem&password=alice1&Sign+In=Submit

Invalid Credentials

Week 9:

9a) Demonstrate a simple example of Spring web MVC framework.

Steps:

1. Create Maven project
2. Select filter: co.ntier->Select the spring-mvc
3. GroupId: com.example
4. ArtifcatId= MyApp->Finish
5. Open pom.xml
6. Paste the following within <plugins> tag :

```
<plugin>
   <groupId>org.apache.maven.plugins</groupId>
   <artifactId>maven-war-plugin</artifactId>
   <version>3.3.1</version>
</plugin>
```

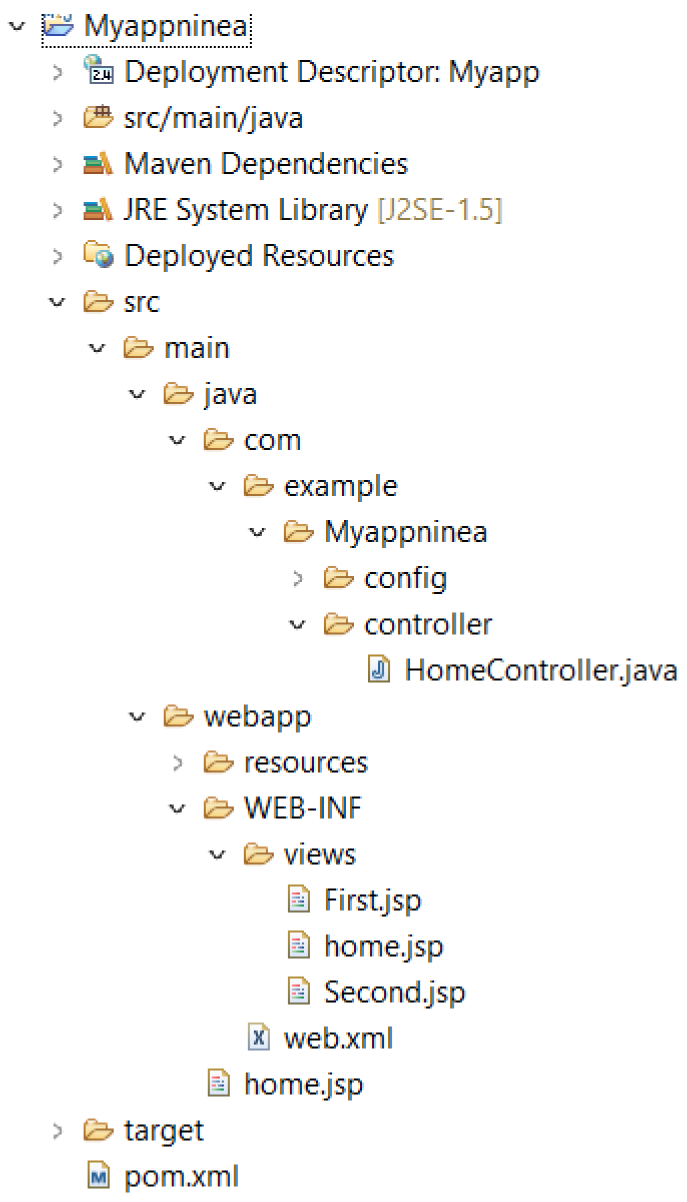
7. Change Java version to 1.8 in properties tag as follows :

```
<properties>
<java.version>1.8</java.version>
<spring.version>3.1.0.RELEASE</spring.version>
<cglib.version>2.2.2</cglib.version>
</properties>
```

8. Save pom.xml
9. Clean the Maven Project, Right click on project → Run as → Maven Clean
10. Update Project ----Maven > Update Project
11. Select "Force Update of Snapshots/Releases"
12. Click on Ok
13. If Error in Web.xml:

open web.xml Change java to Java , version to 3.0 in <web-app> xml tag

Save web.xml
14. Run as Maven Install
15. Right click the project → Run as → Run on server


```
Myappninea
  Deployment Descriptor: Myapp
  src/main/java
  Maven Dependencies
  JRE System Library [J2SE-1.5]
  Deployed Resources
  src
    main
      java
        com
          example
            Myappninea
              config
              controller
                HomeController.java
      webapp
        resources
        WEB-INF
          views
            First.jsp
            home.jsp
            Second.jsp
          web.xml
        home.jsp
  target
  pom.xml
```

HomeController.java:

```
package com.example.Myappninea.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class HomeController {
    @RequestMapping("/user1")
    public String show() {
        return "First";
    }

    @RequestMapping("/user2")
    public String display() {
        return "Second";
    }
}
```

home.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Home</title>
</head>
<body>
    <h1>Multiple Views!</h1>
    <a href="user1">First Page!</a><br>
    <a href="user2">Second Page!</a>
</body>
</html>
```

First,jsp:

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>First Page</title>
</head>
<body>
    <h1>This is First Page!</h1>
    <a href="home.jsp">Back to Home</a>
</body>
</html>
```

Second.jsp:

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <title>Second Page</title>
</head>
<body>
   <h1>This is Second Page!</h1>
   <a href="home.jsp">Back to Home</a>
</body>
</html>
```

Week 10:

10a) Create a simple example of a hibernate application using eclipse IDE.

Steps:

1. **Create Maven Project**
2. **Check the first checkbox->Next**
3. **GroupId:com.hiber**
4. **ArtifactId:Hiber**
5. **Right-Click on project -> New->Class ->select main to create the main class, similarly create a persistence class without main method.**
6. **Add dependencies in pom.xml**
7. **To run-> run as java application**

HiberMain.java:

```
package Hiberdemo;
import java.util.List;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
public class HiberMain {

        @SuppressWarnings("deprecation")
        public static void main(String[] args) {
```

```
		// TODO Auto-generated method stub
		Configuration con=new
Configuration().configure().addAnnotatedClass(HiberPersistance.class);
		SessionFactory sft=con.buildSessionFactory();
		Session ses=sft.openSession();
		Transaction ts=ses.beginTransaction();
		HiberPersistance e1=new HiberPersistance();
		e1.setId(1);
		e1.setName("Sumayya");
		ses.save(e1);
		ts.commit();
	List<HiberPersistance> l = ses.createQuery("from HiberPersistance",
HiberPersistance.class).list();
		for(HiberPersistance s : l)
		System.out.println(s.getName());
		ses.close();
		sft.close();
		}

	}
```

HiberPersistence.java:

```
package Hiberdemo;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Entity
@Table(name="HiberPersistance")
public class HiberPersistance {
	@Id
	@GeneratedValue(strategy=GenerationType.IDENTITY)
	private int id;
	private String name;
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}

}
```

Pom.xml:

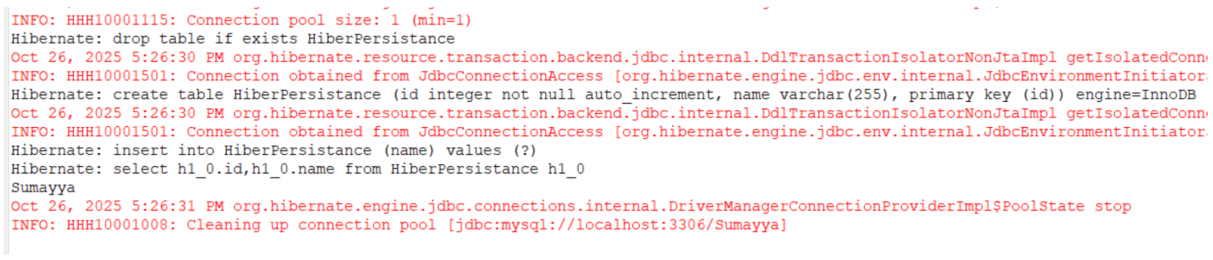
```
<project xmlns="http://maven.apache.org/POM/4.0.0"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
   <modelVersion>4.0.0</modelVersion>

   <groupId>com.hiber</groupId>
   <artifactId>Hiberdemo</artifactId>
   <version>0.0.1-SNAPSHOT</version>

   <dependencies>
      <dependency>
         <groupId>junit</groupId>
         <artifactId>junit</artifactId>
         <version>3.8.1</version>
         <scope>test</scope>
      </dependency>

      <dependency>
         <groupId>org.hibernate.orm</groupId>
         <artifactId>hibernate-core</artifactId>
         <version>6.2.32.Final</version>
      </dependency>

      <dependency>
         <groupId>com.mysql</groupId>
         <artifactId>mysql-connector-j</artifactId>
         <version>8.1.0</version>
      </dependency>
   </dependencies>
```

```
</project>
```

hibernate.cfg.xml

```
<!DOCTYPE hibernate-configuration PUBLIC  "-//Hibernate/Hibernate
Configuration DTD 3.0//EN"
        "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <!-- JDBC Database connection settings -->
        <property
name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property
name="connection.url">jdbc:mysql://localhost:3306/Sumayya</property>
        <property name="connection.username">root</property>
        <property name="connection.password">root</property>
        <!-- JDBC connection pool settings ... using built-in test pool -->
        <property name="connection.pool_size">1</property>
        <!-- Echo the SQL to stdout -->
        <property name="show_sql">true</property>
        <!-- Set the current session context -->
        <property name="current_session_context_class">thread</property>
        <!-- Drop and re-create the database schema on startup -->
        <property name="hbm2ddl.auto"> create </property>
        <!-- dbcp connection pool configuration -->

        <mapping class="Hiberdemo.HiberPersistance"/>
    </session-factory>
</hibernate-configuration>
```

OUTPUT:



```
INFO: HHH10001115: Connection pool size: 1 (min=1)
Hibernate: drop table if exists HiberPersistance
Oct 26, 2025 5:26:30 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConn
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator
Hibernate: create table HiberPersistance (id integer not null auto_increment, name varchar(255), primary key (id)) engine=InnoDB
Oct 26, 2025 5:26:30 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConn
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator
Hibernate: insert into HiberPersistance (name) values (?)
Hibernate: select h1_0.id,h1_0.name from HiberPersistance h1_0
Sumayya
Oct 26, 2025 5:26:31 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/Sumayya]
```

10b) Create an application to demonstrate Hibernate Query Language.

UpdateClass,java

```
package tenb;

import java.util.List;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import org.hibernate.query.Query;


public class UpdateClass {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Configuration con=new
Configuration().configure().addAnnotatedClass(emp.class);
		SessionFactory sft=con.buildSessionFactory();
		Session ses=sft.openSession();
		Transaction ts=ses.beginTransaction();
		emp e2=new emp();
		e2.setId(3);
		e2.setName("abc");
		ses.save(e2);
		String hql="UPDATE emp set name = :n  WHERE id = :i ";
		@SuppressWarnings("deprecation")
		Query q=ses.createQuery(hql);
		q.setParameter("n", "ITB-FSD");
		q.setParameter("i",3);
		int result=q.executeUpdate();
		ts.commit();
		Transaction ts1=ses.beginTransaction();
		System.out.println("Rows affected: "+result);
		List<emp> l=ses.createQuery("from emp",emp.class).list();
		for(emp e : l)
		System.out.println(e.getId()+"\t"+e.getName());
		ts1.commit();
		ses.close();
		sft.close();
```

```
                }

        }
```

emp.java

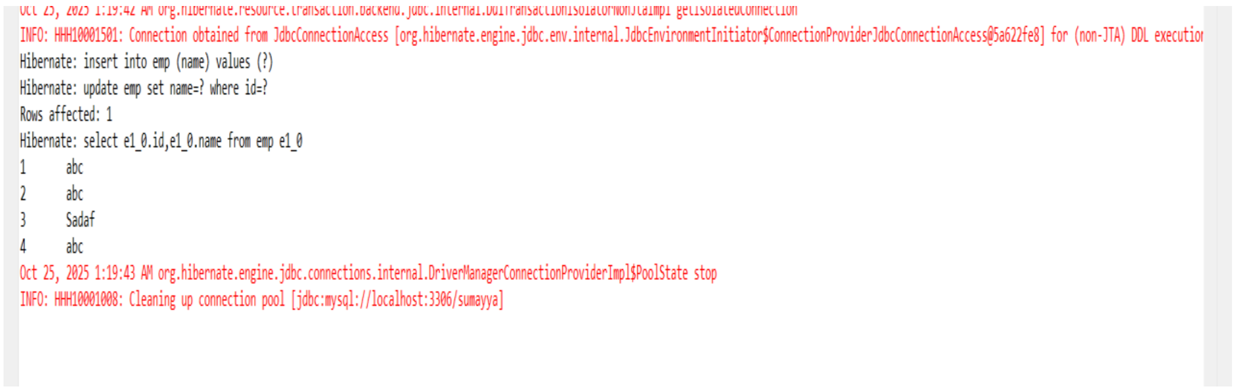
```
package tenb;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Entity
@Table(name="emp")
public class emp {
        @Id
        @GeneratedValue(strategy=GenerationType.IDENTITY)
        private int id;
        private String name;
        public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }
        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }

}
```

pom.xml: Same as in 10a

hibernate.cfg.xml

```
<!DOCTYPE hibernate-configuration PUBLIC  "-//Hibernate/Hibernate
Configuration DTD 3.0//EN"
        "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <!-- JDBC Database connection settings -->
        <property
name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property
name="connection.url">jdbc:mysql://localhost:3306/sumayya</property>
        <property name="connection.username">root</property>
        <property name="connection.password">root</property>
        <!-- JDBC connection pool settings ... using built-in test pool -->
        <property name="connection.pool_size">1</property>
        <!-- Echo the SQL to stdout -->
        <property name="show_sql">true</property>
        <!-- Set the current session context -->
        <property name="current_session_context_class">thread</property>
        <!-- Drop and re-create the database schema on startup -->
        <property name="hbm2ddl.auto"> update </property>
        <!-- dbcp connection pool configuration -->

        <mapping class="tenb.emp"/>
    </session-factory>
</hibernate-configuration>
```

OUTPUT:



```
Oct 25, 2025 1:19:42 AM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@5a622fe8] for (non-JTA) DDL executio
Hibernate: insert into emp (name) values (?)
Hibernate: update emp set name=? where id=?
Rows affected: 1
Hibernate: select e1_0.id,e1_0.name from emp e1_0
1       abc
2       abc
3       Sadaf
4       abc
Oct 25, 2025 1:19:43 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/sumayya]
```