

WT ASSIGNMENT-2

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content

HTML Elements:

- <html>-This tag begins and ends at each and every web page.
- <head>- Tags placed within the head element are not directly displayed by web browsers.
- <title>-The words write between tags will be displayed at the top of a viewer's browser.
- <body>-The body element will encapsulate all of your webpage's viewable content.

CSS

- CSS stands for Cascading Style Sheets
- Styles define how to display HTML elements
- External Style Sheets can save a lot of work
- External Style Sheets are stored in CSS files

There are three ways of inserting a style sheet:

- External style sheet
- Internal/Embedded style sheet

JavaScript is a high-level, dynamic **programming language** used to create interactive effects and functionalities in web browsers. It's one of the **core technologies of the web**, alongside **HTML** and **CSS**

Key Features:

- **Client-side scripting:** Runs directly in the browser without needing a server round-trip.
- **Event-driven:** Reacts to user interactions (like clicks, mouseovers, etc.).
- **Asynchronous capabilities:** Using features like `setTimeout`, `fetch`, `async/await`, and `Promises`.
- **Dynamically typed:** No need to declare variable types.
- **Inline style**

CODE: index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>OLP - Online Learning Platform</title>
  <link rel="stylesheet" href="styles.css" />
</head>
<body>

  <!-- Login Page -->
  <section id="login" class="page">
    <h2>Login</h2>
    <form id="login-form">
      <label>Email:</label>
      <input type="email" id="login-email" required />
      <label>Password:</label>
      <input type="password" id="login-password" required />
      <button type="submit">Login</button>
    </form>
    <p>Don't have an account? <a href="#" onclick="showPage('signup')">Sign Up</a></p>
  </section>

  <!-- Signup Page -->
  <section id="signup" class="page" style="display: none;">
    <h2>Sign Up</h2>
    <form id="signup-form">
      <label>Email:</label>
      <input type="email" id="signup-email" required />
      <label>Password:</label>
      <input type="password" id="signup-password" required />
    </form>
  </section>
</body>
```

```
<button type="submit">Sign Up</button>
</form>
<p>Already have an account? <a href="#" onclick="showPage('login')">Login</a></p>
</section>
```

```
<!-- Navigation -->
<header id="main-header" style="display: none;">
    <h1>OLP</h1>
    <nav>
        <a href="#" onclick="showPage('home')">Home</a>
        <a href="#" onclick="showPage('upload')">Upload Course</a>
        <a href="#" onclick="showPage('courses')">View Courses</a>
        <a href="#" onclick="logout()">Logout</a>
    </nav>
</header>
```

```
<!-- Home Page -->
<section id="home" class="page" style="display: none;">
    <h2>Welcome to OLP</h2>
    <p>Upload and view course PDFs for free!</p>
    <button onclick="showPage('upload')">Upload Course</button>
    <button onclick="showPage('courses')">Browse Courses</button>
</section>
```

```
<!-- Upload Course Page -->
<section id="upload" class="page" style="display: none;">
    <h2>Upload a New Course</h2>
    <form id="upload-form">
        <label>Course Title:</label>
        <input type="text" id="course-title" required />
```

```

<label>Select PDF File:</label>
<input type="file" id="course-pdf" accept="application/pdf" required />
<button type="submit">Upload</button>
</form>
</section>

<!-- View Courses Page -->
<section id="courses" class="page" style="display: none;">
<h2>Available Courses</h2>
<div id="course-list"></div>
</section>

<script>
// Optional: uncomment this line while developing to always start from login
// localStorage.removeItem("loggedIn");

// Show specific page
function showPage(pageId) {
  document.querySelectorAll('.page').forEach(page => {
    page.style.display = 'none';
  });
  document.getElementById(pageId).style.display = 'block';
}

// Logout function
function logout() {
  localStorage.removeItem("loggedIn");
  document.getElementById("main-header").style.display = "none";
  showPage("login");
}

```

```
// On Load
if (localStorage.getItem("loggedIn") === "true") {
    document.getElementById("main-header").style.display = "block";
    showPage("home");
} else {
    showPage("login");
}

// Signup
document.getElementById("signup-form").addEventListener("submit", async function (e)
{
    e.preventDefault();
    const email = document.getElementById("signup-email").value;
    const password = document.getElementById("signup-password").value;

    try {
        const res = await fetch("http://localhost:5000/api/users/signup", {
            method: "POST",
            headers: { "Content-Type": "application/json" },
            body: JSON.stringify({ email, password })
        });

        const data = await res.json();
        if (!res.ok) throw new Error(data.message);

        alert("Signup successful! Please login.");
        showPage("login");
    } catch (err) {
        // alert("Signup failed: " + err.message);
    }
})
```

```
    } catch (err) {
      console.error("Signup error:", err); // Add this
      res.status(500).json({ message: 'Signup error', error: err });
    }
  });

// Login (updated to POST to backend)
document.getElementById("login-form").addEventListener("submit", async function (e) {
  e.preventDefault();
  const email = document.getElementById("login-email").value;
  const password = document.getElementById("login-password").value;

  try {
    const res = await fetch("http://localhost:5000/api/users/login", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ email, password })
    });

    const data = await res.json();
    if (!res.ok) throw new Error(data.message);

    alert("Login successful!");
    localStorage.setItem("loggedIn", "true");
    document.getElementById("main-header").style.display = "block";
    showPage("home");
  } catch (err) {
    alert("Login failed: " + err.message);
  }
});
```

```
// Upload course

    document.getElementById("upload-form").addEventListener("submit", async function
(e) {
    e.preventDefault();

    const title = document.getElementById("course-title").value;
    const file = document.getElementById("course-pdf").files[0];

    const formData = new FormData();
    formData.append("title", title);
    formData.append("pdf", file);

    try {
        const res = await fetch("http://localhost:5000/api/courses/upload", { // ✅ correct route
            method: "POST",
            body: formData,
        });

        const data = await res.json();
        if (!res.ok) throw new Error(data.message);

        alert("Course uploaded successfully!");
        document.getElementById("upload-form").reset();
        showPage("courses"); // optional
        displayCourses(); // refresh
    } catch (err) {
        alert("Upload failed: " + err.message);
    }
});

// Display courses from backend
```

```
async function displayCourses() {
  try {
    const res = await fetch("http://localhost:5000/api/courses");
    const courses = await res.json();
    const courseList = document.getElementById("course-list");
    courseList.innerHTML = "";

    if (courses.length === 0) {
      courseList.innerHTML = "<p>No courses available.</p>";
      return;
    }

    courses.forEach((course) => {
      const courseDiv = document.createElement("div");
      courseDiv.innerHTML = `
        <h3>$ {course.title}</h3>
        <a href="http://localhost:5000/$ {course.pdfPath}" target="_blank"
download>Download PDF</a>
        <hr>
      `;
      courseList.appendChild(courseDiv);
    });
  } catch (error) {
    console.error("Error fetching courses:", error);
    document.getElementById("course-list").innerHTML = "<p>Error loading
courses.</p>";
  }
}

//  

async function deleteCourse(id) {
```

```
try {  
  const res = await fetch(`http://localhost:5000/api/courses/${id}`, {  
    method: "DELETE"  
  });  
  const data = await res.json();  
  
  alert("Course deleted");  
  displayCourses() // Refresh the list  
} catch (err) {  
  console.error("Delete error:", err);  
  alert("Failed to delete course");  
}  
}  
  
// On load  
displayCourses();  
</script>
```

```
<style>  
body {  
  font-family: Arial, sans-serif;  
  text-align: center;  
  background: url('img.jpg');  
  background-size: cover;  
  margin: 0;  
  padding: 0;  
}  
header {  
  background-color: #3e5976;  
  color: white;
```

```
padding: 15px;  
}  
  
nav a {  
margin: 0 10px;  
color: white;  
text-decoration: none;  
font-weight: bold;  
}  
  
.page {  
padding: 20px;  
}  
  
form {  
display: inline-block;  
background: rgba(255, 255, 255, 0.9);  
padding: 20px;  
border-radius: 15px;  
box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);  
}  
  
input, button {  
display: block;  
margin: 10px 0;  
padding: 10px;  
width: 100%;  
}  
  
button {  
background-color: #3e5976;  
color: white;  
border: none;  
cursor: pointer;  
}
```

```
button:hover {  
    background-color: #0056b3;  
}  
</style>  
  
</body>  
</html>  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8" />  
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>  
    <title>OLP - Online Learning Platform</title>  
    <link rel="stylesheet" href="styles.css" />  
</head>  
<body>  
  
    <!-- Login Page -->  
    <section id="login" class="page">  
        <h2>Login</h2>  
        <form id="login-form">  
            <label>Email:</label>  
            <input type="email" id="login-email" required />  
            <label>Password:</label>  
            <input type="password" id="login-password" required />  
            <button type="submit">Login</button>  
        </form>  
        <p>Don't have an account? <a href="#" onclick="showPage('signup')">Sign Up</a></p>  
    </section>
```

```
<!-- Signup Page -->

<section id="signup" class="page" style="display: none;">
  <h2>Sign Up</h2>
  <form id="signup-form">
    <label>Email:</label>
    <input type="email" id="signup-email" required />
    <label>Password:</label>
    <input type="password" id="signup-password" required />
    <button type="submit">Sign Up</button>
  </form>
  <p>Already have an account? <a href="#" onclick="showPage('login')">Login</a></p>
</section>
```

```
<!-- Navigation -->

<header id="main-header" style="display: none;">
  <h1>OLP</h1>
  <nav>
    <a href="#" onclick="showPage('home')">Home</a>
    <a href="#" onclick="showPage('upload')">Upload Course</a>
    <a href="#" onclick="showPage('courses')">View Courses</a>
    <a href="#" onclick="logout()">Logout</a>
  </nav>
</header>
```

```
<!-- Home Page -->

<section id="home" class="page" style="display: none;">
  <h2>Welcome to OLP</h2>
  <p>Upload and view course PDFs for free!</p>
  <button onclick="showPage('upload')">Upload Course</button>
```

```

<button onclick="showPage('courses')">Browse Courses</button>
</section>

<!-- Upload Course Page -->
<section id="upload" class="page" style="display: none;">
  <h2>Upload a New Course</h2>
  <form id="upload-form">
    <label>Course Title:</label>
    <input type="text" id="course-title" required />
    <label>Select PDF File:</label>
    <input type="file" id="course-pdf" accept="application/pdf" required />
    <button type="submit">Upload</button>
  </form>
</section>

<!-- View Courses Page -->
<section id="courses" class="page" style="display: none;">
  <h2>Available Courses</h2>
  <div id="course-list"></div>
</section>

<script>
  // Optional: uncomment this line while developing to always start from login
  // localStorage.removeItem("loggedIn");

  // Show specific page
  function showPage(pageId) {
    document.querySelectorAll('.page').forEach(page => {
      page.style.display = 'none';
    });
  }

```

```
document.getElementById(pageId).style.display = 'block';
}

// Logout function
function logout() {
    localStorage.removeItem("loggedIn");
    document.getElementById("main-header").style.display = "none";
    showPage("login");
}

// On Load
if (localStorage.getItem("loggedIn") === "true") {
    document.getElementById("main-header").style.display = "block";
    showPage("home");
} else {
    showPage("login");
}

// Signup
document.getElementById("signup-form").addEventListener("submit", async function (e) {
    e.preventDefault();
    const email = document.getElementById("signup-email").value;
    const password = document.getElementById("signup-password").value;

    try {
        const res = await fetch("http://localhost:5000/api/users/signup", {
            method: "POST",
            headers: { "Content-Type": "application/json" },
            body: JSON.stringify({ email, password })
        });
    }
})
```

```
const data = await res.json();
if (!res.ok) throw new Error(data.message);

alert("Signup successful! Please login.");
showPage("login");
//} catch (err) {
// alert("Signup failed: " + err.message);
//}

} catch (err) {
console.error("Signup error:", err); // Add this
res.status(500).json({ message: 'Signup error', error: err });
}

});
```



```
// Login (updated to POST to backend)
document.getElementById("login-form").addEventListener("submit", async function (e) {
e.preventDefault();

const email = document.getElementById("login-email").value;
const password = document.getElementById("login-password").value;

try {
const res = await fetch("http://localhost:5000/api/users/login", {
method: "POST",
headers: { "Content-Type": "application/json" },
body: JSON.stringify({ email, password })
});

const data = await res.json();
if (!res.ok) throw new Error(data.message);
```

```
        alert("Login successful!");

        localStorage.setItem("loggedIn", "true");
        document.getElementById("main-header").style.display = "block";
        showPage("home");

    } catch (err) {
        alert("Login failed: " + err.message);
    }
});

// Upload course
document.getElementById("upload-form").addEventListener("submit", async function (e) {
    e.preventDefault();

    const title = document.getElementById("course-title").value;
    const file = document.getElementById("course-pdf").files[0];

    const formData = new FormData();
    formData.append("title", title);
    formData.append("pdf", file);

    try {
        const res = await fetch("http://localhost:5000/api/courses/upload", { // ✅ correct route
            method: "POST",
            body: formData,
        });

        const data = await res.json();
        if (!res.ok) throw new Error(data.message);

        alert("Course uploaded successfully!");
    }
});
```

```
document.getElementById("upload-form").reset();
showPage("courses"); // optional
displayCourses(); // refresh
} catch (err) {
  alert("Upload failed: " + err.message);
}
});

// Display courses from backend
async function displayCourses() {
  try {
    const res = await fetch("http://localhost:5000/api/courses");
    const courses = await res.json();
    const courseList = document.getElementById("course-list");
    courseList.innerHTML = "";

    if (courses.length === 0) {
      courseList.innerHTML = "<p>No courses available.</p>";
      return;
    }

    courses.forEach((course) => {
      const courseDiv = document.createElement("div");
      courseDiv.innerHTML = `
        <h3>${course.title}</h3>
        <a href="http://localhost:5000/${course.pdfPath}" target="_blank"
download>Download PDF</a>
        <hr>
      `;
      courseList.appendChild(courseDiv);
    });
  }
}
```

```
        } catch (error) {
            console.error("Error fetching courses:", error);
            document.getElementById("course-list").innerHTML = "<p>Error loading
courses.</p>";
        }
    }

// On load
displayCourses();
</script>

<style>
body {
    font-family: Arial, sans-serif;
    text-align: center;
}
```

```
background: url('img.jpg');
background-size: cover;
margin: 0;
padding: 0;
}

header {
background-color: #3e5976;
color: white;
padding: 15px;
}

nav a {
margin: 0 10px;
color: white;
text-decoration: none;
font-weight: bold;
}

.page {
padding: 20px;
}

form {
display: inline-block;
background: rgba(255, 255, 255, 0.9);
padding: 20px;
border-radius: 15px;
box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
}

input, button {
display: block;
margin: 10px 0;
padding: 10px;
```

```
width: 100%;  
}  
  
button {  
background-color: #3e5976;  
color: white;  
border: none;  
cursor: pointer;  
}  
  
button:hover {  
background-color: #0056b3;  
}  
  
</style>
```

```
</body>
```

```
</html>
```

Code: server.js

```
require('dotenv').config();  
  
const express = require('express');  
  
const mongoose = require('mongoose');  
  
const cors = require('cors');  
  
const path = require('path');  
  
const app = express();  
  
// Middleware  
app.use(cors());  
app.use(express.json());  
  
// Serve uploaded PDFs
```

```
app.use('/uploads', express.static(path.join(__dirname, 'uploads')));

// Serve frontend static files
app.use(express.static(path.join(__dirname, 'frontend')));

// API Routes
const courseRoutes = require('./routes/courses');
const authRoutes = require('./routes/auth');
const userRoutes = require('./routes/users');

app.use('/api/courses', courseRoutes);
app.use('/api/auth', authRoutes);
app.use('/api/users', userRoutes);

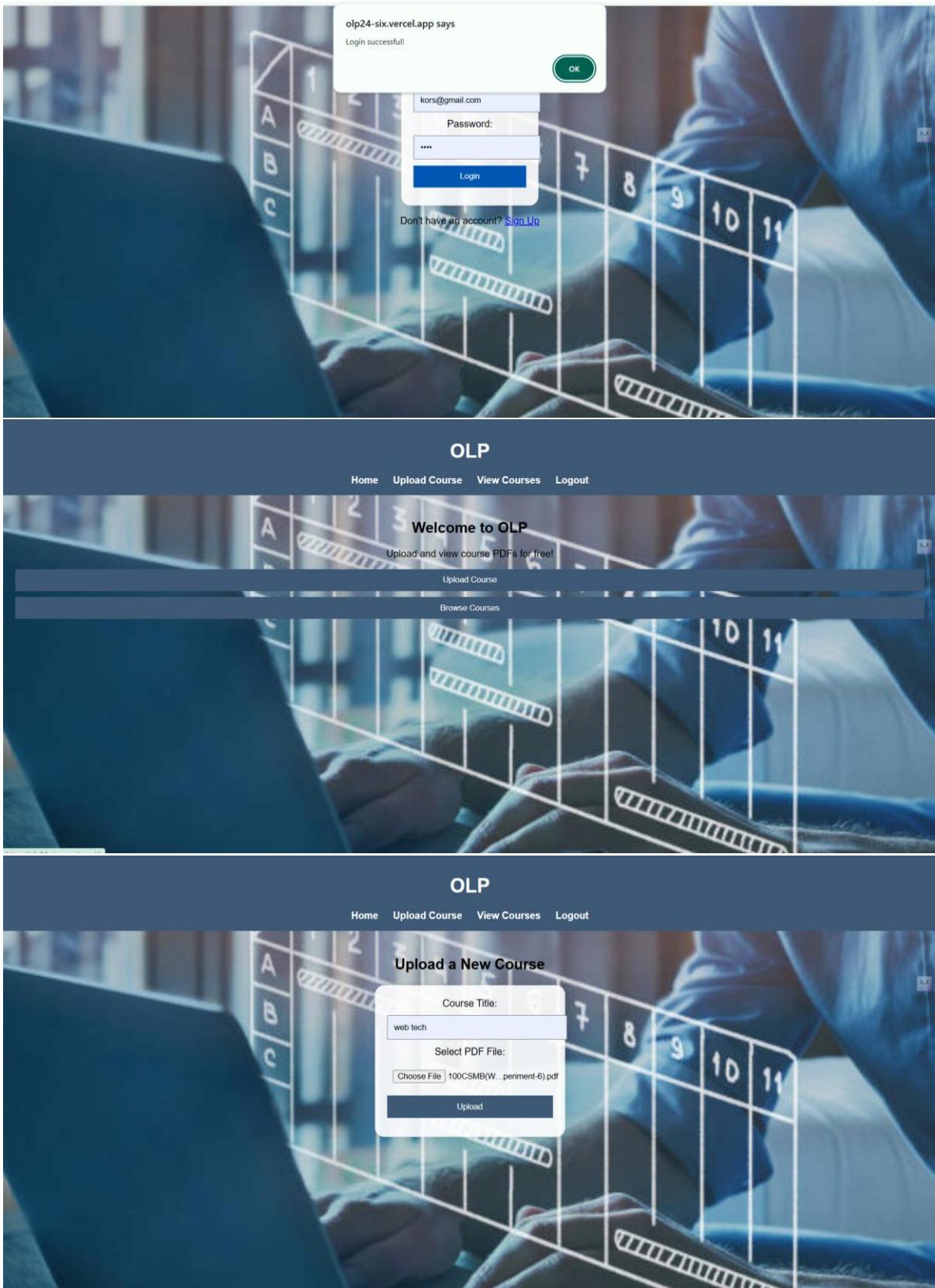
// Default route → serve index.html
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'frontend', 'index.html'));
});

// Connect MongoDB
mongoose.connect(process.env.MONGO_URI, {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
  .then(() => console.log("✅ MongoDB connected"))
  .catch(err => console.error("❌ MongoDB error:", err));

// Start server
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => {
```

```
console.log(`🚀 Server running at http://localhost:${PORT}`);
});
```

OUTPUT:



OLP

Home Upload Course View Courses Logout

Available Courses

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Available Courses																									
web tech																									
Download PDF																									
Delete																									
Available Courses																									
web tech																									
Download PDF																									
Delete																									
Available Courses																									
web tech																									
Download PDF																									
Delete																									
Available Courses																									
web tech																									
Download PDF																									
Delete																									

http://olp24-six.vercel.app/#

