Week 6: Industrial Training Report

Overview

Week 6 of the industrial training at **Auribises Technologies Pvt. Ltd.** focused on **Frontend Web Development**, emphasizing user interface (UI) enhancement using **HTML, CSS, and JavaScript**.

Students learned how to design, structure, and style web pages, and make them interactive with JavaScript. The week combined design principles with coding practice to help trainees create responsive, visually appealing, and dynamic websites.

By the end of the week, students could design fully functional front-end interfaces integrated with their AI-based backend systems from previous weeks.

Day 26: Introduction to Frontend and HTML Fundamentals

The twenty-sixth day introduced the **frontend development ecosystem** and the **role of HTML** in web design. Students learned about the structure of a webpage, semantic elements, and basic tags for text, media, and layout.

We created a simple static web page to understand the Document Object Model (DOM) hierarchy.

Topics Covered:

- Introduction to web architecture: frontend, backend, and database
- HTML document structure (`<!DOCTYPE html>`, `<html>`, `<head>`, `<body>`)
- Common tags: headings, paragraphs, links, lists, tables, images
- Forms and input fields
- Understanding the DOM structure

Example: Basic HTML Page

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
           <meta   name="viewport"   content="width=device-width,
initial-scale=1.0">
    <title>My First Web Page</title>
</head>
<body>
    <h1>Welcome to My Website</h1>
    <p>This is my first web page using HTML.</p>
    <img src="profile.jpg" alt="Profile Picture" width="200">
</body>
</html>
```

Day 27: Styling with CSS and Layout Design

Day 27 focused on **Cascading Style Sheets (CSS)** — the language for styling and visual design. Students learned how to apply inline, internal, and external CSS, and how to use selectors, colors, borders, and typography effectively.
We also explored **Flexbox** and **Grid layouts** for responsive web design.

Topics Covered:

- CSS syntax, selectors, and properties
- Color, padding, margin, border, and box model concepts
- Font styling and text effects
- Layout design using Flexbox and Grid
- Responsive design principles and media queries

Example: CSS Styling

```
body {
```

```
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 20px;
}

h1 {
    color: #0077cc;
    text-align: center;
}

.container {
    display: flex;
    justify-content: space-around;
}
```

Example: HTML + CSS Integration

```
<link rel="stylesheet" href="style.css">
<div class="container">
    <div class="card">Profile</div>
    <div class="card">Projects</div>
    <div class="card">Contact</div>
</div>
```

Day 28: Introduction to JavaScript and DOM Manipulation

On the twenty-eighth day, students were introduced to **JavaScript (JS)** — the scripting language used to make web pages interactive.
They learned how to handle user events, modify HTML elements dynamically, and use the **DOM API** for runtime updates.

Topics Covered:

- JavaScript syntax and variables

- Functions, loops, and conditionals
- DOM manipulation using `getElementById` and `querySelector`
- Event handling: `onclick`, `onchange`, and `addEventListener`
- Alert messages, prompts, and console debugging

Example: Button Click Event

```
<button onclick="showMessage()">Click Me</button>
<p id="message"></p>

<script>
function showMessage() {
        document.getElementById("message").innerText = "Hello,
JavaScript World!";
}
</script>
```

Day 29: Interactive Elements and Form Validation

Day 29 emphasized **JavaScript interactivity**, **form validation**, and **dynamic content rendering**.
Students worked on login and registration forms, implementing field validation and user feedback mechanisms. The session also covered DOM traversal and inline editing.

Topics Covered:

- Form handling and user input validation
- Conditional rendering using JS
- Real-time input checks and error messages
- Dynamic element creation and deletion

- Basic animations using CSS and JS

Example: Form Validation Script

```
<form onsubmit="return validateForm()">
          <input  type="text"  id="username"  placeholder="Enter
Username">
       <input  type="password"  id="password"  placeholder="Enter
Password">
    <button type="submit">Login</button>
</form>

<script>
function validateForm() {
    let user = document.getElementById("username").value;
    let pass = document.getElementById("password").value;
    if (user === "" || pass === "") {
        alert("All fields are required!");
        return false;
    }
    alert("Login successful!");
    return true;
}
</script>
```

Day 30: Mini Project – Personal Portfolio Website

The final day of Week 6 was dedicated to a **hands-on mini project**: creating a **personal portfolio website** using HTML, CSS, and JavaScript.

Students designed a responsive webpage showcasing their profile, skills, training projects, and contact information.

The goal was to integrate all learned frontend concepts into a polished, real-world product.

Topics Covered:

- Project: Personal Portfolio Website
- HTML structure with multiple sections (About, Projects, Contact)
- Responsive navigation bar and image gallery
- JavaScript-based form handling
- Testing and presentation of completed web pages

Example: Portfolio Structure

```
index.html
style.css
script.js
images/
```

Example: Sample Sections

```
<section id="about">
    <h2>About Me</h2>
     <p>I am a passionate web developer trained in Python and AI
integration.</p>
</section>
<section id="projects">
    <h2>Projects</h2>
    <ul>
        <li>ClarifAI Chatbot</li>
        <li>Daily Diary App</li>
    </ul>
</section>
```

Summary

Week 6 provided a strong foundation in **frontend web development**, covering **HTML for structure**, **CSS for design**, and **JavaScript for interactivity**.
Through practical exercises and the **Portfolio Project**, students learned how to build engaging, user-friendly web pages and integrate them with AI-powered backends.

This week bridged the gap between backend logic and user experience design, empowering trainees to create full-stack solutions ready for professional deployment.