IARE

INSTITUTE OF
AERONAUTICAL ENGINEERING

Course Title - Web System Engineering

Topic Title – Introduction to Web Applications

Presenter's Name - **M ArunKumar**
Presenter's ID - **IARE11151**
Department Name - **CSE (Artificial Intelligence, and Machine Learning)**
Lecture Number - 1
Presentation Date - 30/01/2025

Introduction to Web Applications

A **web application** is a software program that runs on a web server and is accessed through a web browser over the internet or an intranet. Unlike traditional desktop applications, web applications do not need to be downloaded or installed; users interact with them through a URL using a web browser like Chrome, Firefox, or Safari.

Common Examples:

- Gmail (email service)
- Facebook (social networking)
- Google Docs (online document editing)
- Online banking systems
- E-commerce platforms like Amazon

Technologies Used:

- **Front-end (client-side):** HTML, CSS, JavaScript
- **Back-end (server-side):** Python, PHP, Node.js, Java, Ruby, etc.
- **Databases:** MySQL, PostgreSQL, MongoDB

What is the Internet?

The **Internet** is a global network of interconnected computers that communicate using standard protocols (mainly **TCP/IP**). It allows devices around the world to share information and services such as websites, emails, and files.

Intranet:

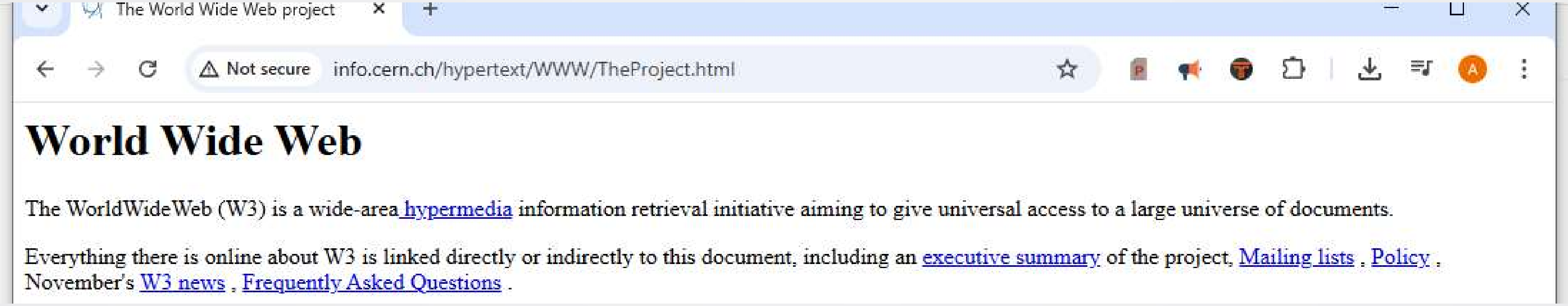
An **intranet** is a **private network** used within an organization to securely share information, collaborate, and manage internal communications and systems. Unlike the public **internet**, an intranet is **restricted to authorized users**, typically employees of the organization.

Difference from Internet and Extranet:

Network Type	Access Scope	Users
Internet	Global, public	Anyone
Intranet	Internal, private	Employees/authorized staff

Brief History:

- **Invented** by: Sir **Tim Berners-Lee** in **1989** at CERN.
- **First website: http://info.cern.ch (still accessible today).**



The World Wide Web project
Not secure
info.cern.ch/hypertext/WWW/TheProject.html

World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

Difference from WWW and Internet:

World Wide Web (WWW)	Internet
A **service** running on the internet	The **global network** of interconnected devices
Made of **websites and webpages**	Includes **email, FTP, VoIP, and more**
Accessed via **browsers** like Chrome or Firefox	Accessed through various **protocols** and tools

Web Clients; Web Servers

Feature	Web Client	Web Server
Definition	A software (like a browser) that requests web data from a server	A computer/system that stores and serves web content to clients
Main Role	Sends requests and displays responses (e.g., webpages)	Receives requests and sends back the appropriate data or page
Examples	Google Chrome, Mozilla Firefox, Safari, Postman	Apache, Nginx, Microsoft IIS, Node.js servers
Initiates Request?	✅ Yes	❌ No – It only responds to requests
Responds to Request?	❌ No	✅ Yes
Data Type Handled	HTML, CSS, JS, images, etc. (displayed content)	Files, scripts, database queries, APIs
User Interaction	Direct interaction with users (UI)	Usually no direct interaction (backend only)
Location	Runs on a user's device (PC, phone, etc.)	Runs on a remote or cloud-based server

Static and Dynamic Web Page

Feature	Static Web Page	Dynamic Web Page
Content Type	Fixed	Changes based on logic or data
Technologies Used	HTML, CSS	HTML + Server-side (PHP, Node.js, etc.), JavaScript
Personalization	❌ Not personalized	✅ Personalized (e.g., user login)
Speed	Fast	Slightly slower (due to data processing)
Hosting Cost	Low	Moderate to High
Database Interaction	❌ No	✅ Yes

What is Web Design?

Web design is the process of **planning and creating** the **visual layout, user experience (UX), and user interface (UI)** of a website.

Aspect	Description
Layout	How content is arranged (grids, columns, spacing)
Color Scheme	Consistent use of colors for branding and readability
Typography	Choice of fonts and sizes for readability
Navigation	Menus, links, and buttons for moving around the site
Responsiveness	Site adapts to different screen sizes (mobile-friendly)
Accessibility	Ensuring people with disabilities can use the site

Basics of HTML5 and web design

- Text Editors: VS Code, Sublime Text
- Browsers: Chrome, Firefox (with DevTools)
- Design Tools: Figma, Adobe XD, Canva
- Frameworks: Bootstrap, Tailwind CSS
- Version Control: Git + GitHub

Basic HTML.html

🔍 Basic tags list:

<!DOCTYPE html>: Declares the document as HTML5.

<html>: Root element of the page.

<head>: Contains metadata, title, links to CSS, etc.

<body>: The visible content of the webpage.

<h1> to <h6>: Headings.

<p>: Paragraph.

Other common elements: <a>, <img>, <div>, <span>, <ul>, <table>, etc.…

creating tables, lists, HTML forms.

Tag	Purpose
<table>	Starts the table
<tr>	Table row
<th>	Table header cell (bold & center)
<td>	Table data cell
border="1"	Adds a visible border (basic use)

M02CreatingTable.html

creating **lists in HTML**.

1. ✅ **Ordered List** (<ol>) – numbered list
2. ✅ **Unordered List** (<ul>) – bulleted list
3. ✅ **Description List** (<dl>) – term/definition list

```
<h3>Steps to Start a Blog</h3>
<ol>
  <li>Choose a topic</li>
  <li>Pick a domain name</li>
  <li>Set up hosting</li>
  <li>Write your first post</li>
</ol>
```

```
<h3>Grocery List</h3>
<ul>
  <li>Milk</li>
  <li>Eggs</li>
  <li>Bread</li>
  <li>Fruits</li>
</ul>
```

```
<h3>HTML Tags</h3>
<dl>
  <dt>&lt;h1&gt;</dt>
  <dd>Main heading of a page</

  <dt>&lt;p&gt;</dt>
  <dd>Paragraph text</dd>

  <dt>&lt;a&gt;</dt>
  <dd>Creates a hyperlink</dd>
</dl>
```

HTML forms

An **HTML form** is used to collect **user input** and submit it to a server or process it with JavaScript.

```
<form action="/submit" method="post">
  <label for="name">Name:</label><br>
  <input type="text" id="name"
name="name" required><br><br>

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

  <input type="submit" value="Submit">
</form>
```

- action: Where to send the form data (URL)

- method: get (appends data to URL) or post (sends data securely)

Common Form Elements

Tag/Type	Purpose
<input>	Basic input field
type="text"	Single-line text input
type="email"	Validated email input
type="password"	Password input
type="radio"	Select one from a group
type="checkbox"	Select multiple options
type="submit"	Submit the form
<textarea>	Multi-line input
<select> / <option>	Drop-down list
<label>	Defines a label for an input

Styles and classes to your web pages

how to **add styles and classes to your web pages** using **CSS**.

What is CSS?

CSS (Cascading Style Sheets) is used to **style HTML elements** — such as setting colors, fonts, spacing, layout, and more.

You can apply styles in **3 ways**:

Method	**Where It Goes**	**Best For**
Inline	Inside the tag	Quick, one-off styling
Internal	In a <style> tag	Styling a single HTML file
External	Linked CSS file	Reusing styles across pages

Styles and classes to your web pages

What Are Classes in HTML?

A **class** is a name you assign to one or more HTML elements, and then style them using CSS.

```
<p class="highlight">This is a highlighted paragraph.</p>
```

```
.highlight {
  background-color: yellow;
  font-weight: bold;
  padding: 10px;
}
```

Styles and classes to your web pages

✅ A. Inline CSS (not recommended for large projects)

```
<h1 style="color: red; text-align: center;">Welcome!</h1>
```

✅ B. Internal CSS

Internal CSS style details.txt

Styles and classes to your web pages

✅ C. External CSS (Best practice)

1. <link rel="stylesheet" href="styles.css"> --- In the HTML

2. In style.css file

```
h1 {
  color: navy;
}

.button {
  background-color: black;
  color: white;
}
```

web page layouts with CSS

1. Common Web Page Layout Structure

```
+--------------------------+
|        Header          |
+-------+------------------+
| Nav  |     Main         |
| Bar  |   Content        |
+-------+------------------+
|        Footer         |
+--------------------------+
```

2. Using CSS to Create Layouts

✅ **A. Layout Using Flexbox**

✅ **B. Layout Using CSS Grid**

We'll use **CSS Flexbox** and **CSS Grid** — modern tools for responsive, clean layouts.

Layout using Flex Box.txt

Layout using Grid.txt

Summary of **Module I: Introduction to Web Application, HTML and CSS**:

- Web applications run on the Internet, Intranet, or the World Wide Web (WWW).
- The Internet is a global network, while an Intranet is a private internal network.
- Web pages can be **static** (fixed) or **dynamic** (interactive and data-driven).
- Web clients (like browsers) request content from web servers.
- Web servers respond to client requests by delivering HTML, CSS, or other files.
- HTML5 structures web content using semantic tags, tables, lists, and forms.
- Forms allow user input using elements like text boxes, checkboxes, and buttons.
- CSS styles HTML elements and supports reusable styles via classes and IDs.
- Flexbox and Grid help build responsive and organized web layouts.
- Responsive design with CSS3 and HTML5 ensures compatibility across devices.

Thank You!