The World Islamic Sciences and Education University

جامعة العلوم الاسلامية العالمية

Faculty of Information Technology

كلية تكنولوجيا المعلومات

Training

Title

Web Development

Company Name

جمعية الأردنية الحاسبات

Student Name

Ahmad Haitham Hasan Alsemren

Supervisor

Dr. Salah Alghyaline

Acknowledgements

I would like to express my sincere gratitude to all those who contributed to the development of my skills and knowledge. First, I would like to thank our supervisor, Dr. Salah Alghyaline.

I am also grateful to my peers, whose collaborative spirit and constructive feedback played a vital role in refining my work. Special thanks to the online communities and forums that provided invaluable resources, troubleshooting advice, and inspiration for implementing best practices in HTML, CSS, and (Bootstrap) and JS

Abstract

This training program focused on enhancing my skills in HTML, CSS, and, JS foundational technologies for web development. Over the course of the training, I gained a comprehensive understanding of how to structure web content using HTML, apply effective styling and layout techniques with CSS, and implement interactive features using JavaScript. The program emphasized hands-on learning, allowing me to work on practical projects that reinforced theoretical concepts. By collaborating with peers and receiving feedback from instructors, I developed problem-solving skills and a deeper appreciation for web accessibility and user experience design. Ultimately, this training has equipped me with the tools and confidence to create dynamic and responsive web applications, paving the way for further exploration in the field of web development.

Table of Contents:

CSS

1. **CSS -Introduction :**
   a. **Selectors:**
   b. **Properties and Values:**
   c. **Responsive Design:**
2. **Inline -CSS :**
3. **Embedded -CSS :**
4. **External -CSS :**
5. **Class and Id :**
   a. **Class**
   b. **Id**
6. **Colors in CSS :**
7. **Grouping Styles in CSS:**
8. **Comments :**
9. **CSS properties :**
10. **The assigned task:**

JS

1. **MixItUp JS**

List of Table:

List of Figure:

Introduction

Web development is the process of creating websites and web applications for the internet. It encompasses a range of tasks, from building simple static pages to complex web applications that interact with databases and perform various functions.

Key Components of Web Development

1. Frontend Development:

This is the part of web development that focuses on what users see and interact with in their browsers. It involves using technologies like:

- **HTML (Hyper Text Markup Language)**: The backbone of web content, providing structure.
- **CSS (Cascading Style Sheets)**: Used for styling and layout to enhance the visual appeal.
- **JavaScript**: A programming language that adds interactivity and dynamic behavior to websites.

2. Backend Development:

This involves server-side programming and focuses on the functionality that users don't see. It includes:

Server-side languages: Such as Python, Ruby, PHP, Java, and Node.js.

Databases: Systems like MySQL, PostgreSQL, and MongoDB store and manage data.

Server management: Configuring and maintaining the server that hosts the website.

3. Full-Stack Development:

Full-stack developers are proficient in both front-end and back-end technologies, allowing them to build complete web applications from start to finish.

Topics

1. **HTML (HyperText Markup Language).**
2. **CSS (Cascading Style Sheet).**
3. **JS (JavaScript) → (MixItUp JS,Overlay JS,AOS JS,swiper JS ,Cleave JS(Library))**

HTML

1. HTML- Introduction:

- Is the foundational language used to create and structure content on the web. It defines the structure of web pages by using a series of elements, or tags, that browsers interpret to display content.

- As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

- Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.
- Key Features of HTML:

a. Structure and Semantics:

HTML provides a way to organize content into a meaningful structure. Elements like headings (<h1>, <h2>), paragraphs ( ), lists (<ul>, <ol>), and sections (<section>, <article>) help convey the hierarchy and relationship of information.

b. Elements and Tags:

HTML is composed of elements, each defined by opening and closing tags. For example, a paragraph is marked up as Your text here . Some elements are self-closing, like <img> for images.

c. Attributes:

a. Id Attribute
b. title Attribute
c. class Attribute
d. style Attribute

Tags can have attributes that provide additional information. For instance, Link uses the href attribute to specify the URL the link points to.

d. Links and Multimedia:

HTML allows for the inclusion of hyperlinks (<a>), images (<img>), videos (<video>), and audio (<audio>), enabling rich multimedia experiences

- Basic HTML Document:

```
<html>
    <head>
        <title>This is document title</ title >
    </head>
    <body>
        <h1>This is heading</h1>
        <p>Document content …</p>
    </body>
</html>
```

2. HTML - Basic Tags:

- Heading Tags

  Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5>,** and **<h6>**.

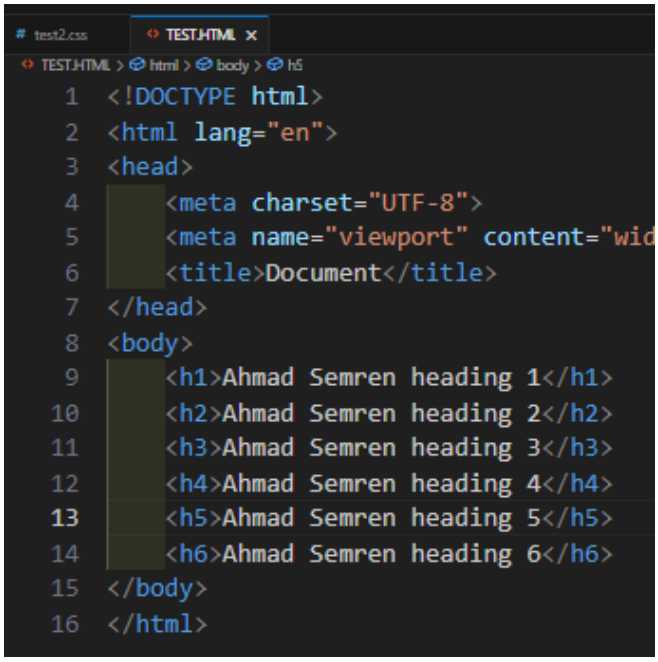


Figure 1 :

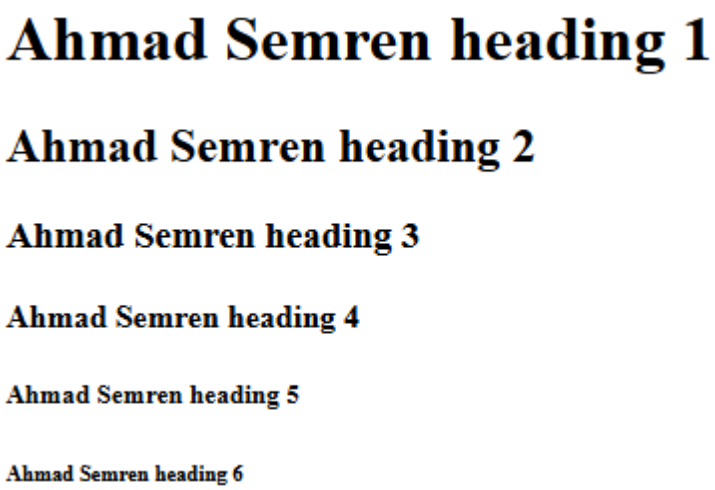


Figure 2 :

- Paragraph Tag
  The tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening and a closing tag

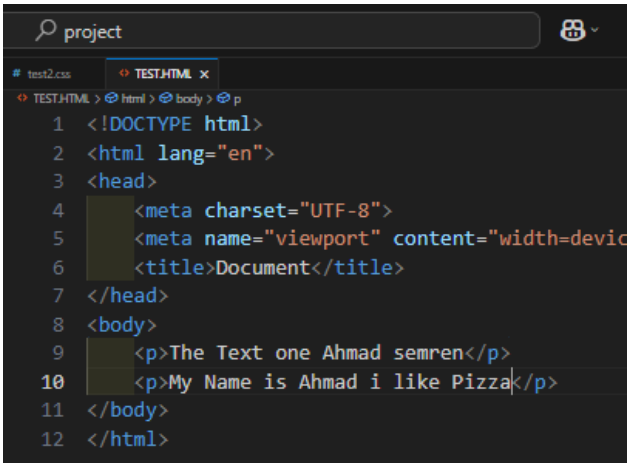


Figure 3 :

The Text one Ahmad semren

My Name is Ahmad i like Pizza

Figure 4:

- Line Break Tag
  Whenever you use the **
** element, anything following it starts from the next line. This tag is an example of an **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

- Horizontal Lines
  Horizontal lines are used to visually break-up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

- div Tag
  The HTML tag is used for defining a section of your document. With the div tag, you can group large sections of HTML elements together and format them with CSS.

- span Tag
  The HTML tag is used for grouping and applying styles to inline elements.

- Difference between the span tag and the div tag the span tag is used with inline elements whilst the div tag is used with block-level content.

```
<div class="end-donation">
    <h1>Raise Funds For poor</h1>
    <div class="w3-orange" style="height:7px;width:60%; margin-bottom: 30px;"></div>
    <h5>Raised:</h5>
    <span class="Raised"> $400.00</span>
    <h5>Goal:</h5>
    <span class="Goal"> $650.00</span>
</div>
```

Figure 5 :

3. HTML – Text-Formatting:

- Bold Text
  Anything that appears within ... element, is displayed in bold.

- Italic Text
  Anything that appears within ... element is displayed in italicized.

- Underlined Text
  Anything that appears within <u>...</u> element, is displayed with underline.

- Strike Text
  Anything that appears within <strike>...</strike> element is displayed with strikethrough, which is a thin line through.

- Superscript Text
  The content of a ^{...} element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

- Subscript Text
  The content of a _{...} element is written in subscript; the font size used is the same as the characters surrounding it but is displayed half a character's height beneath the other characters.

- Inserted Text
  Anything that appears within <ins>...</ins> element is displayed as inserted text.

- Deleted Text
  Anything that appears within ~~...~~ element, is displayed as deleted text.

- Larger Text
  The content of the <big>...</big> element is displayed one font size larger than the rest of the text surrounding it.

- Smaller Text
  The content of the <small>...</small> element is displayed one font size smaller than the rest of the text surrounding it.

- Comments
  HTML comments are placed in between <!-- ... --> tags.

```
<body>
    <p>Wise <b>University</b> </p>
    <p>Wise <i>University</i> </p>
    <p>Wise <u>University</u> </p>
    <p>Wise <strike>University</strike> </p>
    <p>Wise <sup>University</sup> </p>
    <p>Wise <sub>University</sub> </p>
    <p>Wise <ins>University</ins> </p>
    <p>Wise <del>University</del> </p>
    <p>Wise <big>University</big> </p>
    <p>Wise <small>University</small> </p>
    <!--Text Formatting  -->
</body>
```

Figure 6 :

Wise **University**

Wise *University*

Wise University

Wise ~~University~~

Wise ^{University}

Wise $_{University}$

Wise University

Wise ~~University~~

Wise University

Wise University

Figure 7 :

4. HTML – Images:

- You can insert any image in your web page by using <img> tag. Following is the simple syntax to use this tag.
-

5. HTML – Tables:

- tables are created using the tag <table>…</table>.
- <tr>tag is used to create table rows.
- <td>tag is used to create data cells.
- <th> tag is used to create table heads.
- Colspan to merge two or more columns.
- Rowspan to merge two or more rows.

6. HTML – Lists:

- An unordered list.
- An ordered list.
- To insert data in the list.

```
<img src="image/Real-Madrid-Logo.png" alt="" width="400" height="200" >
<table border="3" | width="200px">
    <tr>
        <th>Name</th>
        <th>Salary</th>
    </tr>
    <tr>
        <td>Vini</td>
        <td>1B$</td>
    </tr>
    <tr>
        <td>Valvarde</td>
        <td>500M$</td>
    </tr>
    <tr>
        <td colspan="3">Avg =</td>
    </tr>
</table>
</body>
</html>
```

Figure 8 :

Name	Salary
Vini	1B$
Valvarde	500M$
Avg =	

Figure 9 :

7. HTML - Text Links:

- The text between the opening <a> and closing tags is what users will see and click on.
- Go to google
- To open a link in a new tab, you can use the target attribute with the value _blank.
- Example: Go to google

8. HTML – Forms:

- <Form> …</Form> be inside the < body > It is used at the beginning of the form It also contains action and attributes.
- < input type = "…" > Be inside the < body > And This is one of the most important elements in the form And it has many types.

Table 1 :

<input type="text">	Displays a single-line text input field
<input type=" email ">	Displays a Text Email
<input type=" number ">	Displays a type of <text> that requires a number
<input type="password">	Displays a Text password
<input type="date">	Displays a input fields that let the user enter a date
<input type="file">	Displays a file select control for uploading a file
<select> <option >…</option> </select>	Displays a list of items a user can select from
<input type="radio">	Displays a radio button
<input type="checkbox">	Displays a checkbox
<input type="submit">	Displays a clickable button to submit value
<input type="reset">	Displays a clickable button to reset value

```html
<form action="#">
    <input type="text" placeholder="Type Your Name..."><br><br>
    <input type="number" placeholder="Type Your Age..."><br><br>
    <input type="email" placeholder="Type Your Email..."><br><br>
    <input type="password" placeholder="Type Your Password..."><br><br>
    <input type="date"><input type="file"><br><br>
    <select id = "Country">
        <option value="jordan">Jordan</option>
        <option value="usa">USA</option>
        <option value="serbia">Serbia</option>
    </select>
    <label>Male
        <input type="radio" name="same" value="male">
    </label>
    <label>Female
        <input type="radio" name="same" value="female">
    </label><br><br>
    <label>
        <input type="checkbox" name="chebox">
        I agree to the privacy policy
    </label><br><br>
    <input type="submit" value="OK">
    <input type="reset" value="RESET">
</form>
```

Figure 10:

Figure 11:

CSS

CSS (Cascading Style Sheets) is a stylesheet language used to control the visual presentation of HTML elements on a webpage. While HTML structures the content, CSS is responsible for making that content visually appealing by specifying styles such as colors, fonts, layouts, and spacing.
With CSS, developers can separate content from design, allowing for more flexibility in styling and ensuring that web pages are consistent in appearance across different devices and screen sizes. CSS also makes it easy to apply the same style rules across multiple pages, reducing redundancy and improving maintainability.

Key Features of CSS:

1. **Styling and Layout:**
   CSS defines how HTML elements should be displayed on the screen. It allows you to style text (e.g., color, font, size), add backgrounds, control spacing, and arrange elements in various layouts (e.g., grid).
2. **Selectors:**
   CSS works by using selectors to target HTML elements. These selectors can be based on element type, class, ID, or attributes.
   - **Element selector: p {color: blue;} (styles all <p> elements)**
   - **Class selector: .box {border: 1px solid black;} (styles elements with the class "box")**
   - **ID selector: #header {font-size: 24px;} (styles the element with the ID "header")**
3. **Box Model:**
   CSS operates on the "box model" concept, where every HTML element is considered a rectangular box. The box model consists of four areas: content, padding, border, and margin. CSS controls the size, spacing, and layout of these areas.
4. **Responsive Design:**
   CSS plays a crucial role in creating responsive websites, meaning the design adapts to different screen sizes. Using media queries, developers can apply specific styles depending on the screen width, height, or device type.

5. **Positioning:**
   CSS allows you to control the position of elements using properties such as position, float, display, and z-index. This helps in creating complex page layouts.

Inline CSS

Inline CSS applies styles directly to individual HTML elements using the style attribute within the opening tag. This method allows you to style specific elements without affecting other parts of the page.

```
6       <p style="color: red; font-size: 18px;">This text is red and larger.</p>
```

Figure 12:

Advantages:

- Quick and easy to apply for individual elements.
- Useful for overriding other styles in certain cases.

Disadvantages:

- Makes the HTML code messy and harder to maintain.
- Not ideal for large websites where consistent styling is required.
- Styles can't be reused across multiple elements or pages.

Embedded (Internal) CSS

Embedded CSS (also called internal CSS) is placed inside the <style> tag within the <head> section of an HTML document. It allows you to style multiple elements on the same page using selectors, but the styles apply only to that page.

```
4       <style>
5           p {
6               color: blue;
7               font-size: 16px;
8           }
9           h1 {
10              color: green;
11          }
12      </style>
13  </head>
14  <body>
15      <p >This text is blue.</p>
16      <h1>This text is green.</h1>
```

Figure 13:

This text is blue.

This text is green.

Figure 14:

Advantages**:**

- Allows you to style multiple elements on the same page with one block of CSS.
- Keeps HTML and CSS in the same file for easier editing when working on small projects.

Disadvantages**:**

- Styles only apply to a single page, making it less efficient for larger projects.
- Increases the size of the HTML file, potentially slowing downloading times.

External CSS

External CSS is written in a separate .css file, and the styles are linked to an HTML document using the <link> tag inside the <head> section. This method is best for large websites where styles need to be consistent across multiple pages.

```
5        <link rel="stylesheet" type="text/css" href="styles.css">
```

Figure 15:

```
1  p {
2      color: black;
3      font-size: 14px;
4  }
5  h1 {
6      color: navy;
7  }
```

Figure 16:

This text is black.

This text is navy.

Figure 17:

Advantages**:**

- Keeps the HTML and CSS separate, which improves organization and maintainability.
- Allows for consistent styling across multiple pages.
- Reduces the size of the HTML file, improving page load speed.
- Styles can be reused across an entire website by linking to the same stylesheet.

Disadvantages**:**

- Requires an additional HTTP request to load the CSS file, which can slightly delay page load times if not optimized.
- Changes in the CSS file will affect all linked pages, which may not always be desirable.

Class in CSS

A class is used to apply the same style to multiple HTML elements. You can assign the same class to different elements, allowing you to style groups of elements consistently.

- Syntax in HTML: Use the class attribute to assign a class to an element.
- Syntax in CSS: Use the dot (.) symbol followed by the class name to select the elements.
-

Key Points about Classes:

- Reusable: A class can be used on multiple elements across a page.
- Flexibility: Ideal when you want to style multiple elements in the same way.
- Naming Convention: Class names should be descriptive and meaningful but can be reused across multiple pages.

ID in CSS

An ID is used to uniquely identify a single HTML element. Unlike classes, an ID should only be used once per page for one specific element.

- Syntax in HTML: Use the id attribute to assign an ID to an element.
- Syntax in CSS: Use the hash (#) symbol followed by the ID name to select the element.

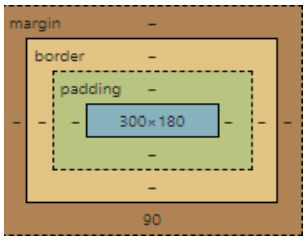
In this example, the main-text ID applies styles only to the specific paragraph with that ID.

Key Points about IDs:

- **Unique**: An ID must be unique within a page; no two elements should have the same ID.
- **Specificity**: IDs have higher specificity in CSS than classes, so they will override class styles if both are applied to the same element.
- **Use Case**: Best used for elements that need unique styles, like headers or specific sections of a page.

CSS properties :

Table 2 :

width		Defines the width of an element.
height		Defines the height of an element.
font-size		Specifies the size of the font for text within an element.
font-family		Specifies the typeface used for the text within an element.
font-weight		Specifies the thickness of the font for text within an element.
font-style		Specifies the style of the font for text within an element.
Padding		Specifies the space between the content of an element and its border.
margin		Specifies the outer space around an element.
border		Specifies the properties of the border around an element.
Position		Specifies how an element is positioned in the document.
border-radius		Specifies the rounding of the corners of an element.
direction		Specifies the text direction within an element.
background-color		Specifies the background color of an element.
background-image		Specifies a background image for an element.
background-attachment		Specifies how the background image is attached relative to the viewport or the element.
background-size		Specifies the size of the background image.
background-repeat		Specifies how the background image is repeated.

color	Specifies the color of the text within an element.
text-align	Specifies the alignment of text within an element.
text-decoration	Specifies decorations applied to text.
line-height	Specifies the height of a line box for text within an element.
cursor	Specifies the shape of the cursor when hovering over an element.
z-index	Specifies the stack order of elements along the Z-axis (depth) when they overlap.
hover:	used to define effects or changes on an element when the mouse pointer hovers over it.
transition: all 1s.	used to apply transition effects to all possible properties of an element over a duration of one second (1s).
clip-path	used to define a specific shape for an element by clipping it to that shape.
display	determines how elements are displayed on the page.
display: flex.	determine how space is distributed among flex items.
justify-content	determine how flex items are distributed along the main axis.
align-items	determine how flex items are aligned along the cross axis (vertical axis).
align-content	determine how the space between rows (or columns) is distributed when there is extra space on the cross axis.
flex-direction: row or column	determine the direction in which flex items are laid out.
flex-wrap	determine how flex items wrap when they cannot fit in a single line.
flex-grow	determine how much a flex item should grow to fill the available space.

display: block.	1.The element starts on a new line. 2. It takes up the full available width.
display: inline.	**1.**The element does not start on a new line **2.** You cannot set a width or height for the element directly.
display: inline block.	**1**.The element does not start on a new line **2**. The elements can have defined width and height.
display: grid.	You can define the number of rows and columns.

9. The Project 1:

Here's a shorter, focused introduction for your portfolio:
"Hi, I'm [SEMREN], a web developer specializing in [Selling video games,Programer]. I create responsive and user-friendly websites with a focus on clean code and modern design principles. I'm passionate about [specific area - e.g., front-end development, UI/UX, etc.] and always eager to take on new challenges."

```html
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.css">
    <link rel="stylesheet" href="test2.css">


    <title>Txr Portfolio Store</title>
</head>
<body>
        <header>
                <a herf = "#" class = "logo">txr</a>
                    <nav class = "navbar">
                        <a href="#services">services</a>
                        <a href="#my_cource">my Cource</a>
                        <a href="#contact">contact</a>
                    </nav>
        </header>
        <section class="main">
                    <div>
                        <p>Hello , My Name Is</p>
                        <h2><span>SIMREN</span></h2><br>
                        <h3>Gamer | Programmer | Online Video Game Seller</h3>
                        <a href="#services" class="main-btn">My Services</a>

                    <div class ="social-links">
                        <a href="#instagram"><i class="fa-brands fa-instagram"></i></a>
                        <a href="#discord"><i class="fa-brands fa-discord"></i></a>
                        <a href="#gethub"><i class="fa-brands fa-github"></i></i></a>
                        <a href="#steam"><i class="fa-brands fa-steam"></i></i></a>
                    </div>
        </section>
        <section class="cards" id="services">
                <h2 class = "title">Services</h2>
                <div class="content">
                <div class ="card">
                    <div class="icon">
                            <i class="fa-solid fa-gamepad"></i>
                    </div>
                    <div class = "info">
                        <h3>Cheapest Video Game</h3>
                            <p>You can buy the cheapest games and subscriptions from the txr store such as:</p>
                            <span style="color: rgb(131, 3, 3);">
                            <ul>
                                <li>Fortnite</li>
                                <li>CSGO 2 (item)</li>
                                <li>League Of Legends</li>
                            </ul>
                            </span>
                    </div>
                </div>
                <div class ="card">
                    <div class="icon">
                        <i class="fa-solid fa-user"></i>
                    </div>
                    <div class = "info">
                        <h3>Design and interface services (Front-End)</h3>
                            <p>Ensures that the website works purfectly on all devices <span style="color: rgb(131, 3, 3);">(mobile, tablet, computer)</span>
                            Tools like:<span style="color: rgb(131, 3, 3);"> Bootstrap or Tailwind CSS.</span>
                            Distinctive icons and fonts:
                            <span style="color: rgb(131, 3, 3);"> Icon library: Font Awesome, Iconify.</span>
                            Web fonts: <span style="color: rgb(97, 1, 75);">Google Fonts</span>
                            <span style="color: rgb(131, 3, 3);">Effects and animations:</span>

                            Libraries such as:<span style="color: rgb(131, 3, 3);"> Animate.css, or GSAP.</span></p>
                    </div>
                </div>
                </div>
        </section>
        <section class="cour" id="my_cource">
        <h2 class="title">My Cource</h2>
        <div class="contact">
            <div class ="card2">
                <div class="video-container">
            <div class="vid">
                <video autoplay loop muted >
                    <source src="video/txrlogo.mp4" type="video/mp4" >
                </video>
            </div>
            </div>
            <div class="info2" id="my_cource">
                <h2>My Cource About :</h2>
                <p>You can purchase the course from our store on Instagram.<br> It contains how to purchase video games and subscriptions</p>
            </div>
            <div class="btn2">
                <a href="#my_cource" class="main-btn2">Purchese Cource</a>
            </div>
        </div>
        </div>
        </section>
        <section class="contactsec" id="contact">
                <h2 class="title">Contact</h2>
                <div class="content">
                    <div class="card">
                    <div class="icon">
                        <i class="fa-solid fa-phone"></i>
                    </div>
                    <div class="info">
                        <h3>Phone</h3>
                        <p>+962799192303</p>
                    </div>
                    </div>
                    <div class="card">
                        <div class="icon">
                            <i class="fa-solid fa-envelope"></i>
                        </div>
                        <div class="info">
                            <h3>Mail</h3>
                            <p>ahmadsk1921@gmail.com</p>
                        </div>
                        </div>
                </div>
        </section>
        <footer class="footer">
            <p class="footer-title">Copyrights <span style="color: yellow;">@</span><span>Ahmad AlSemren</span></p>
            <div class ="social-links">
                <a href="#instagram"><i class="fa-brands fa-instagram"></i></a>
                <a href="#discord"><i class="fa-brands fa-discord"></i></a>
                <a href="#gethub"><i class="fa-brands fa-github"></i></i></a>
                <a href="#steam"><i class="fa brands fa-steam"></i></i></a>
            </div>
        </footer>
</body>
</html>
```

Html code : *Figure 18:*

Css code : *Figure 19:*

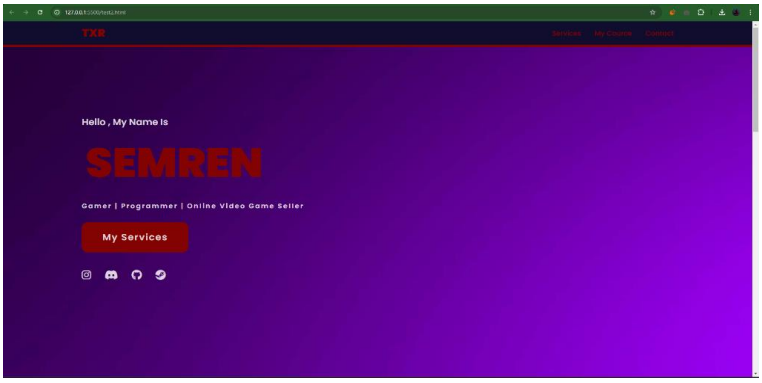


Figure 19:

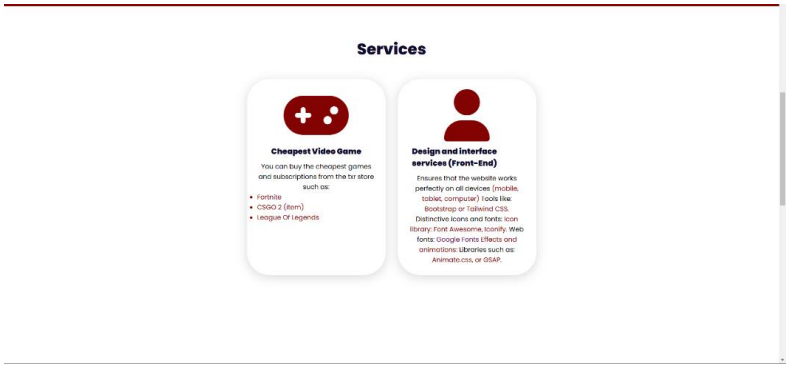


Figure 20:

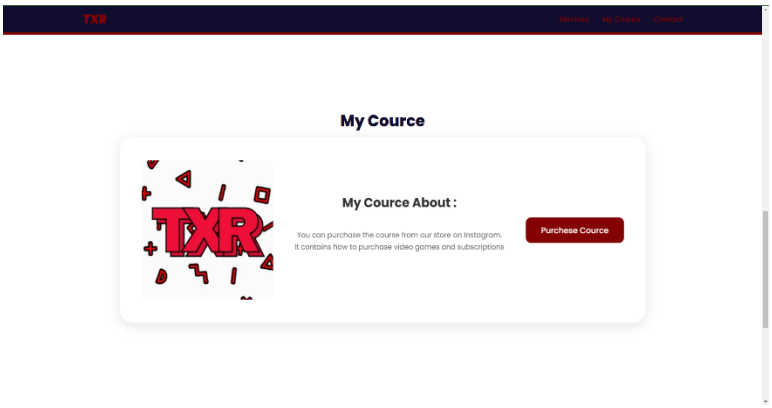


Figure 21:

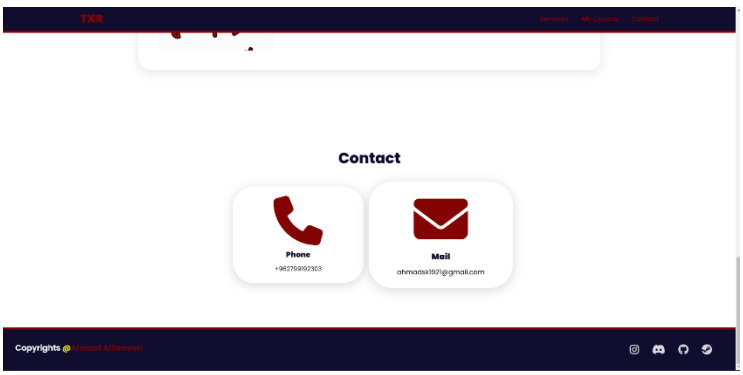


Figure 22:

Bootstrap

Introduction to Bootstrap

Bootstrap is a popular open-source front-end framework for developing responsive and mobile-first websites. It provides a collection of pre-designed CSS and JavaScript components that simplify web development.

Key Features:

- Responsive grid system
- Pre-designed components (buttons, modals, navbars, etc.)
- Customizable via Sass variables
- Extensive JavaScript plugins

Getting Started

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

    <title>Document</title>
</head>
<body>









    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Figure 23:

HOW TO GET THE NAVBAR FROM GET.BOOTSTRAP.COM

Figure 24:

```
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="#">Brand</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
            <li class="nav-item">
            <a class="nav-link active" href="#">Home</a>
            </li>
            <li class="nav-item">
            <a class="nav-link" href="#">Features</a>
            </li>
        </ul>
        </div>
    </nav>
```

Figure 25:

Utilities

Spacing:

- **Margin:** .m-1, .m-2, ...
- **Padding:** .p-1, .p-2, ...

Typography:

- Headings: <h1>, <h2>, ...
- Text utilities: .text-center, .text-muted, ...

Colors:

- Backgrounds: .bg-primary, .bg-success
- Text: .text-danger, .text-warning

Craeate form on Bootstrap:

```
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sh
</head>
<body>
    <form>
        <div class="mb-3">
        <label for="exampleInputEmail1" class="form-label">Email address</label>
        <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
        <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
        </div>
        <div class="mb-3">
        <label for="exampleInputPassword1" class="form-label">Password</label>
        <input type="password" class="form-control" id="exampleInputPassword1">
        </div>
        <div class="mb-3 form-check">
        <input type="checkbox" class="form-check-input" id="exampleCheck1">
        <label class="form-check-label" for="exampleCheck1">Check me out</label>
        </div>
        <button type="submit" class="btn btn-primary">Submit</button>
    </form>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrY
</body>
</html>
```

Figure 26:

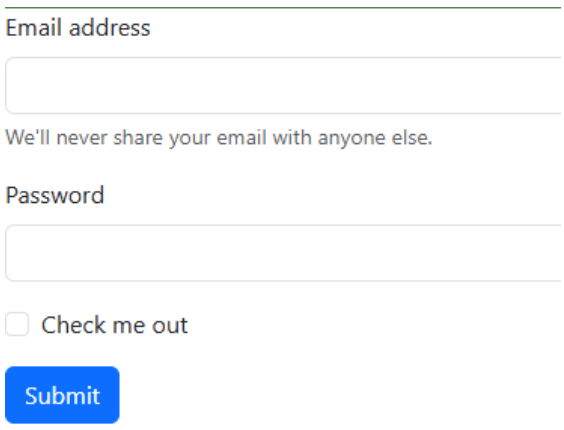


Figure 27:

Js

MIXITUP JS

MixItUp is a powerful, flexible, and easy-to-use JavaScript library for filtering and sorting content on web pages. It is commonly used for creating interactive galleries, portfolios, and dynamic lists.

Key Features:

- Filter content dynamically.
- Sort items by different criteria.
- Support for animations and transitions.
- Simple API for customization.

Including MixItUp:

You can include MixItUp in your project via:

CDN:

```
<script src="https://cdn.jsdelivr.net/npm/mixitup/dist/mixitup.min.js"></script>
```

Basic Usage

HTML Structure:

MixItUp requires specific data attributes to define filters and targets.

```
    <title>MixItUp Demo</title>
</head>
<body>
    <div class="container">
        <!-- Filter Buttons -->
        <div class="controls">
        <button type="button" class="filter" data-filter="all">All</button>
        <button type="button" class="filter" data-filter=".category-a">Category A</button>
        <button type="button" class="filter" data-filter=".category-b">Category B</button>
        </div>

        <!-- Items to Filter/Sort -->
        <div class="mix category-a">Item 1</div>
        <div class="mix category-b">Item 2</div>
        <div class="mix category-a">Item 3</div>
    </div>

    <!-- Scripts -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/mixitup/3.3.1/mixitup.min.js" integrity="sha512-nKZDK+ztK6Ug+2B6DZx+Qt
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lH
</body>
```

Figure 28:

This is the most basic implementation of MixItUp that includes:

1. Filter buttons at the top
2. Content items with different categories
3. Basic MixItUp initialization

```
<body>
    <!DOCTYPE html>
    <html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Simple MixItUp Example</title>
    </head>
    <body>

        <div>
            <button type="button" data-filter="all">Show All</button>
            <button type="button" data-filter=".category-1">Category 1</button>
            <button type="button" data-filter=".category-2">Category 2</button>
            <button type="button" data-filter=".category-3">Category 3</button>
        </div>

        <div class="container">
            <div class="mix category-1">Item 1 (Category 1)</div>
            <div class="mix category-2">Item 2 (Category 2)</div>
            <div class="mix category-3">Item 3 (Category 3)</div>
            <div class="mix category-1">Item 4 (Category 1)</div>
            <div class="mix category-2">Item 5 (Category 2)</div>
            <div class="mix category-3">Item 6 (Category 3)</div>
        </div>

        <script src="https://cdnjs.cloudflare.com/ajax/libs/mixitup/3.3.1/mixitup.min.js"></script>
        <script>

            var mixer = mixitup('.container');
        </script>
    </body>
    </html>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/mixitup/3.3.1/mixitup.min.js" integrity="sha512-nKZDK+ztK6Ug+2
</html>
```

Figure 29:

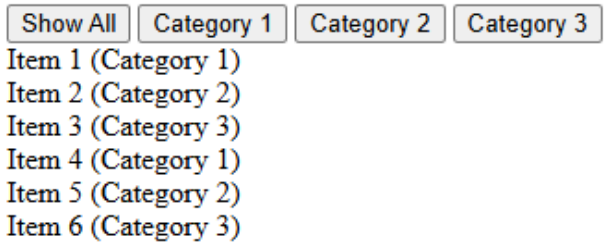


Figure 30:

Figure 31:

Figure 32:

Figure 33:

References :

https://elzero.org/ →html+css

https://cdnjs.com/libraries/ → for libraries

https://getbootstrap.com/ → Getting bootstrap

https://fontawesome.com/ → To get Fonts and icons

https://github.com/ → Help me look at people's projects

Thanks

Dr. Salah Alghyaline