1. Introduction

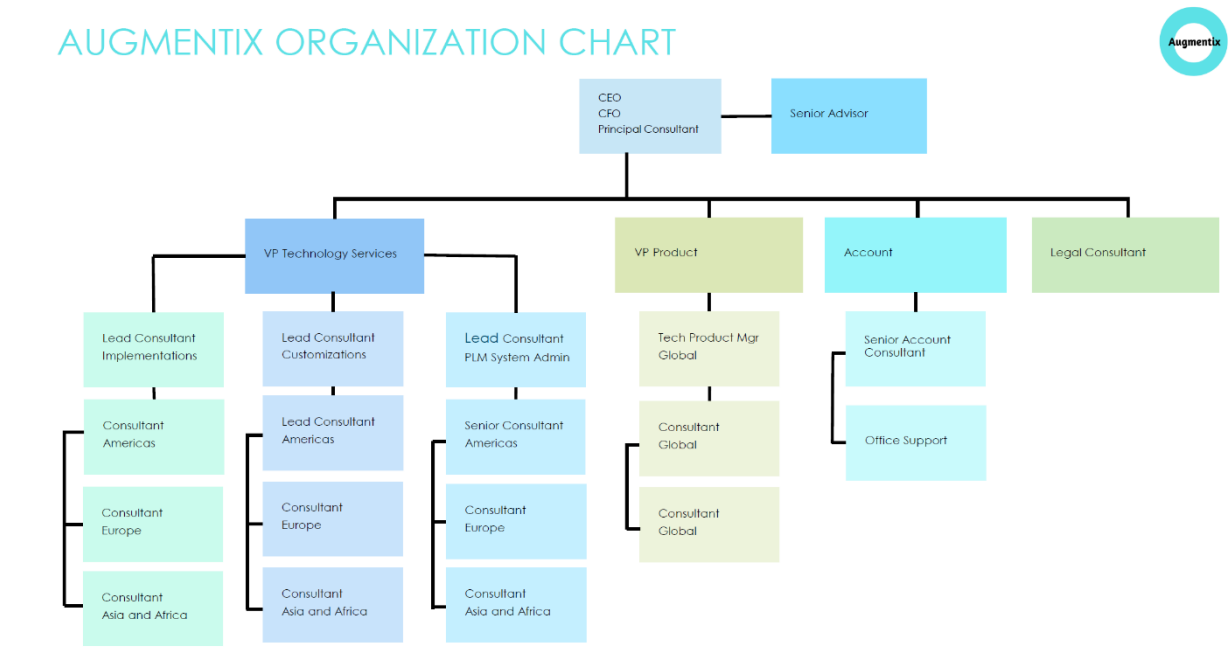
1.1 About Company:

Augmentix Pvt. Ltd.

Founded in 2020, Augmentix is a software solutions company that assists companies worldwide in improving design, manufacturing, and service processes for a smart, connected world. We provide innovative services across various domains, including Mechanical CAD (MCAD), Product Lifecycle Management (PLM), Application Lifecycle Management (ALM), Internet of Things (IoT), Augmented Reality (AR), Digital Twin, and Industry 4.0. Our offerings help customers unlock the value of digitization, extract more value from third-party applications, and optimize business functions such as engineering, software development, supply chain management, manufacturing, and service. By coordinating these processes, we enable product and service advantages for our clients in discrete manufacturing organizations.

Address: J-602, Astonia Royale, Near Sinhgad Road Mumbai-Bangalore Highway, Pune, Maharashtra, 410046

1.2 Organizational Structure



2. Services

2.1 Introduction to Services

Specialties: Business Consulting, Training Program, Software Solution, Process Consultant, Windchill Customization, Functional Consultant, System Administration, Business Administration, Test Engineer, CAD customization, CAD Designers, Product Developers, Migration, PLM implementation, Value Road Map.

2.2 Services offered by the company

Public relation firm

1. Business Consulting
2. Training Program
3. Software Solution
4. Process Consultant
5. Windchill Customization
6. Functional Consultant
7. System Administration
8. Business Administration
9. Test Engineer
10. CAD customization
11. CAD Designers
12. Product Developers
13. Migration
14. PLM implementation
15. Value Road Map

2.3 Number of employees: 16

3. Web development

3.1 Introduction to Web development:

Web development is the process of creating websites and web applications. It encompasses several different tasks, including web design, web content development, client-side/server-side scripting, and network security configuration, among others. Web development can be divided into two main categories: front-end development and back-end development.

1. **Front-end development:**
    - Html (Hyper Text Markup Language)
    - CSS (Cascading style sheets)
    - JavaScript

2. **Back-end development:**
    - Server side languages
    - Databases
    - Web servers

3. **Full-stack development**

4. **Web development tools and platforms:**
    - Text editors and IDEs
    - Version control systems
    - Build tools and task runners
    - Testing frameworks
    - Content management systems
    - Hosting and deployment services

Web development is a vast field with many specializations, and it's a critical component of the digital economy, enabling businesses and individuals to establish an online presence and deliver services over the internet.

3.2 Need of web development:

- Online presence
- Information dissemination
- E-commerce
- Communication and collaboration
- Accessibility

3.3 Significance of web development:

- User experience
- Brand identity
- Search engine optimization (SEO)
- Security
- Social performance

3.4 Use of web development:

- Business and e-commerce
- Content management
- Social media and networking
- Education and e-learning
- Entertainment and media
- Government and public services
- Healthcare and telemedicine

4. Front-end development

4.1 Introduction:

Front-end development, also known as client-side development, is the practice of producing HTML, CSS and JavaScript for a website or web application so that a user can see and interact with them directly the challenge associated with front-end development is the development of the websites or application visual layout and ensuring that it is user- friendly and responsive.

4.2 Technologies involved in Front-end development:

1. **Core technologies:**
   - HTML (Hypertext markup language)
   - CSS (cascading style sheets)
   - JavaScript

2. **Frameworks and libraries:**
   - React
   - Angular
   - Vue.js

3. **Tools and technologies:**
   - Package managers
   - Build tools
   - version control
   - Testing
   - CSS preprocessors
   - Responsive design

4.3 Use for front-end development

- User interface (ui) creation
- User experience(ux) enhancement
- Interactivity and dynamic content
- Accessibility
- Performance optimization

4.4 Need for front-end development

- Digital presence
- User engagement
- Mobile compatibility
- Competitive advantage
- SEO benefits

4.5 Significance of Front-end development:

- First impression
- User retention
- Conversion rates
- Technological advancements
- Cross-platform compatibility

5. HTML

5.1 Introduction:

HTML (Hyper Text Markup Language) is the standard markup language used to create the structure of web pages. It consists of a series of elements that tell the browser how to display content

5.2 Basics of html:

HTML documents are made up of tags, which are keywords surrounded by angle brackets `<>`. Most tags come in pairs: an opening tag `<tag>` and a closing tag `</tag>`. The content goes between these tags.

Example:

```
<!DOCTYPE html>
<html>
<head>
   <title>My First Web Page</title>
</head>
<body>
   <h1>Welcome to My Web Page</h1>
   <p>This is a paragraph of text on my web page.</p>
</body>
</html>
```

5.3 Common html elements:

1. <!DOCTYPE html>:Declares the document type and version of HTML (HTML5 in this case).
2. <html>: The root element of an HTML page.
3. <head>: Contains meta-information about the document, like its title and links to stylesheets.
4. <title>: Defines the title of the document, shown in the browser's title bar or tab.
5. <body>: Contains the content of the document, such as text, images, and other media.
6. <h1> to <h6>: Heading elements, with <h1> being the highest level.
7. <p>: Defines a paragraph.
8. <a>: Creates a hyperlink.
9. <img>: Embeds an image in the document.
10. : Defines a division or section in the document.
11. : Used to group inline elements for styling purposes.
12. and <ol>: Unordered and ordered lists.
13. : List items within <ul> or <ol>.
14. <table>: Defines a table.
15. <form>: Creates an HTML form for user input.
16. <input>: Used within forms to create various types of input fields.
17. <button>: Creates a clickable button.

5.4 Attributes:

HTML elements can have attributes, which provide additional information about the element. Attributes are always specified in the start tag and usually consist of a name and a value, separated by an equals sign (`=`).

Example:
<img src="image.jpg" alt="Description of the image">

- `src` is an attribute that specifies the source of the image.
- `alt` is an attribute that provides alternative text for the image

5.5 Formatting elements:

Formatting Elements
HTML provides various elements for formatting text:

- : Indicates strong importance.
- : Indicates emphasis.
- <mark>: Represents marked or highlighted text.
- <code>: Represents computer code.
- <sup>: Superscript text.
- <sub>: Subscript text.

5.6 Semantic elements:

HTML5 introduced semantic elements that clearly describe the content they contain:

- <header>: Represents a container for introductory content or a set of navigational links.
- <footer>: Represents a footer for a document or section.
- <nav>: Defines a set of navigation links.
- <article>: Represents an independent piece of content of a document, such as a blog post.
- <section>: Defines a section in a document.
- <aside>: Defines content aside from the page content, like a sidebar

5.7 Tables:

Tables are created using the `<table>` element, with `<tr>` for table rows, `<th>` for table headers, and `<td>` for table cells.

Example:

```
<table>
   <thead>
      <tr>
         <th>Header 1</th>
         <th>Header 2</th>
      </tr>
   </thead>
   <tbody>
        <tr>
         <td>Row 1, Cell 1</td>
         <td>Row 1, Cell 2</td>
```

```
        </tr>
        <tr>
            <td>Row 2, Cell 1</td>
            <td>Row 2, Cell 2</td>
        </tr>
    </tbody>
</table>
```

- **Table with Data**

```
<!DOCTYPE html>
<html>
<head>
    <title>Table Example</title>
</head>
<body>
    <table>
        <tr>
            <th>Name</th>
            <th>Age</th>
        </tr>
        <tr>
            <td>Alice</td>
            <td>30</td>
        </tr>
        <tr>
            <td>Bob</td>
            <td>25</td>
        </tr>
    </table>
</body>
</html>
```

5.8 Form:

Forms are created using the `<form>` element, with various form elements like `<input>`, `<textarea>`, and `<button>`.

Example:

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

- **Form with Input Fields**

```
<!DOCTYPE html>
<html>
<head>
  <title>Form Example</title>
</head>
<body>
  <form action="/submit" method="post">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required>
    <br>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required>
    <br>
    <button type="submit">Login</button>
  </form>
</body>
</html>
```

6. CSS

6.1 Basics of CSS:

CSS, or Cascading Style Sheets, is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML, or XHTML). CSS can be added to HTML documents to style the elements and control the document's layout.
Here are the basics and syntaxes of CSS:

1. **Selectors**: CSS selectors are used to select the HTML elements you want to style.

```
/* Select all paragraphs */
p {
   color: blue;
}

/* Select elements with the class 'special' */
. special {
    font-weight: bold;
}

/* Select elements with the id 'header' */
#header {
    background-color: yellow;
}
```

2. **Properties and Values**: Inside the curly braces, you define the properties and their values. A property is a characteristic you want to change (like color, font, size), and each property has a value that specifies how much or what should be changed.

```
/* Set the text color of paragraphs to red */
p {
   color: red;
}
```

3. **Comments**: CSS comments are written between `/* */` and are not rendered by the browser.

```
/* This is a comment and will be ignored by the browser */
```

4. **Units**: CSS uses various units to specify measurements for properties like width, height, font-size, etc.

```
/* Pixels */
div {
   width: 100px;
      }

/* Percentage */
```

```
section {
   width: 50%;
}

/* Em (relative to font-size of element) */
p {
   font-size: 1.5em;
}
```

5. **The Cascade**: CSS stands for Cascading Style Sheets. The cascade is a feature that allows multiple stylesheets to influence the rendering of a single document. Rules can come from the browser, an external stylesheet, or an inline style. The cascade determines which rule takes precedence.

6. **Inheritance**: Inheritance is a key feature in CSS where properties are passed from parent elements to their children. Some properties are inherited naturally, while others are not.

7. **Specificity**: When multiple CSS rules point to the same element, the most specific rule is applied. Specificity is a ranking system for determining which styles are applied to an element.

8. **The Box Model**: The box model is a concept that describes how elements are rendered on the page. It consists of the content area, padding, border, and margin.

```
div {
   width: 100px;
   padding: 10px;
   border: 5px solid black;
   margin: 20px;
}
```

9. **Pseudo-classes and Pseudo-elements**: Pseudo-classes and pseudo-elements are used to style elements under certain conditions or to create fictional elements.

```
/* Pseudo-class example (hover state of a link) */
a:hover {
   color: red;
}

/* Pseudo-element example (first line of a paragraph) */
p::first-line {
   font-weight: bold;
}
```

10. **Media Queries**: Media queries are used to apply different styles depending on the device's screen size, resolution, and other features.

```
@media screen and (max-width: 600px) {
   body {
      background-color: lightblue;
    }
 }
```

11. Grouping: You can group selectors to apply the same styles to multiple elements.

```
h1, h2, h3 {
    color: green;
}
```

12. Nesting (with preprocessors like SASS or LESS): Although not part of standard CSS, preprocessors allow nesting of selectors for more organized and readable code.

```
/* This is SASS (or SCSS) syntax and will be compiled to CSS */
nav {
    ul {
        padding: 0;
    }
    li {
        display: inline;
    }
    a {
        color: white;
        text-decoration: none;
    }
}
```

Remember that CSS is constantly evolving, and new features are being added regularly. It's important to keep up with the latest developments and best practices to write efficient and maintainable stylesheets.

CSS supports various color systems for specifying colors:

- Hexadecimal: `#RRGGBB` or `#RGB`
  color: #FF0000; /* Red */

- RGB: `rgb(red, green, blue)`
  color: rgb(255, 0, 0); /* Red */

- RGBA: `rgba(red, green, blue, alpha)` (alpha for opacity)
  background-color: rgba(255, 0, 0, 0.5); /* Semi-transparent red */

- HSL: `hsl(hue, saturation, lightness)`
  color: hsl(0, 100%, 50%); /* Red */

-HSLA: `hsla(hue, saturation, lightness, alpha)`
  background-color: hsla(0, 100%, 50%, 0.3); /* Semi-transparent red */

6.2 Text Properties:

CSS has properties for styling text:

- Font Family: `font-family: "Helvetica", sans-serif;`
- Font Size: `font-size: 16px;`
- Font Weight: `font-weight: bold;`
- Line Height: `line-height: 1.5;`

- Text Align: `text-align: center;`
- Color: `color: #333;`

6.3 Box Model:

The box model consists of:

- Margin: `margin: 10px;`
- Border: `border: 1px solid #000;`
- Padding: `padding: 20px;`
-Content: (defined by `width` and `height`)

6.4 Display Properties:

The `display` property specifies the display behavior of an element:

- `display: block;`
- `display: inline;`
- `display: inline-block;`
- `display: flex;`
- `display: grid;`
- `display: none;`

6.5 Units:

CSS uses different units for measurements:

- Absolute units: `px`, `cm`, `mm`, `in`, `pt`, `pc`
- Relative units: `em`, `rem`, `vw`, `vh`, `%`

6.6 Position:

The `position` property specifies the type of positioning method used for an element:

- `position: static;`
- `position: relative;`
- `position: absolute;`
- `position: fixed;`
- `position: sticky;`

6.7 Background:

CSS has properties for setting backgrounds:

- Background Color: `background-color: #fff;`
- Background Image: `background-image: url("image.jpg");`
- Background Repeat: `background-repeat: no-repeat;`
- Background Position: `background-position: center top;`
- Background Size: `background-size: cover;`

6.8 Flex Box:

Flexbox is a layout model that manages space distribution and alignment:

- Display: `display: flex;`
- Flex Direction: `flex-direction: row;` or `column;`
- Flex Wrap: `flex-wrap: wrap;`
- Justify Content: `justify-content: center;`
- Align Items: `align-items: center;`
- Align Content: `align-content: space-between;`

6.9 Transition:

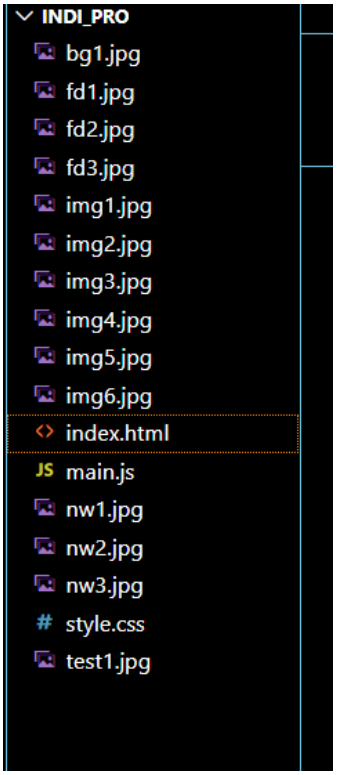
Transitions allow property changes to occur smoothly:

- Transition Property: `transition-property: background-color;`
- Transition Duration: `transition-duration: 0.5s;`
- Transition Timing Function: `transition-timing-function: ease-in;`
- Transition Delay: `transition-delay: 1s;`
- Shorthand: `transition: background-color 0.5s ease-in 1s;`

These are just a few examples of the many properties and values available in CSS. The syntax generally follows a pattern of `property: value;` within a rule set that targets elements with selectors. Remember that CSS is case-sensitive and that the order of properties within a rule set does not matter.

7. Individual project

For individual project we were told to make a simple website using HTML, CSS and JavaScript. I made a blogging website page which is called heart.out. In that we have the home page which contains the code and a as u scroll down u can see the blogs have been into into sub types which makes it easy to find.



This project includes:

HTML Files: index.html

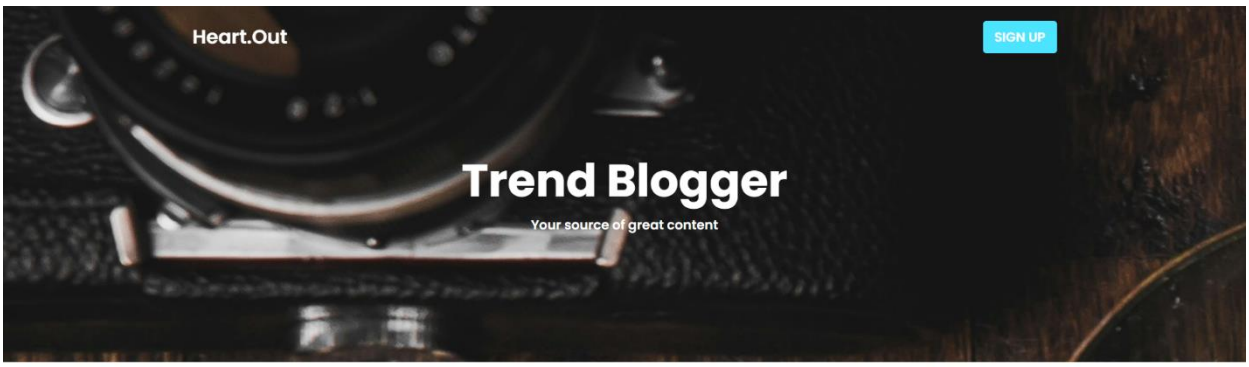
CSS Files: style.css

JavaScript Files: script.js

Media Files: Images

These files collectively define the structure, style, and functionality of the project, along with enhancing user experience through multimedia elements.

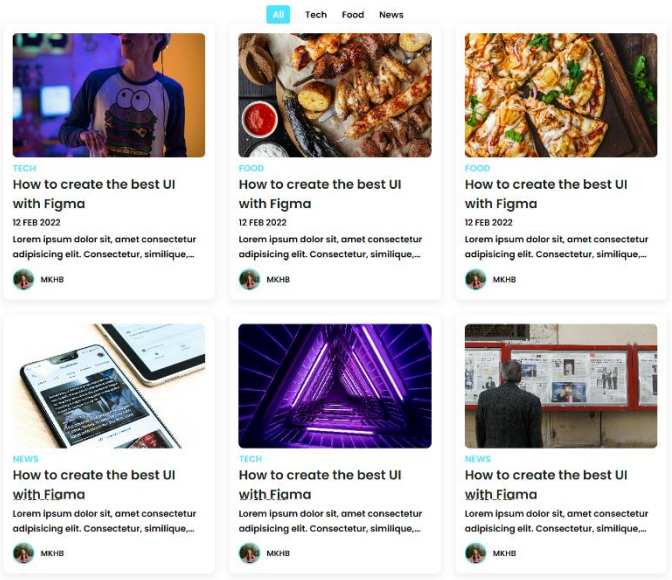
1. The website Home Page

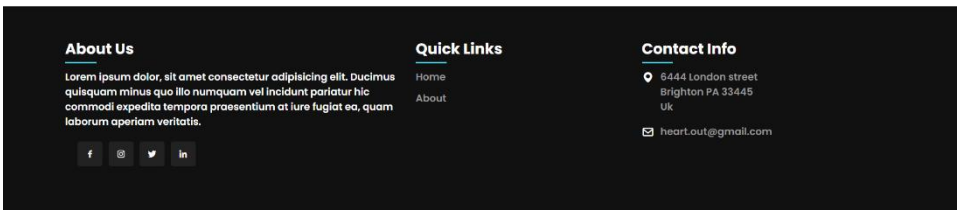


Catch up with the trending topics

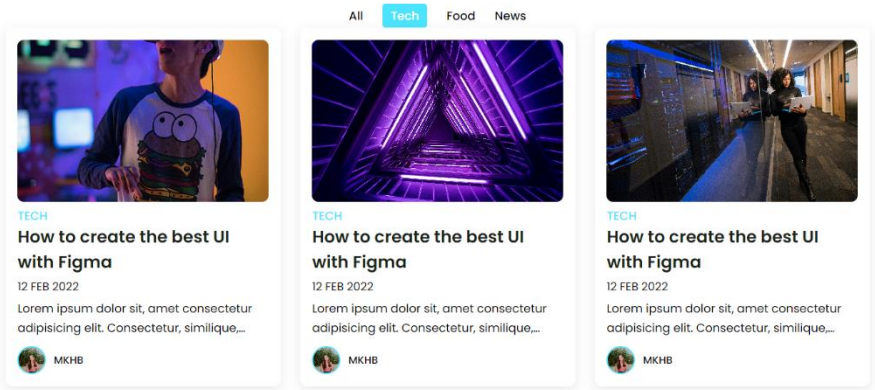
Lorem ipsum dolor sit amet consectetur adipisicing elit. Laborum eos consequuntur voluptate dolorum totam provident ducimus cupiditate dolore doloribus repellat. Saepe ad fugit similique quis quam. Odio suscipit incidunt distinctio.

2. About and contact info

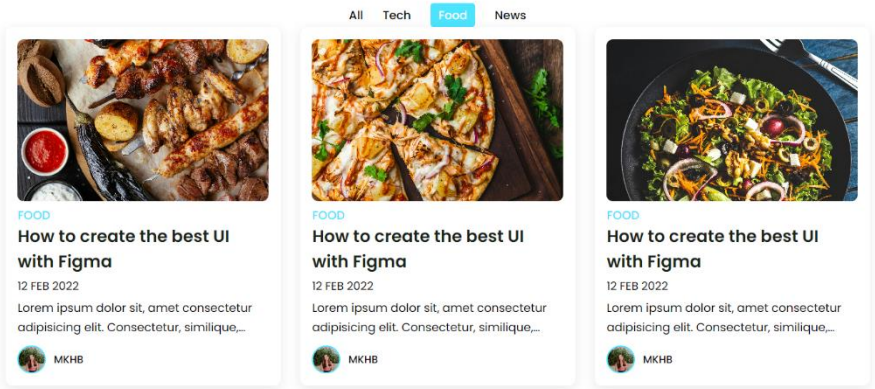




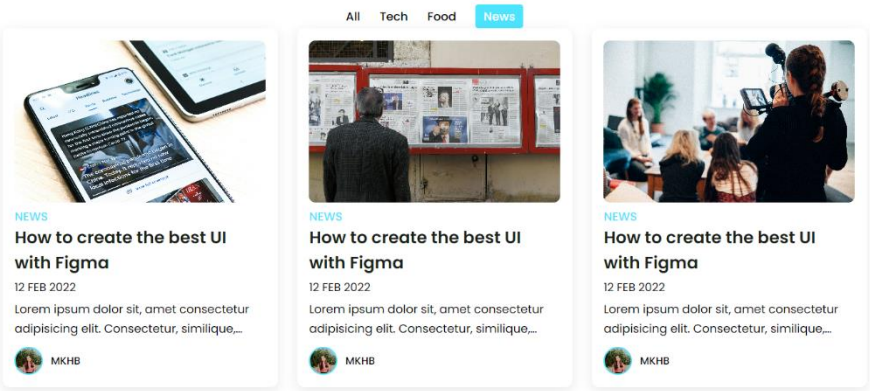
3. Tech Page



4. Food Page



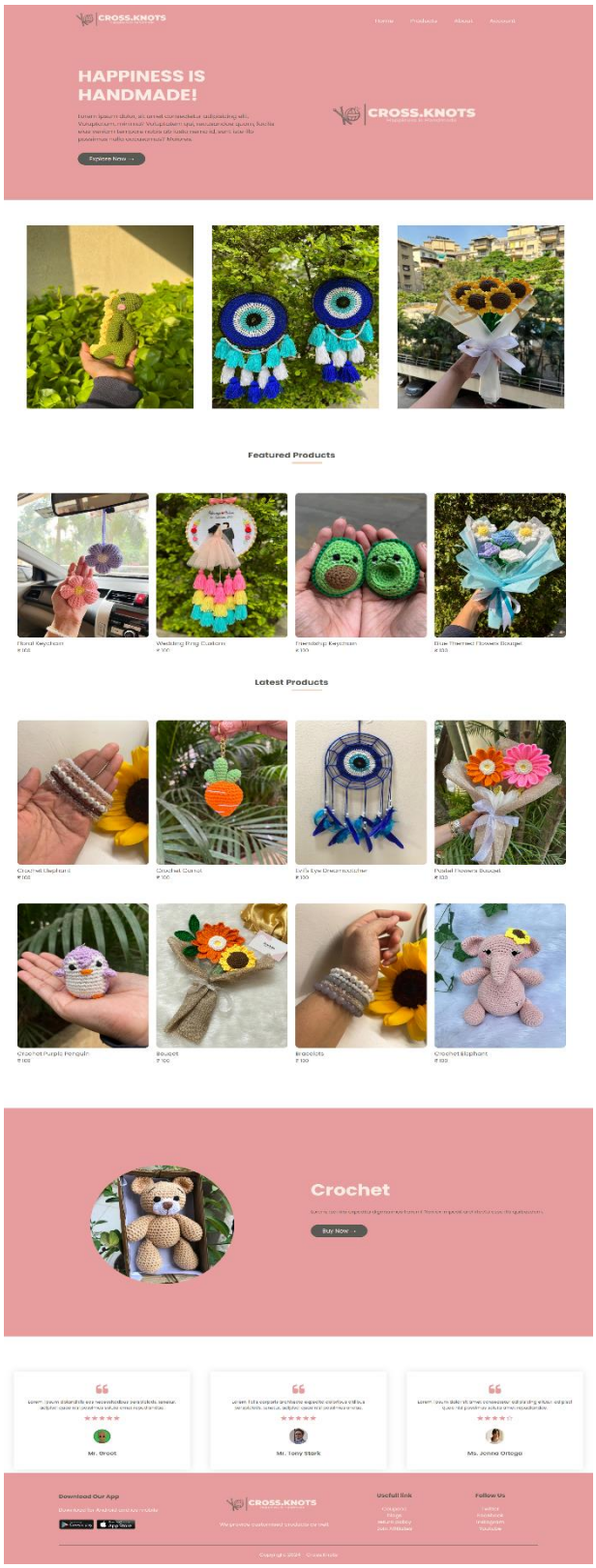
5. News Page

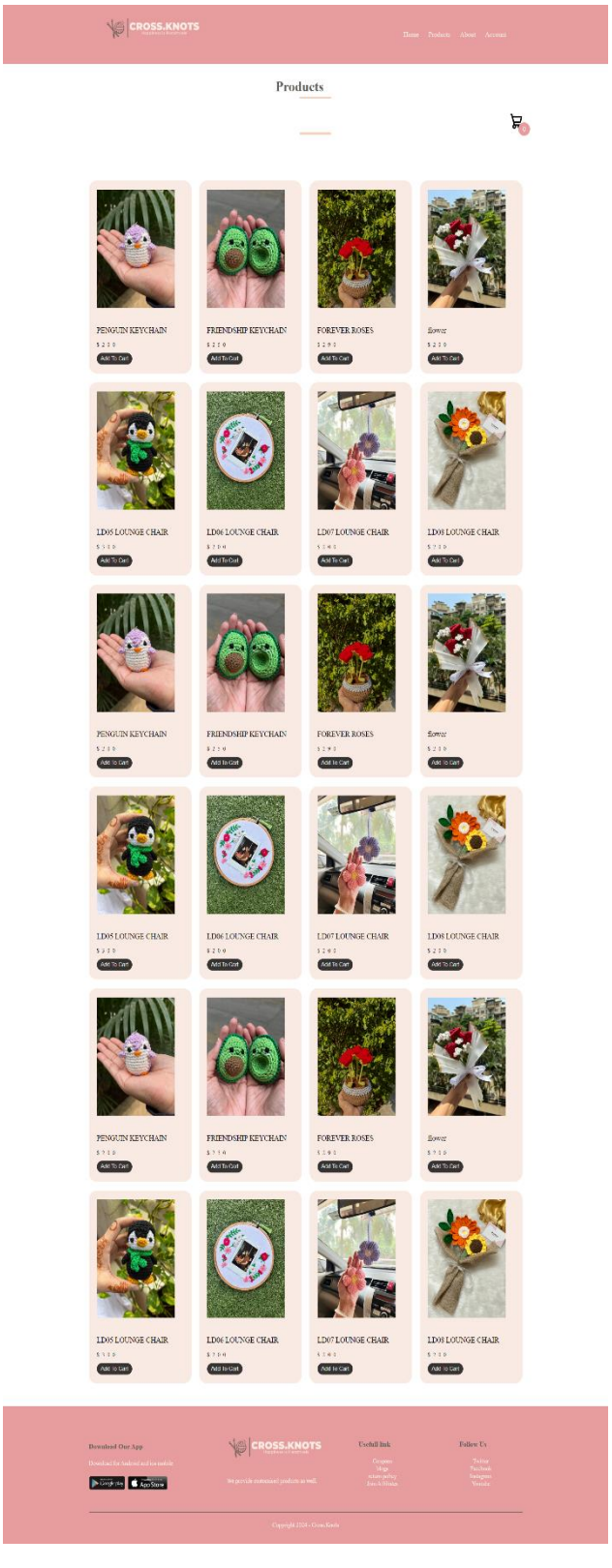

All
Tech
Food
News
NEWS
How to create the best UI with Figma
12 FEB 2022
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Consectetur, similique,...
MKHB
NEWS
How to create the best UI with Figma
12 FEB 2022
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Consectetur, similique,...
MKHB
NEWS
How to create the best UI with Figma
12 FEB 2022
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Consectetur, similique,...
MKHB

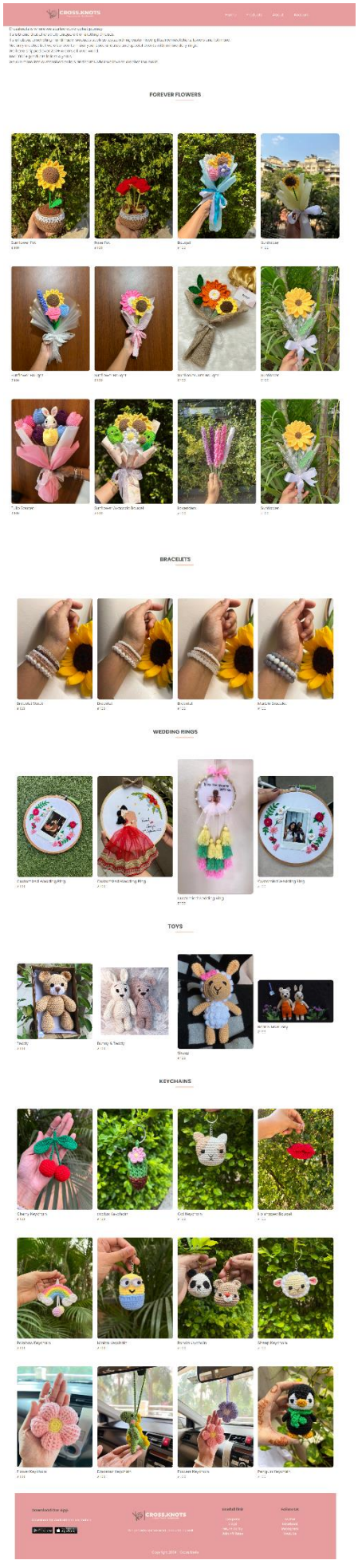
8. Group Project

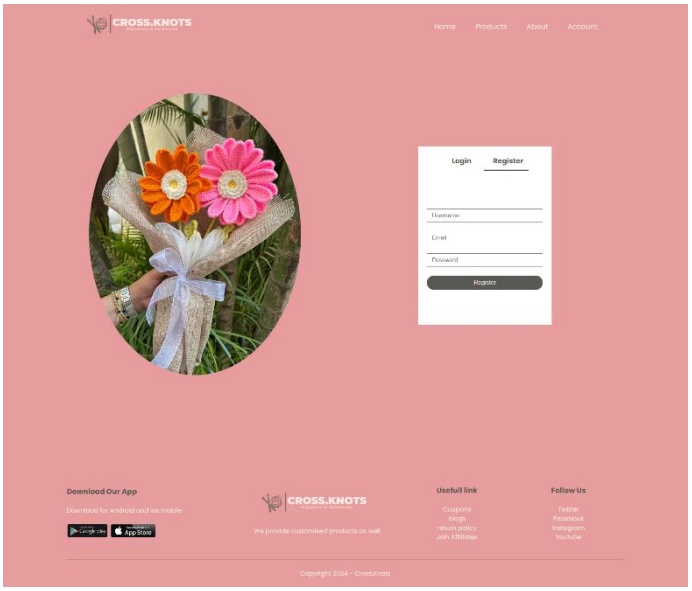
8.1 Introduction:

Creating an E-Commerce website for a small Crochet business.






CROSS.KNOTS
FOREVER FLOWERS
BRACELETS
WEDDING RINGS
TOYS
KEYCHAINS


CROSS.KNOTS
Home
Products
About
Account
Login
Register
Download Our App
CROSS.KNOTS
Usefull link
Follow Us

9. Software Development Lifecycle

9.0 Introduction:

The Software Development Life Cycle (SDLC) is a framework that defines the stages involved in the development of software. It is a structured process that helps in the construction of high-quality software. The SDLC hierarchy is not typically represented as a hierarchy in the sense of a tree structure, but rather as a sequence of phases or stages that a project goes through from conception to maintenance

9.1 Hierarchy:

- **Release theme**

  A "release theme" in the context of software development typically refers to a central concept or focus that defines a particular version or iteration of a software product. The theme helps to guide the development team in making decisions about which features to prioritize and how to align the work with the overall goals of the project or product. Release themes are often used in Agile development methodologies, such as Scrum, where each sprint or release cycle is intended to deliver a potentially shippable increment of product.

- **Epic**

  In the context of software development, particularly within Agile methodologies like Scrum, an "Epic" is a large user story, which is too big to fit into a single sprint or iteration. Epics are used to capture large, coarse-grained requirements or ideas that can later be broken down into smaller, more manageable pieces called "user stories" or "features."
  An Epic typically represents a significant amount of work and can often be complex, involving multiple systems or components. It is a placeholder for a group of related user stories that share a common theme and can be prioritized as a single unit within the product backlog.

- **Stories**

  In the context of software development, particularly within Agile methodologies, "stories" typically refer to "user stories." User stories are a simple way to articulate a requirement or a feature from the perspective of the end user or customer. They are used to ensure that the development team focuses on delivering value to the user with each piece of functionality they create.

- **Stabilization**

  Stabilization in the context of software development refers to the process of making a software product more stable, reliable, and bug-free. This typically occurs after the initial development phase and before the final release of the software. The goal of stabilization is to ensure that the software performs well under expected conditions and that it meets the quality standards set by the development team or the organization.

9.2 Stakeholders:

1. Developers:

A developer, in the context of software development, is a person who is responsible for building and maintaining software applications, systems, and platforms. Developers write code in various programming languages to create software that meets the needs of users, businesses, and other stakeholders.

2. Tester:

A tester, in the context of software development, is a professional who is responsible for evaluating and verifying the quality of software applications, systems, or components. Testers work to ensure that software meets specified requirements, is reliable, and performs as expected before it is released to users.

3. Documentation team:

A documentation team in the context of software development is a group of professionals responsible for creating, maintaining, and updating the written materials that describe the software product. This team ensures that all stakeholders, including end-users, administrators, support staff, and developers, have the necessary information to understand and work with the software effectively.

4. Scrum Master:

A Scrum Master is a key role within the Scrum framework, which is an agile methodology used for managing complex projects, most commonly in software development but also in other fields. The Scrum Master is responsible for promoting and supporting Scrum by helping everyone understand Scrum theory, practices, rules, and values. The Scrum Master is not a traditional project manager but rather a facilitator for the Agile process

5. Product Manager:

A Product Manager (PM) is a professional role that is responsible for the planning, development, launch, and lifecycle management of a product or product line. The product manager acts as the bridge between various teams and departments, ensuring that the product meets market needs and user requirements. The role can vary significantly depending on the company, industry, and the nature of the product (e.g., physical goods, software, services).

9.3 Technical Support Team:

A technical support team is a group of individuals who provide assistance and resolve issues related to technology products or services. This can include software applications, hardware devices, networks, and other technical systems. The team is typically composed of professionals with various levels of expertise, ranging from entry-level support staff to highly skilled technical experts.

Here are some key roles and responsibilities of a technical support team:

1. **Troubleshooting**: Identifying and resolving technical problems that users encounter. This involves systematic investigation and diagnosis to find the cause of the issue and applying the appropriate solution.

2. **Customer Service**: Providing assistance to users in a friendly and professional manner. This includes answering questions, guiding users through the use of products or services, and ensuring customer satisfaction.

3. **Documentation**: Keeping records of customer interactions, process documentation, and knowledge base articles to help users solve common issues on their own and to assist other support team members.

4. **Communication**: Effectively communicating with users to understand their problems and explain solutions in a clear and concise manner. This may involve using different communication channels such as phone, email, chat, or remote desktop support.

5. **Escalation**: When issues cannot be resolved at the initial level of support, the team is responsible for escalating the problem to higher-level support or to the development team if necessary.

6. **Monitoring and Reporting**: Keeping track of support tickets, monitoring system performance, and reporting on common issues or trends that may require further attention or product improvements.

7. **Training**: Staying updated with the latest product knowledge and technology trends through continuous learning and training. They may also be involved in training end-users or other support staff.

8. **Quality Assurance**: Ensuring that the support provided meets the company's quality standards and that users are satisfied with the service.

9.4 Life Cycles:

1. Product lifecycle Management:

PLM, or Product Lifecycle Management, is a comprehensive approach to managing the life of a product from its conception through the end of its life. The PLM process encompasses a wide range of activities, including

1. Idea and Concept Development
2. Design and Engineering
3. Prototyping and Testing**.**
4. Manufacturing**.**
5. Sales and Marketing**.**
6. Service and Support
7. Retirement and Disposal**.**

PLM systems are software solutions that help businesses manage these processes more efficiently. They provide tools for collaboration, data management, and process automation, allowing teams to work together effectively across different stages of the

product lifecycle. PLM can help reduce time-to-market, improve product quality, and increase profitability by optimizing the product development process.

2. Application lifecycle management:

ALM stands for Application Lifecycle Management. It is a methodology used in software development for managing the lifecycle of an application or software system from conception to retirement. The goal of ALM is to unify the various aspects of software development, including requirements management, software design, implementation, testing, deployment, maintenance, and eventual decommissioning.

The application lifecycle:

1. Requirements and Planning
2. Design and Architecture
3. Implementation and Coding
4. Testing
5. Deployment
6. Maintenance and Support
7. Retirement

ALM tools and platforms help organizations automate and manage these processes, providing visibility into the development lifecycle, improving collaboration among teams, and ensuring that the software development process is efficient and effective. ALM is closely related to other disciplines such as Software Configuration Management (SCM), Continuous Integration (CI), and Continuous Delivery (CD), and it often integrates with other systems like issue tracking, version control, and automated testing tools.

Different ways to access sever:

- VMware
- VPN: OPEN SOURCE/PAID SOURCE
- SNAPSHOT
- CLONNIN

9.5 Application:

1. Desktop based

A desktop-based application, also known as a desktop application or a standalone application, is a software program that is installed and runs directly on a personal computer's desktop or laptop's operating system. Unlike web-based applications, which are accessed through a web browser and run on a remote server, desktop applications are executed locally on the user's machine.

Desktop applications can provide a richer user experience with more advanced features and functionalities, as they have direct access to the system's resources such as the file system, CPU, and GPU. They can also work offline, as they do not require an internet connection to function (unless the application specifically needs internet access for its features).

Examples of desktop applications include:

- Microsoft Office Suite (Word, Excel, PowerPoint, etc.)

- Adobe Photoshop
- Mozilla Firefox and Google Chrome (while primarily web browsers, they are also desktop applications)
- Video games that are installed on a computer
- Email clients like Microsoft Outlook or Mozilla Thunderbird
- Accounting software like QuickBooks
- Development environments like Visual Studio or Eclipse

Desktop applications are typically developed using programming languages and frameworks that are specific to the operating system they are designed for

2. Web based

A web-based application, also known as a web application or simply a web app, is a software application that is accessed via a web browser over the Internet or a private network. Unlike desktop applications, which are installed on a local computer, web-based applications are hosted on remote servers and can be accessed from any device with an Internet connection and a compatible web browser.

Web-based applications are built using web technologies such as HTML, CSS, and JavaScript, and they often rely on server-side scripts, databases, and other server-side technologies like PHP, ASP.NET, Java, Ruby on Rails, Node.js, and Python to process data and generate dynamic content.

Examples of web-based applications include:

- Google Docs, Sheets, and Slides
- Web-based email services like Gmail and Outlook
- Social networking sites like Facebook and LinkedIn
- Online shopping platforms like Amazon and eBay
- Project management tools like Asana and Trello
- Customer Relationship Management (CRM) systems like Salesforce

Web-based applications have become increasingly popular due to their convenience, accessibility, and the ability to provide a consistent experience across different devices

9.6 Architecture (3 TIER):

Three-tier architecture, also known as 3-tier architecture, is a client-server architecture pattern that divides an application into three separate layers, each with its own specific responsibilities. This architecture is designed to provide scalability and maintainability by separating the presentation, application processing, and data management functions.

The three tiers are typically as follows:

1. **Presentation Tier (User Interface)**: This is the front-end layer that the user interacts with. It is responsible for displaying information to the user and receiving user input. In web applications, this tier is often composed of HTML, CSS, and JavaScript that run in a web browser. For desktop applications, it could be a GUI (Graphical User Interface) created with a specific programming language or framework.

2. **Application Tier (Business Logic Layer)**: This is the middle tier that contains the application logic and workflows. It processes the user's requests, often by interacting with the data tier, and then sends the appropriate response back to the presentation tier. This tier is where most of the application's business rules are implemented, and it ensures that the data is consistent and valid before it is sent to the data tier.

3. **Data Tier (Data Access Layer)**: This is the back-end layer that interacts with the database or other data sources. It is responsible for storing, retrieving, and persisting data. The data tier ensures that the data is managed securely and efficiently, and it often includes components for data validation, transaction processing, and error handling.

9.7 Servers:

1. **Web server:** A web server is a computer program or a device that serves HTTP (Hypertext Transfer Protocol) requests from clients, which are typically web browsers running on desktop computers or mobile devices. When you type a website address into your browser or click on a web link, your browser sends an HTTP request to the web server that hosts the website you want to visit.

2. **Database server:** A database server is a computer program that provides database services to other computer programs or to computers, as defined by the client-server model. The primary purpose of a database server is to store, manage, and retrieve data upon request by clients. Database servers are often used in conjunction with web servers, application servers, and other types of servers to provide the back-end functionality needed by modern applications.

3. **Directory server:** A directory server, also known as a network directory server or simply a directory service, is a type of server that stores and organizes information about network resources in a directory that is accessible to users and applications. The primary function of a directory server is to provide a centralized location for managing and connecting various network resources, such as user identities, computers, printers, and services.

4. **Application server:** An application server is a software framework that provides both the facilities to generate web pages dynamically and a platform for running applications that operate over the network, especially on the internet. Application servers are part of the middle tier of a multi-tier architecture, which typically includes a database server (backend), an application server (middle tier), and a web server or client (frontend).

5. **File server:** A file server is a computer attached to a network that has the primary purpose of providing a location for shared disk access, i.e., storing and managing files so that other computers on the network can access them. File servers are commonly used in businesses, schools, and other organizations to centralize file storage, provide data access to multiple users, and facilitate collaboration.

9.8 Windchill:

Windchill is a Product Lifecycle Management (PLM) application developed by PTC (Parametric Technology Corporation). It is designed to help companies manage their product-related information and processes throughout the entire lifecycle of a product, from

its conception, through design and manufacturing, to service and disposal.
Windchill is used by a variety of industries, including aerospace, automotive, industrial machinery, medical devices, and consumer goods, to streamline product development processes and improve product quality and time-to-market. It is part of PTC's broader suite of software solutions that also include CAD, augmented reality (AR), and service lifecycle management (SLM) tools.

Steps to open windchill:

1. Open all the servers required
    - Web server
    - Database server
    - Directory server
    - Application server
2. Windchill Shell
3. Command prompt
4. Windchill start

10. Conclusion

We learned how to effectively use HTML, CSS, and JavaScript to build and style web pages, creating responsive and visually appealing designs. Our experience with modern frameworks and libraries like React, Angular, and Vue.js enhanced our ability to develop interactive and dynamic user interfaces. Additionally, by working on projects such as the blogging website "heart.out" and an E-Commerce platform for a small crochet business, we honed our skills in both individual and collaborative settings. We also gained a deep understanding of the Software Development Lifecycle (SDLC), learning how to navigate through phases from conception to stabilization, ensuring the delivery of high-quality software solutions. This comprehensive learning journey has equipped us with the knowledge and expertise to tackle real-world web development challenges and contribute effectively to the digital transformation of businesses.

11. References

- https://scaledagileframework.com/
- https://www.w3schools.com/html/
- https://html.com/
- https://en.wikipedia.org/wiki/CSS
- https://developer.mozilla.org/en-US/docs/Web/CSS