

ASSIGNMENT 1 FRONT SHEET

| | | | |
|--|---------------------------------------|-------------------------------------|---------------|
| Qualification | BTEC HND in Computing | | |
| Unit number and title | Unit 13: Website Design & Development | | |
| Submission date | | Date Received 1st submission | |
| Re-submission Date | | Date Received 2nd submission | |
| Student Name | Lê Đức Anh | Student ID | BH01739 |
| Class | SE06304 | Assessor name | Tạ Quang Hiếu |
| Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
| | | Student's signature | Duc Anh Lee |

Grading grid

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| P1 | P2 | P3 | P4 | M1 | M2 | M3 | D1 |
|----|----|----|----|----|----|----|----|



| | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | | | | |
|--|--|--|--|--|--|--|--|

☐ Summative Feedback:

☐ Resubmission Feedback:

Grade:

Assessor Signature:

Date:

Signature & Date:

Contents

| | |
|--|----|
| I. Introduction..... | 4 |
| II. Content..... | 5 |
| 1. Identify the purpose and types of DNS, including explanations on how domain names are organized and managed | 5 |
| 1.1 Website | 5 |
| 1.2 How the website works..... | 5 |
| 1.3 What is domain name? | 6 |
| 1.4 How the domain names are organized and managed? | 7 |
| 1.5 What is Domain Name System? | 8 |
| 2. Explain the purpose and relationships between communication protocols, server hardware, operating systems and web server software with regards to designing, publishing and accessing a website | 10 |
| 2.1. Web communication protocols..... | 10 |
| 2.2. Server hardware | 11 |
| 2.3. Operating systems (OS)..... | 12 |
| 2.4. Web server software | 14 |
| 2.4. Relationship between communication protocols, server hardware, operating systems and web server software with regards to designing, publishing and accessing a website | 15 |

| | |
|--|----|
| 3. Discuss the capabilities and relationships between front-end and back-end website technologies and explain how these relate to presentation and application layers..... | 17 |
| 3.1. Front-end website technologies | 17 |
| 3.2. Back-end website technologies..... | 19 |
| 3.3. How front-end and back-end website technologies relate to presentation and application layers..... | 21 |
| 4. Discuss the differences between online website creation tools and custom-built sites with regards to design flexibility, performance, functionality, User Experience (UX) and User Interface (UI). | 23 |
| 4.1. Online website creation tools | 23 |
| 4.2. Differences between online website creation tools and custom-built sites | 29 |
| III. Conclusion | 30 |
| IV. References..... | 31 |

I. Introduction

As the leader of a website development team at a leading digital technology solutions company, I am preparing for the development of a Men's fashion sales website, which involves selecting the right tools and technologies for a custom build; to support this, I will create two targeted presentations: one for junior staff, providing a foundation in web technologies by covering website hosting, management, and server basics with a focus on how these elements impact design, functionality, performance, and management; and another for senior staff, offering a deeper dive into front-end and back-end technologies, development tools, techniques, and performance optimization, serving as a practical guide to choosing the best solutions for future projects.

II. Content

1. Identify the purpose and types of DNS, including explanations on how domain names are organized and managed

1.1 Website

A website is a collection of web pages and related content that is identified by a common domain name and published on at least one web server. It is accessible over the internet through a web browser.



Figure 1: website

1.2 How the website works

Websites work by hosting their files on web servers. When a user enters the website's domain name in a browser, the browser sends a request to the web server hosting the website. The server then sends the requested files (HTML, CSS, images, etc.) back to the user's browser, which renders the content.



Figure 2:example

1.3 What is domain name?

A domain name is a human-readable web address that is used to identify a particular website on the internet. It serves as a user-friendly way to access websites without needing to remember the complex numerical IP addresses that actually point to those sites. Domain names are made up of different parts separated by dots, with the last part indicating the type of organization or country associated with the domain.



1.4 How the domain names are organized and managed?

Domain names are organized and managed through a hierarchical system overseen by various organizations. Here's how domain names are structured and managed:

+ Top-Level Domains (TLDs): At the top of the hierarchy are TLDs, which are divided into two main categories:

- Generic Top-Level Domains (gTLDs): Examples include .com, .org, .net, .edu, etc.
- Country-Code Top-Level Domains (ccTLDs): These correspond to specific countries or territories, like .us, .uk, .de, etc.

+ Domain Name Registrars: Domain registrars are companies accredited by the Internet Corporation for Assigned Names and Numbers (ICANN) or national authorities to manage the reservation and registration of domain names for individuals and organizations.

+ Domain Registration Process:

- To register a domain name, an individual or entity needs to choose a registrar and check the availability of the desired domain name.
- Registrars facilitate the registration process, collect the required information, and maintain the registration details in a central database.
- Registrants typically pay a registration fee and provide contact information for administrative, technical, and billing purposes.

+Domain Name System (DNS): As mentioned earlier, the DNS translates domain names into IP addresses, enabling users to access websites using human-readable names. DNS management involves configuring name servers, adding DNS records like A (Address), CNAME (Canonical Name), MX (Mail Exchange), etc., and ensuring the proper resolution of domain names.

+ ICANN: The Internet Corporation for Assigned Names and Numbers (ICANN) is a non-profit organization responsible for coordinating the global internet's systems of unique identifiers, including domain names. ICANN oversees the assignment of domain names, management of TLDs, and policy development related to domain name registration.

+ Registry Operators: Registry operators manage the databases of domain names within specific TLDs. They maintain the authoritative databases and ensure the proper functioning of the TLD.

+ WHOIS Database: The WHOIS database contains information about domain name registrants, such as contact details and registration dates, which can be accessed for verification purposes

1.5 What is Domain Name System?

The domain name system is a configuration facility of the internet that translates user-readable domain names into computer IP addresses for accessing resource resources on the network. The primary purpose of the domain name system is to help facilitate easy and efficient access to websites and services on the internet.



Types of system name domains include:

- Top-Level Domain (TLD): Is the last part of a domain name, created like ".com", ".org", ".net", ".gov". TLDs are managed by organizations such as ICANN (Internet Corporation for Assigned Names and Numbers).
- Country Code Top-Level Domain (ccTLD): These are TLDs designed to represent a specific country or territory, such as ".uk" for the United Kingdom, ".de" for Germany.
- Subdomain: Is part of the main domain name, often used to divide possible websites or services within a large, considered limited domain name such as "blog.example.com" or "shop .example.com".

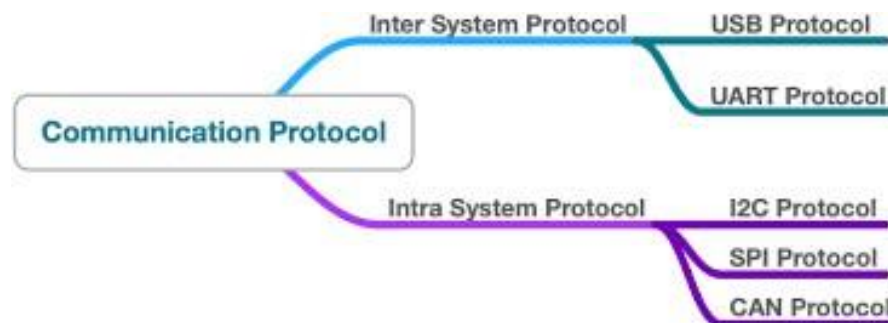
The domain name system operates as follows:

- User Input: When you type a domain name (like `www.example.com`) into your web browser and hit Enter, your device needs to find the corresponding IP address to establish a connection.
- Resolver: Your device sends a DNS query to a DNS resolver (often provided by your ISP or configured manually), asking for the IP address associated with the domain name.
- Root Name Servers: If the resolver doesn't have the IP address in its cache, it will query the root name servers. These servers are the starting point of the DNS resolution process.
- TLD Name Servers: Once the root name servers direct the resolver to the appropriate TLD (Top-Level Domain) name servers (like `.com`, `.net`, or `.org`), the resolver queries these TLD name servers to find the authoritative name servers for the specific domain.
- Authoritative Name Servers: The TLD name servers provide the IP address of the authoritative name servers responsible for the queried domain. The resolver then requests the IP address from these authoritative name servers.
- IP Address Response: The authoritative name servers respond with the IP address associated with the domain name.
- Cache: The resolver caches the IP address it received for future reference, speeding up the process for subsequent queries for the same domain name.
- Connection: Finally, armed with the IP address, your device can establish a connection to the web server hosting the website associated with the domain name you entered.

2. Explain the purpose and relationships between communication protocols, server hardware, operating systems and web server software with regards to designing, publishing and accessing a website

2.1. Web communication protocols

Web communications protocols are rules and standards that define how information is exchanged between devices, servers, and clients on the World Wide Web. This protocol supports data transmission and facilitates subsequent transactions on the internet. Some popular web communication protocols include HTTP, HTTPS, FTP, SMTP, IMAP, and WebSocket.



+ Here are some commonly used web communication protocols:

- Hypertext Transfer Protocol (HTTP): HTTP is the foundation of data communication on the web. It defines how messages are formatted and transmitted between web servers and browsers. HTTP governs the transfer of resources such as HTML pages, images, scripts, and other content.
- Hypertext Transfer Protocol Secure (HTTPS): HTTPS is an extension of HTTP that adds a layer of encryption using SSL/TLS to secure the data exchanged between clients and servers. It ensures data integrity and privacy, making it essential for secure communication, especially for sensitive transactions like online payments.
- File Transfer Protocol (FTP): FTP is a protocol used for transferring files between a client and a server on a network. It enables users to upload and download files to and from a server, making it a common choice for managing website files and data transfers.

- Simple Mail Transfer Protocol (SMTP): SMTP is a protocol used for sending and relaying email messages between servers. It defines how email messages are formatted, transmitted, and delivered to recipient mail servers.
- IMAP (Internet Message Access Protocol) and POP3 (Post Office Protocol): IMAP and POP3 are protocols used by email clients to retrieve email messages from a mail server. IMAP allows users to access and manage email messages directly on the server, while POP3 downloads the messages to the user's device.
- Transmission Control Protocol (TCP) and Internet Protocol (IP): TCP/IP is a suite of protocols that enables communication between devices on the internet. TCP ensures reliable data transmission by establishing connections, managing data packets, and guaranteeing packet delivery. IP is responsible for addressing and routing data packets across networks.
- WebSocket Protocol: WebSocket is a communication protocol that provides full-duplex communication channels over a single TCP connection. It enables real-time, bidirectional communication between web browsers and servers, making it suitable for applications requiring low-latency data exchange

2.2. Server hardware

Figure 6: Server hardware

Server hardware is the physical component of the server, responsible for processing, storage, and resource management. Here are some important components of server hardware:

- Processor (CPU): Is the heart of the server, processing tasks and commands from the operating system and applications. The processor determines the processing capacity and performance of the server.
- RAM memory (Random Access Memory): Is a place to temporarily store data so the server can access and process it quickly. The size and speed of the RAM image affects the overall performance of the server.
- Hard drive (Solid state drive): Is the server's permanent data storage, including the operating system, applications and user data. SSDs are usually faster than HDDs in the process of reading/writing data.
- Network card: Allows the server to connect to the network, manage remote access, and transfer data between the server and other devices on the network.

- Power supply and power supply unit (PSU): Provides power to server components. The PSU converts power from the Wall outlet into power that the server can use.

Motherboard: Connects all other components together and provides interfaces so they can interact with each other.

Heater system (Cooling system): Includes fans and heaters to maintain a stable temperature of the server, preventing them from overheating and affecting performance.

Remote management devices: Modern servers typically integrate remote management technologies, allowing managers to control and monitor the server without needing to be physically present where the server is located.

- RAID controller card: If the server uses a RAID configuration to increase the security and performance of the hard drive, the RAID controller card will manage this process.

Server hardware plays an important role in providing services, applications and resources to users, while ensuring system stability and performance.

2.3. Operating systems (OS)

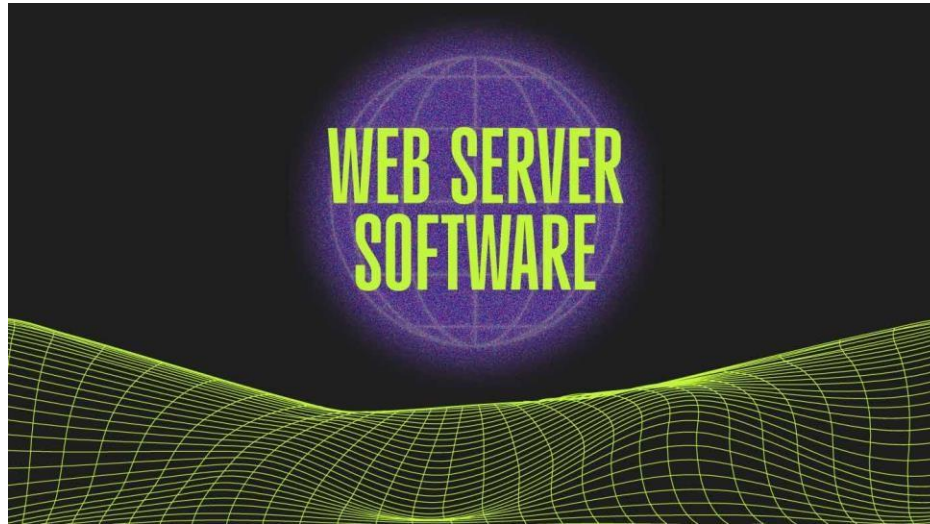
Figure 7: Operating systems

Operating systems (OS) are software that manages computer hardware resources and provides common services for computer programs. Here are some key points about operating systems:

- Resource Management: The OS manages hardware resources such as memory, CPU, storage, and input/output devices to ensure efficient use by applications.
- User Interface: It provides a user interface (UI) for interacting with the computer system. This can be a graphical user interface (GUI) or a command-line interface (CLI).
- File System Management: The OS manages how data is stored, organized, and retrieved on storage devices. It provides file management utilities for creating, deleting, copying, and moving files.

- Process Management: The OS manages processes or programs running on the computer, including allocating CPU time, prioritizing tasks, and handling interrupts.
- Memory Management: It allocates memory to processes, handles virtual memory, and swaps data between memory and storage devices to optimize performance.
- Security: Operating systems provide security features such as user authentication, access control, encryption, and firewall capabilities to protect the system from unauthorized access and malware.
- Networking: Many operating systems include networking capabilities to enable communication between computers and devices, supporting protocols like TCP/IP for internet connectivity.
- Device Drivers: The OS includes device drivers to communicate with hardware components like printers, graphics cards, and network adapters, enabling applications to interact with these devices.
- Updates and Maintenance: Operating system vendors regularly release updates and patches to address security vulnerabilities, improve performance, and add new features to the OS.
- Types of Operating Systems: There are different types of operating systems, including Windows, macOS, Linux, Unix, and mobile operating systems like Android and iOS, each tailored for specific devices and usage scenarios.

2.4. Web server software



Web server software is a crucial component for serving web content over the internet. It handles requests from clients (such as web browsers) and delivers web pages, files, and other content to users. Here are some key points about web server software:

- Handling HTTP Requests: Web server software listens for incoming Hypertext Transfer Protocol (HTTP) requests from clients, processes these requests, and sends back responses typically in the form of HTML pages, images, scripts, or other resources.
- Static vs. Dynamic Content: Web servers can serve both static content (files that do not change, like HTML and images) and dynamic content (generated on-the-fly by applications like content management systems or e-commerce platforms).
- Common Web Servers: Popular web server software includes Apache HTTP Server, Nginx, Microsoft Internet Information Services (IIS), and LiteSpeed. These servers have different performance characteristics, features, and configurations.
- Configuration: Web server software allows administrators to configure settings such as security measures, caching mechanisms, virtual hosts (multiple websites on the same server), and SSL/TLS certificates for secure connections.
- Load Balancing: In high-traffic environments, web servers can be set up in clusters with load balancers to distribute incoming requests across multiple servers for better performance, scalability, and fault tolerance.

- **Logging:** Web servers log information about incoming requests, server errors, and other events, providing administrators with data for troubleshooting, performance monitoring, and security analysis
- **Security:** Web servers implement security features like access control, encryption (using HTTPS), firewalls, and intrusion detection/prevention systems to protect against threats such as hacking, DDoS attacks, and data breaches.
- **Web Application Support:** Web server software can support various technologies for running web applications, including PHP, Python, Ruby, Node.js, and Java, by integrating with application servers or modules.
- **Performance Tuning:** Administrators can optimize web server performance by adjusting settings related to caching, compression, connection handling, and resource utilization to ensure fast response times and efficient use of resources.
- **Continuous Development:** Web server software is continuously updated to address security vulnerabilities, improve performance, and add new features in response to evolving web technologies and standards.

2.4. Relationship between communication protocols, server hardware, operating systems and web server software with regards to designing, publishing and accessing a website

The relationship between communication protocols, server hardware, operating systems, and web server software plays a crucial role in the design, publishing, and access of a website. Let's explore how these components interact in the process:

+Communication Protocols:

- **Design:** Communication protocols, such as HTTP or HTTPS, define the rules for data exchange between the client (web browser) and the server. The choice of protocols influences the design of the website, determining how data is transmitted, secured, and the types of interactions supported.
- **Publishing:** Protocols are essential for transmitting web content. During the publishing phase, the protocols ensure that data is transmitted correctly, and any security measures, such as encryption (as in HTTPS), are implemented to protect data integrity during transfer.
- **Access:** When accessing a website, communication protocols enable the interaction between the client and the server. They govern how requests are made and responses are received, influencing the speed and security of the browsing experience.

+ Server Hardware:

- Design: The hardware specifications of the server influence the design decisions for the website. The server's processing power, memory, and storage capacity impact the website's scalability, performance, and ability to handle concurrent users.
- Publishing: The server hardware is responsible for hosting the website's files and serving them to users. Robust server hardware ensures that the publishing process is efficient, with quick response times and minimal downtime.
- Access: Server hardware affects the website's accessibility. A well-configured and maintained server ensures reliable and fast access for users, even during periods of high traffic.

+ Operating Systems:

- Design: The choice of operating system affects the development environment and tools available for designing the website. Compatibility with certain software and development frameworks is influenced by the operating system.
- Publishing: The operating system manages server resources, file systems, and network connections during the publishing process. It ensures that the web server software can operate efficiently and serves the website to users.
- Access: The operating system plays a crucial role in managing concurrent connections, security, and resource allocation during user access. It ensures the stability and reliability of the website.

+ Web Server Software:

- Design: The selection of web server software (e.g., Apache, Nginx) is part of the design phase. Different web servers may have specific features and configurations that influence how the website is designed and developed.
- Publishing: Web server software is responsible for serving web pages and handling requests. It works in conjunction with the operating system and communication protocols to publish the website content to users.
- Access: The web server software impacts the user experience by handling requests efficiently, supporting features like caching, and ensuring security measures such as SSL/TLS are implemented for secure access.

In summary, a well-coordinated relationship between communication protocols, server hardware, operating systems, and web server software is crucial for the successful design, publishing, and access of a website. Each component contributes to the overall performance, security, and user experience of the website throughout its lifecycle.

3. Discuss the capabilities and relationships between front-end and back-end website technologies and explain how these relate to presentation and application layers

3.1. Front-end website technologies

1. HTML (Hypertext Markup Language)

Definition:

HTML is the standard markup language for creating web pages. It structures the content by using elements such as headings, paragraphs, images, and links.

Advantages:

- Easy to learn and use: Beginner-friendly.
- Widely supported: Works on all browsers without extra configurations.
- Forms the backbone of web content: Every webpage needs HTML.

Disadvantages:

- Limited styling capabilities: Needs CSS for design and visual formatting.
- Static nature: Not suitable for dynamic interactions without JavaScript.

2. CSS (Cascading Style Sheets)

Definition:

CSS is used to control the appearance and layout of HTML elements. It separates content (HTML) from design (CSS), making pages more maintainable.

Advantages:

- Improves user experience: Provides better layouts and aesthetics.
- Reusability: Styles can be reused across multiple pages.
- Responsive design: Enables websites to be mobile-friendly with media queries.

Disadvantages:

- Cross-browser inconsistencies: Some CSS features behave differently in different browsers.
- Complexity in large projects: Managing stylesheets can become challenging in complex apps.
- No dynamic behavior: Requires JavaScript for interactivity.

3. JavaScript (JS)

Definition:

JavaScript is a programming language that adds interactivity and dynamic features to web pages (e.g., animations, form validations, pop-ups).

Advantages:

- Highly interactive: Enables real-time updates without reloading pages.
- Versatile: Works on both front-end and back-end (via Node.js).
- Rich ecosystem: Numerous libraries and frameworks (e.g., React, Vue).

Disadvantages:

- Browser-dependent: Some features may work differently across browsers.
- Security vulnerabilities: JavaScript can be exploited if not properly managed.
- Performance issues: Poorly optimized scripts can slow down applications.

4. ReactJS

Definition:

ReactJS is a JavaScript library for building user interfaces, especially single-page applications. It uses a component-based architecture and manages the user interface's state effectively.

Advantages:

- Reusable components: Enhances development efficiency.
- Virtual DOM: Increases performance by updating only the necessary parts of the UI.
- Strong ecosystem and community: Access to libraries and tools.

Disadvantages:

- Steep learning curve: Requires understanding of JSX and state management.

- Incomplete framework: React focuses only on the UI layer, needing additional tools for routing and state management.
- Frequent updates: Requires developers to keep up with changes.

5. AngularJS

Definition:

AngularJS is a front-end framework developed by Google, designed for building dynamic web applications. It allows developers to use HTML as a template language and extend it with features like data binding and dependency injection.

Advantages:

- Two-way data binding: Syncs data between the model and the view automatically.
- Comprehensive framework: Provides solutions for routing, form handling, and state management.
- Modular development: Code is divided into modules, making it easier to maintain.

Disadvantages:

- Complex and heavy: Can be overkill for small projects.
- Performance issues: Two-way data binding can slow down complex applications.
- Steep learning curve: Requires knowledge of TypeScript and Angular concepts like dependency injection.

3.2. Back-end website technologies

1. PHP

Definition:

PHP (Hypertext Preprocessor) is a widely-used, open-source server-side scripting language suited for web development.

Uses:

- Creating dynamic web pages and content management systems (e.g., WordPress, Joomla).
- Building RESTful APIs for websites and applications.

- Server-side handling of forms, authentication, and session management.

Advantages:

- Easy to learn: Great for beginners with a low learning curve.
- Wide community support: Extensive documentation and tutorials.
- Cross-platform compatibility: Runs on Windows, Linux, and macOS.
- Seamless integration with databases: Works well with MySQL and other SQL databases.

Disadvantages:

- Performance limitations: Slower compared to more modern technologies like Node.js.
- Security concerns: Requires careful management to prevent vulnerabilities.
- Not ideal for large-scale applications: Can become hard to maintain for complex projects.

2. Java

Definition:

Java is a high-level, object-oriented programming language designed to be platform-independent, often used for large-scale enterprise applications.

Uses:

- Building enterprise-level web applications and Android apps.
- Server-side development with frameworks like Spring and Hibernate.
- Handling back-end logic for financial institutions and e-commerce platforms.

Advantages:

- Robust and secure: Strong typing and built-in security features.
- Highly scalable: Suitable for large, distributed systems.
- Platform-independent: Code runs on any device with a JVM (Java Virtual Machine).
- Strong community and frameworks: Well-supported frameworks (e.g., Spring).

Disadvantages:

- High memory consumption: May require more system resources.
- Steeper learning curve: Requires knowledge of object-oriented principles.

- Verbose code: Development can be more time-consuming compared to other languages.

3. Node.js

Definition:

Node.js is a JavaScript runtime built on Chrome's V8 engine that allows developers to execute JavaScript on the server side.

Uses:

- Building scalable network applications, such as chat applications.
- Developing RESTful APIs and microservices.
- Powering real-time applications (e.g., multiplayer games, collaborative tools).

Advantages:

- Non-blocking I/O: Can handle concurrent requests efficiently, making it ideal for I/O-heavy applications.
- Unified language: JavaScript on both the client and server side improves code reusability.
- Rich ecosystem: Access to numerous packages via npm (Node Package Manager).
- Great for real-time apps: Supports WebSockets and real-time data updates.

Disadvantages:

- Single-threaded nature: While it can handle many connections, CPU-heavy tasks may block the event loop.
- Call-back hell: Asynchronous code can become complex to manage, though tools like async/await help.
- Relatively new: Fewer mature tools and libraries compared to PHP or Java.

3.3. How front-end and back-end website technologies relate to presentation and application layers

+ Front-end and Back-end Website Technologies in OSI Model:

- Front-end Technologies (Presentation Layer):

Role: Front-end technologies primarily operate in the presentation layer. They are responsible for rendering and displaying the user interface that end-users interact with.

Functions: HTML, CSS, and JavaScript are common front-end technologies. HTML structures the content, CSS styles the presentation, and JavaScript handles dynamic interactions. These technologies ensure that the user sees and interacts with the website's content in a visually appealing and user-friendly manner.

- Back-end Technologies (Application Layer):

Role: Back-end technologies operate in the application layer. They manage the server-side logic, handle data storage and retrieval, and perform the business logic of an application.

Functions: Server-side scripting languages like PHP, Java, Python, and frameworks like Node.js are common in the back end. They process user requests, interact with databases, and handle the application's core functionality, such as user authentication, data processing, and server-side computations.

+ Relationship between Front-end and Back-end:

- Communication:

Front-end to Back-end: The front-end sends user requests to the back-end for processing through HTTP requests. The back-end processes these requests and returns the necessary data.

Back-end to Front-end: The back-end sends data to the front-end, which then uses HTML, CSS, and JavaScript to present the information to the user.

- Data Exchange:

Front-end: Deals with how data is presented and interacts with the user. Back-

end: Manages how data is processed, stored, and retrieved.

- User Interaction:

Front-end: Focuses on user interface and user experience.

Back-end: Manages the underlying functionality and business logic.

4. Discuss the differences between online website creation tools and custom-built sites with regards to design flexibility, performance, functionality, User Experience (UX) and User Interface (UI).

4.1. Online website creation tools

Online website creation tools are platforms that enable users to design and manage websites with minimal technical expertise or coding knowledge. These tools typically feature intuitive drag-and-drop editors, pre-built templates, and element libraries, allowing users to create websites efficiently. They cater to various needs, including personal blogs, business sites, online stores, and portfolios. Many of these platforms also provide integrated hosting services, simplifying the process of launching and maintaining websites.

1. WordPress

Definition:

WordPress is a free, open-source content management system (CMS) that powers over 40% of websites globally. It offers flexibility through themes, plugins, and a large developer ecosystem.

Advantages:

- Highly customizable with themes and plugins.
- Strong SEO capabilities and tools.
- Large community support and documentation.

Disadvantages:

- Requires technical knowledge for advanced customization.
- Security vulnerabilities if plugins/themes are not managed carefully.

- Some hosting and premium plugins can be expensive.
- Steps to Create a Website with WordPress:
- Register a domain and choose a hosting provider (e.g., Bluehost, SiteGround).
- Install WordPress via the hosting control panel or manually.
- Select a theme (free or paid) from the WordPress theme library.
- Customize the site using the WordPress editor.
- Add essential plugins (e.g., SEO plugins, contact forms).
- Publish pages and posts.

2. Wix.com

Definition:

Wix is a cloud-based website builder with drag-and-drop functionality, designed for users without coding experience.

Advantages:

- Extremely easy to use with drag-and-drop editor.
- Provides hosting, domain management, and templates in one place.
- Good for small businesses and personal websites.

Disadvantages:

- Limited flexibility compared to open-source platforms.
- Free version shows Wix ads.
- Migration to other platforms is difficult.
- Steps to Create a Website with Wix:
- Create an account on Wix.com.
- Choose a template or use Wix ADI (Artificial Design Intelligence) for auto-building a website.
- Customize the layout using the drag-and-drop editor.
- Add essential pages (home, about, contact).
- Connect a domain (free or paid).

- Publish the website.

3. Joomla

Definition:

Joomla is an open-source content management system (CMS) that offers advanced features and more flexibility than WordPress, suitable for complex websites.

Advantages:

- Powerful user management and access control features.
- Offers multilingual support out of the box.
- Highly flexible for developers.

Disadvantages:

- Steeper learning curve than WordPress.
- Fewer plugins and themes compared to WordPress.
- Requires more technical knowledge for setup and maintenance.
- Steps to Create a Website with Joomla:
 - Register a domain and hosting provider that supports Joomla.
 - Install Joomla through the hosting control panel or manually.
 - Choose and install a Joomla template.
 - Add content using the Joomla editor and modules.
 - Install extensions (e.g., for SEO, security).
 - Configure menus and publish the site.

4. CoffeeCup

Definition:

CoffeeCup offers web design software tools, including a responsive website builder that allows users to create and customize web pages with minimal coding knowledge.

Advantages:

- Allows full control over design and code.
- One-time purchase for software (no recurring costs).
- Suitable for developers who prefer custom designs.

Disadvantages:

- Not cloud-based: Software must be installed on a computer.
- Requires some coding knowledge for advanced customization.
- Limited templates compared to Wix or WordPress.
- Steps to Create a Website with CoffeeCup:
 - Download and install CoffeeCup Responsive Site Designer.
 - Choose a template or start from scratch.
 - Use the visual editor or code editor to customize the design.
 - Add content, images, and other media.
 - Export the project and upload it to your web hosting server.
 - Connect a domain and publish the site.

Example Steps to Create a Website Using Different Tools**1. WordPress**

Choose a web hosting provider (e.g., Bluehost, SiteGround).

Install WordPress via the hosting control panel.

Log in to the WordPress dashboard.

Choose a theme and install plugins (like SEO and contact forms).

Customize the site with pages, posts, and menus.

Launch your website.

2. Wix.com

Create a Wix account.

Choose a website template or use the Wix ADI (Artificial Design Intelligence).

Customize the layout with the drag-and-drop editor.

Add necessary pages (e.g., About, Contact).

Optimize with Wix SEO tools.

Publish your site.

3. Joomla

Choose a hosting provider and install Joomla.

Log in to the Joomla admin panel.

Select and install a template.

Add extensions to enhance functionality.

Create articles and menus.

Launch your website.

4. CoffeeCup

Download and install the CoffeeCup software.

Start a new project and design the layout.

Write HTML and CSS for your website.

Use CoffeeCup's responsive tools for mobile optimization.

Export the code and upload it to your web host.

Test the site and go live.

Comparison Table of Website Creation Tools:

| Tool | Ease of Use | Best For | Hosting Included | Customization | Cost | Support/Community |
|------------------|-------------|----------------------------|------------------|---------------------|----------------------|-----------------------|
| WordPress | Moderate | Blogs, CMS-based websites | No | High (with plugins) | Free + Paid Themes | Large community |
| Wix.com | Easy | Small business, portfolios | Yes | Limited | Free + Premium Plans | Good built-in support |

| | | | | | | |
|------------------|------------------|----------------------------|----|------------------------|--------------------|------------------|
| Joomla | Moderate to Hard | Complex websites, apps | No | High (with extensions) | Free + Paid Themes | Active community |
| CoffeeCup | Hard | Developers, offline design | No | Very High (Code-based) | One-time purchase | Limited |

4.2. Differences between online website creation tools and custom-built sites

- UI (User Interface):

Definition: UI is the abbreviation for "User Interface". It is a part of a system or application with which users directly interact to perform work or operations.

Functiona: UI includes all the subtle elements that the user can see and interact with, such as buttons, toolbars, colors, graphics, and other elements of the interface.

- UX (User Experience):

Definition: UX is the abbreviation for "User Experience". It describes the total experience that a user might have when using a product or service.

Function: UX includes feelings, thoughts, emotions and all other aspects of the experienter. It involves making a product or service easy to use, effective, and satisfying.

| Comparison criteria | Online website creation tools | Custom built sites |
|---------------------|-------------------------------|--------------------|
| | | |

| | | |
|-----------------------------|---|--|
| Flexibility | These tools typically offer a range of pre- designed templates that users can customize to a certain extent. While they provide some level of flexibility in terms of layout and colors, the customization options may be limited compared to custom-built sites. | Custom-built sites offer the highest level of design flexibility as they can be tailored to meet specific design requirements. Designers and developers have complete control over every aspect of the website's design, allowing for a unique and highly customized look. |
| Performance | Websites built using online tools may sometimes suffer from performance issues, especially if the platform's servers are slow or overloaded. The performance may also be affected by the amount of customizations made to the template. | Custom-built sites have the potential to achieve better performance as developers can optimize the code, server configuration, and other technical aspects to ensure the site runs smoothly and loads quickly. |
| Functionality | These tools often come with a set of built-in features and plugins that users can easily add to their websites. However, the range of available functionalities may be limited compared to custom-built sites. | Custom-built sites can have any functionality imaginable, as developers can create custom features and integrations tailored to the specific needs of the site. This allows for more advanced and specialized functionalities. |
| User Experience (UX) | The user experience of websites built with online tools can vary depending on the platform used. Some tools offer intuitive interfaces and design features that can enhance UX, while others may be more limiting. | With custom-built sites, designers and developers have full control over the user experience. They can create seamless and user-friendly interfaces that are tailored to the target audience, resulting in a more personalized and engaging UX. |

III. Conclusion

In conclusion, as we embark on the journey of developing a Men's fashion sales website, the careful selection of tools and techniques plays a pivotal role in ensuring the success of our project. The foundation of our web solutions will rely on a comprehensive understanding of basic web technologies, hosting, and website management, which are crucial elements for creating

a seamless and efficient user experience. The impact of these technologies on website design, functionality, management, and performance cannot be overstated, and by imparting this knowledge to our junior staff members, we empower them to contribute effectively to the project.

Simultaneously, our technical presentation to senior staff members delves into the intricacies of front-end and back-end technologies, along with an exploration of various tools, techniques, and software options ranging from user-friendly online website creation tools to the complexities of custom-built solutions. Recognizing the significance of these choices in shaping the project's outcome, we have provided a roadmap for informed decision-making, aligning our development strategy with the project's unique requirements.

In the dynamic landscape of web development, the synergy between technology and creativity is paramount. Our presentation serves as a compass, guiding us towards the selection of suitable tools and techniques that will elevate our Men's fashion sales website to new heights. By fostering collaboration and understanding across our team, we are poised to deliver a cutting-edge solution that not only meets but exceeds the expectations of our clients. Together, we are ready to embark on this exciting venture and create a digital shopping experience that stands out in the competitive online marketplace.

IV. References

Smith, J. (2022). What is Web Hosting? - Definition from Techopedia. [online] Available at:

<https://www.techopedia.com/definition/5350/web-hosting>

HostGator. (n.d.). Choosing the Best Web Hosting Plan for Your Business. [online] Available at:

<https://www.hostgator.com/blog/choosing-web-hosting-plan/>

GoDaddy. (2023). What is a Website Builder? A Guide for Beginners. [online] Available at: <https://www.godaddy.com/website-builder/overview>

DigitalOcean. (2020). Introduction to Cloud Servers. [online] Available at: <https://www.digitalocean.com/community/tutorials/what-is-a-cloud-server>

Linode. (2021). Server Infrastructure Explained. [online] Available at: <https://www.linode.com/server-infrastructure/>

TechRadar. (2023). Best Front-End Frameworks in 2023. [online] Available at: <https://www.techradar.com/front-end-frameworks>

W3Schools. (2023). Introduction to Backend Development. [online] Available at: <https://www.w3schools.com/backend/>

OWASP Foundation. (n.d.). What is Website Security? [online] Available at: <https://owasp.org/website-security-guide/>

MDN Web Docs. (2023). What is CSS? - Cascading Style Sheets Overview. [online] Available at: <https://developer.mozilla.org/css-intro>

StackPath. (2022). Content Delivery Networks (CDN): How They Work. [online] Available at: <https://www.stackpath.com/cdn-overview/>