**ASSIGNMENT FINAL REPORT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC level 5 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** | HOANG ANH QUAN | **Student ID** | BD00812 |
| **Class** | SE07203 | **Assessor name** | NGUYEN TRUNG ANH |
| **Plagiarism**  Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.  **Student Declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I declare that the work submitted for assessment has been carried out without assistance other than that which is acceptable according to the rules of the specification. I certify I have clearly referenced any sources and any artificial intelligence (AI) tools used in the work. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** |  |

**Grading grid**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | P6 | P7 | M1 | M2 | M3 | M4 | M5 | D1 | D2 | D3 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| ** Summative Feedback:  Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Internal Verifier’s Comments:** | | |
| **Signature & Date:** | | |

**TABLE OF CONTENT**

**TABLE OF FIGURE**

**LIST OF TABLE**

**INTRODUCE**

**CHAPTER 1: Explain server technologies and management services associated with hosting and managing websites (LO1)**

**1.1 Identify the purpose and types of DNS, including explanations on how domain names are organized and managed.(P1)**

**1.1.1 Website definition**

A website is "a location connected to the Internet that maintains one or more pages on the World Wide Web"

**1.1.2 How Does a Website Work?**

A website works through the interaction between a web browser, a web server, and the internet. How website works:

1. User Enters Website Address

* The user types a web address (e.g., www.example.com) into a browser like Chrome or Firefox.
* This address is called a URL (Uniform Resource Locator).

2. Request Sent to Web Server

* The browser sends a request over the internet to the server that hosts the website.
* This request is for the specific web page the user wants to view.

3. Server Processes the Request

* The web server receives the request and locates the relevant files (HTML, CSS, images, scripts, etc.).
* If the site is dynamic, the server might also interact with a database to generate the correct content.

4. Response Sent Back to Browser

* The server sends the web page files back to the user’s browser.

5. Browser Displays the Page

* The browser receives the files and renders them into a viewable format — text, images, layout, and interactive elements.
* The user can now read, scroll, click, or interact with the website.

**1.1.3 Definition of Domain Name**

A domain name is "a unique name that identifies a particular website on the internet, made up of a series of characters separated by dots"

**1.1.4 How are the domain names organized and managed?**

Domain names are a fundamental part of the internet's addressing system, enabling users to access websites using easy-to-remember names instead of numerical IP addresses. Their organization and management follow a clear hierarchical structure and are overseen by global authorities to maintain order and prevent duplication.

1. Hierarchical Structure of Domain Names

* A domain name is structured in levels, read from right to left. For example, in www.example.com:
* Top-Level Domain (TLD): .com – indicates the type or origin of the domain (e.g., .org, .gov, or country codes like .uk, .vn).
* Second-Level Domain: example – chosen by the registrant, this is the main identity of the website.
* Subdomain (optional): www – used to organize or direct to specific parts of the site.
* This hierarchy ensures domains are logically organized and easy to manage globally.

2. Domain Name Management

The global coordination of domain names is managed by the Internet Corporation for Assigned Names and Numbers (ICANN). ICANN is responsible for maintaining the Domain Name System (DNS), ensuring that every domain name is unique and correctly mapped to an IP address.

Under ICANN, domain management is divided into:

* Registries, which maintain the database of all domain names under a specific TLD (e.g., Verisign manages .com).
* Registrars, which are companies authorized to sell domain names to the public (e.g., GoDaddy, Namecheap). Registrars handle the actual registration and renewal process for users.

3. Domain Name Registration

When a user registers a domain name through a registrar, their ownership is recorded in a global database (called WHOIS), which includes the domain's expiration date and contact information. Domains must be renewed periodically, or they risk becoming available to others.

4. Role of the DNS (Domain Name System)

The DNS acts like the phonebook of the internet. It translates human-friendly domain names into machine-readable IP addresses. For example, when a user types www.google.com, the DNS helps locate the correct server by converting the name into its corresponding IP address.

**1.1.5 Definition of Domain Name System (DNS)**

“A system that translates domain names into IP addresses, allowing users to access websites using easy-to-remember names instead of numeric addresses”

**1.1.6 Purpose and Types of DNS – How It Works**

What’s the Purpose of DNS?

The Domain Name System (DNS) helps us use the internet more easily. Instead of remembering long numbers (IP addresses), we just type names like www.google.com, and DNS figures out where to take us. It’s like a contact list: you click someone’s name, and your phone dials their number — DNS does the same, but for websites.

* Types of DNS Servers:

There are a few types of DNS servers, and they all work together to help your browser find the right website:

a) DNS Resolver (or Recursive Resolver)

This is the first stop. It’s the one your device asks, “Where is this website?”

b) Root DNS Server

Points the resolver in the right direction — like saying, “Ask someone who handles .com addresses.”

c) TLD Server (Top-Level Domain)

Handles domains like .com, .org, or country-specific ones like .uk, .vn.

d) Authoritative Name Server

This is where the real answer is. It knows the exact IP address for the website you’re trying to reach.

* How DNS Works (Simple Steps)

1. You type a website name in your browser.

2. Your device sends a request to a DNS resolver.

3. The resolver asks the root server, “Where can I find .com domains?”

4. The root server replies, “Ask this TLD server.”

5. The TLD server says, “Ask this authoritative server for example.com.”

6. The authoritative server gives back the IP address.

7. Now your browser knows where to go and loads the website.

**1.2 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.(P2)**

**1.2.1 Web communication protocols**

Web communication protocols are the rules that allow computers and servers to talk to each other over the internet. The most important one is HTTP, which stands for Hypertext Transfer Protocol. This protocol sets the way browsers request web pages and how servers respond with the content. Without HTTP, the web as we know it wouldn’t work.

Because security is important, we also use HTTPS, which is basically HTTP but with encryption. It keeps the data exchanged between your browser and the website private and secure — this is why you see the little lock icon in your browser bar on secure sites.

Besides these, there are other protocols like FTP (File Transfer Protocol), which is useful when you want to upload or download files to and from your web server. Another is WebSocket, which is designed for real-time communication, like chat apps or live updates, enabling the server and client to keep a constant connection open.

Each protocol plays a specific role in how websites send and receive data, ensuring the web works quickly, securely, and reliably.

**1.2.2 Server hardware**

Server hardware is the backbone of any web server, and it plays a critical role in how well a website performs. Main components:

* CPU (Central Processing Unit):
  + This is the server's brain. It processes all the requests that come in from users. A powerful CPU can handle more requests at once, which is essential for busy websites.
* RAM (Random Access Memory):
  + Think of RAM as the server's short-term memory. It temporarily stores data that the CPU needs to access quickly. More RAM means the server can manage more tasks simultaneously, which is crucial during peak traffic times.
* Storage:
  + This is where all your website files, databases, and applications are kept. You can choose between Hard Disk Drives (HDD) and Solid State Drives (SSD). SSDs are faster and more reliable, which can significantly speed up how quickly your website loads.
* Network Interface Card (NIC):
  + This component connects the server to the internet. A good NIC can improve data transfer speeds, which is important for delivering content to users quickly.
* Power Supply and Cooling Systems:
  + These are essential for keeping the server running smoothly. A reliable power supply ensures that the server stays on, while effective cooling systems prevent overheating, which can lead to hardware failure.

**1.2.3 Operating Systems.**

The operating system (OS) is the software that manages the server's hardware and provides a platform for running web server software. Choosing the right OS is crucial for the performance and security of a web server. Here are the most common operating systems used for web servers:

* Linux:
  + This is the most popular choice for web servers, and for good reason. It’s open-source, which means it’s free to use and has a large community for support. Linux is known for its stability and security, making it ideal for hosting websites. Popular distributions include Ubuntu, CentOS, and Debian.
* Windows Server:
  + This is a proprietary operating system from Microsoft. It’s often used in environments that require integration with other Microsoft products, like ASP.NET applications. While it can be more expensive than Linux, it offers a user-friendly interface and strong support for enterprise applications.
* Unix:
  + Known for its robustness and security, Unix is often used in high-performance computing environments. It’s less common for general web hosting but is still a solid choice for specific applications that require its capabilities.

**1.2.4 Web Server Software**

Web server software is crucial because it handles requests from clients (like web browsers) and serves the appropriate content. The choice of web server software can significantly impact the performance, scalability, and security of a website. Here are some of the most popular options:

* Apache HTTP Server:
  + This is one of the oldest and most widely used web servers. It’s open-source and highly customizable, with a vast array of modules that can extend its functionality. Apache is known for its flexibility and is suitable for a variety of applications.
* Nginx:
  + Nginx is a high-performance web server that excels at handling a large number of concurrent connections. It’s often used as a reverse proxy and load balancer, making it a great choice for high-traffic websites. Its efficient use of resources allows it to serve static content quickly.
* Microsoft Internet Information Services (IIS):
  + This is a web server for Windows Server environments. IIS is known for its integration with other Microsoft technologies, making it a good choice for businesses that rely on the Microsoft ecosystem. It offers a user-friendly interface and strong support for ASP.NET applications.
* LiteSpeed:
  + LiteSpeed is a commercial web server known for its speed and efficiency, particularly in serving dynamic content. It can be a drop-in replacement for Apache, providing better performance without requiring significant changes to existing configurations.

**1.3 Analyze the impact of common web development technologies and frameworks with regards to website design, functionality and management.(M1)**

**1.3.1 State the role of frameworks in website design and development**

A framework in web development is a pre-built collection of tools, libraries, and conventions that help developers build websites and applications more efficiently and consistently. Frameworks provide a structured foundation, so developers don’t have to write everything from scratch. They include reusable components, design patterns, and rules to follow, which speeds up development, improves code quality, and ensures better maintainability.

Roles of frameworks:

* Simplify and speed up the development process by offering ready-made solutions.
* Ensure consistency in design and coding style across a project or team.
* Help manage complexity in large applications by providing organized structures.
* Improve security by including tested components and best practices.
* Facilitate responsive design and cross-browser compatibility.
* Provide tools for testing, debugging, and optimizing websites.

**1.3.2 List a few frameworks for front end and back end**

* Front-End Frameworks:
* React.js – A JavaScript library for building user interfaces, especially single-page applications.
* Angular – A full-featured TypeScript-based framework developed by Google.
* Vue.js – A progressive JavaScript framework for building user interfaces.
* Bootstrap – A CSS framework for designing responsive and mobile-first websites.
* Back-End Frameworks:
* Node.js (with Express.js) – A JavaScript runtime with Express as a minimal web framework.
* Django – A Python-based high-level framework focused on rapid development.
* Ruby on Rails – A Ruby framework known for convention over configuration.
* Laravel – A PHP framework with elegant syntax and powerful features.

**1.3.3 Characteristics, Advantages, and Disadvantages of Frameworks**

1.Front-End Frameworks

* React.js
  + Characteristics:
    - Component-based architecture
    - Uses a virtual DOM for fast rendering
    - Maintained by Facebook
    - Focuses on building UI components
  + Advantages:
    - High performance thanks to virtual DOM
    - Reusable components promote modularity
    - Large community and lots of resources
    - Easy integration with other libraries
  + Disadvantages:
    - Only handles the UI layer (needs other tools for full apps)
    - Learning curve with JSX syntax
    - Frequent updates can cause compatibility issues
* Angular
  + Characteristics:
    - Full MVC (Model-View-Controller) framework
    - Uses TypeScript
    - Developed and maintained by Google
    - Provides two-way data binding
  + Advantages:
    - Comprehensive: handles both UI and business logic
    - Two-way data binding reduces boilerplate code
    - Built-in tools for testing and routing
    - Strong community support
  + Disadvantages:
    - Steeper learning curve due to complexity
    - Larger framework size can affect load times
    - Verbose and complex syntax for beginners

2. Back-End Frameworks

* Express.js (Node.js framework)
  + Characteristics:
    - Minimalist and flexible web framework for Node.js
    - Middleware-based architecture
    - Lightweight and fast
  + Advantages:
    - Easy to learn for JavaScript developers
    - Highly customizable due to minimal design
    - Large ecosystem of plugins and middleware
    - Good performance for scalable applications
  + Disadvantages:
    - Requires manual configuration for many features
    - Not as opinionated, so development practices vary
    - Less built-in functionality compared to full-stack frameworks
* Django
  + Characteristics:
    - High-level Python framework
    - Follows the “batteries-included” philosophy
    - Built-in ORM, admin panel, and authentication
  + Advantages:
    - Rapid development with many features out-of-the-box
    - Strong security features
    - Scalable and robust for complex applications
    - Excellent documentation and community support
  + Disadvantages:
    - Can be heavyweight for simple projects
    - Less flexible due to built-in conventions
    - Steeper learning curve for beginners in Python web dev

**1.3.4 Choose the appropriate framework for the scenario and give reasons.**

Scenario:

I work as a comprehensive website development team leader for a leading digital technology solutions company. My team is about to get a big contract to develop an online fashion sales website. One of the preparatory tasks is choosing the appropriate tools and techniques to realize a custom-built website. As part of my role, I am required to create an engaging presentation to help train junior staff in basic web technologies including website hosting and management as well as server technology. Your presentation should not only explain the basics in that field but also show the impact of these technologies on the design, functionality, management or performance of the website. You also need to give more technical presentations to senior staff to discuss front-end and back-end technologies as well as other tools, techniques, and software used to develop websites from simple ( online website creation tools) to complex (custom-built) . Your presentation will serve as a guide to choosing the right tools and techniques for your next project.

For the given scenario — developing a custom-built online fashion sales website — the selected frameworks are:

* Front-End Framework: React.js
* Back-End Framework: Django
* Why React.js for the Front End?
* React.js is a powerful JavaScript library maintained by Meta (Facebook). It is ideal for dynamic, modern web interfaces like those needed in a fashion e-commerce platform. The use of component-based architecture allows developers to build reusable UI elements such as product cards, filters, shopping carts, and sliders — all of which are common in fashion websites.
* Key Benefits:
* High performance using the virtual DOM, which enhances the user experience.
* Flexibility to integrate with other libraries or tools like Redux (for state management).
* Mobile-first design capabilities when combined with frameworks like React Native or Bootstrap.
* These benefits are crucial for fashion e-commerce, where visual appeal and quick interactions (like changing size, color, or quantity) matter greatly to customers.
* Why Django for the Back End?
* Django is a Python-based framework that follows the “batteries-included” philosophy. It provides built-in features such as an admin panel, user authentication, and ORM (Object-Relational Mapping) which makes it easy to manage databases. For a fashion sales platform that requires secure logins, product inventory management, and order processing, Django is a reliable and efficient choice.
* Key Benefits:
* Rapid development – saves time with built-in components.
* Strong security features – protection against SQL injection, CSRF, and XSS attacks.
* Scalability – can handle large product inventories and high traffic with proper optimization.

**1.4 Review the influence of search engines on website performance.(M2)**

**1.4.1 Definition of SEO**

SEO stands for Search Engine Optimization. It refers to the process of improving a website’s visibility in search engine results pages (SERPs) to attract more organic (non-paid) traffic.

In simple terms, SEO involves making changes to a website’s structure, content, and performance to make it easier for search engines like Google, Bing, or Yahoo to find, understand, and rank it higher when users search for relevant terms.



Figure 1-1: SEO

**1.4.2 Roles and Types of SEO**

1. Roles of SEO:

SEO plays a vital role in website success by:

* Improving Visibility:
  + SEO helps a website appear higher in search engine results, making it easier for potential customers to find the site.
* Driving Organic Traffic:
  + A well-optimized website attracts more visitors without relying on paid ads.
* Enhancing User Experience (UX):
  + SEO involves optimizing page speed, mobile responsiveness, and structure — all of which improve how users experience the site.
* Building Trust and Credibility:
  + Websites that rank higher are often seen as more reliable and trustworthy by users.
* Boosting Engagement and Conversions:
  + Higher visibility and better UX increase the chances of users staying on the site and making a purchase or inquiry.

2. Types of SEO:

* On-Page SEO:

Involves optimizing elements within the website, such as:

* Keywords in content
* Title tags, meta descriptions
* Internal links and image alt text
* Clean and semantic HTML structure
* Off-Page SEO:

Focuses on building credibility and authority from outside the website:

* Backlinks (links from other sites to yours)
* Social media shares
* Online reviews and brand mentions
* Technical SEO:

Deals with the technical setup of the website:

* Site speed optimization
* Mobile-friendliness
* Secure connection (HTTPS)
* XML sitemaps and structured data
* Local SEO:

Optimizes a website for local searches:

* Useful for businesses with a physical location
* Includes setting up Google Business Profile and local keywords

**1.4.3 How Are Search Engine Rankings Established?**

Search engine rankings are determined by complex algorithms used by platforms like Google, which evaluate many different ranking factors to decide which websites should appear first in search results.

Key Factors That Influence Rankings:

* Content Quality and Relevance
  + Search engines look for content that is original, useful, and answers users' search intent.
  + Use of relevant keywords, headings, and informative structure is important.
* Backlinks (External Links)
  + The number and quality of backlinks pointing to a site act as a “vote of confidence.”
  + The more trusted and relevant the linking sites are, the better for SEO.
* Page Load Speed
  + Faster websites rank better. Google considers page speed a direct ranking factor.
* Mobile-Friendliness
  + Google uses mobile-first indexing, which means it prioritizes the mobile version of a website for ranking.
* User Experience (UX) Signals
  + Metrics like bounce rate, time on site, and click-through rate tell search engines whether users find the content helpful.
* Technical SEO Setup
  + Clean URLs, secure connection (HTTPS), proper HTML tags, and structured data help search engines crawl and index the site effectively.
* Domain Authority
  + This is an overall score (not officially used by Google but widely referenced) based on age, popularity, and size of the website.

**1.4.4 SEO Standard Website Design Criteria**

Key SEO Design Criteria:

* Responsive Design
  + The website should adapt to all screen sizes (desktop, tablet, mobile).
  + Mobile-friendly sites are ranked higher due to Google’s mobile-first indexing.
* Clean and Semantic HTML Structure
  + Use correct tags like <h1>, <h2>, <p>, <nav> to structure content.
  + Helps search engines understand page layout and hierarchy.
* Fast Loading Speed
  + Compress images, minify CSS/JS, and use efficient hosting.
  + Users stay longer on fast sites, and Google rewards this with better rankings.
* Optimised URL Structure
  + URLs should be short, readable, and include keywords (e.g., /mens-fashion/shirts).
  + Avoid long, messy URLs with random characters.
* Accessible Navigation
  + Use menus and internal links that help both users and search engines easily find pages.
  + Include a sitemap and breadcrumbs where appropriate.
* Title Tags and Meta Descriptions
  + Each page must have a unique title and description using relevant keywords.
  + This is what appears in Google search results and influences click-through rates.
* Alt Text for Images
  + Describe images using alt attributes to improve accessibility and help Google index them.
* Secure Connection (HTTPS)
  + SSL certificates are essential. Google favors secure sites.
* Content-to-Code Ratio
  + Avoid excessive code or scripts; ensure the main content is quickly accessible to crawlers.

**1.4.5 The Influence of Search Engines on Website Performance**

Search engines, especially Google, not only determine how visible a website is but also how it should perform to meet user expectations. The way a website is ranked directly affects how much traffic it receives — and performance plays a big role in that ranking.

How Ranking Factors Affect Website Performance:

* Page Loading Speed
  + Faster pages get better rankings. Users leave slow websites quickly (this increases bounce rate, which negatively impacts SEO).
  + Google has confirmed that site speed is a ranking signal for both desktop and mobile.
  + Tools like Google PageSpeed Insights and GTmetrix can help identify speed issues.
* Mobile Performance
  + Websites must be responsive and optimized for mobile devices.
  + Poor mobile experience can lead to lower rankings under Google’s mobile-first indexing approach.
* Core Web Vitals
  + Google uses Core Web Vitals (real-world performance metrics) to assess:
  + Loading (LCP) – how fast the main content appears
  + Interactivity (FID) – how quickly the page responds to user input
  + Visual Stability (CLS) – how much the layout shifts as it loads
* Search Engine Crawling Efficiency
  + A poorly optimized website may take too long to load or use unnecessary code/scripts, making it harder for search engine bots to crawl and index the site properly.
* UX and SEO Connection
  + Good SEO design enhances user experience — clean layout, intuitive navigation, and engaging content keep users on the site longer.
  + The more time users spend, the more positive the user behavior signals sent to Google.

**1.4.6 Examples Demonstrating How to Improve Search Rankings Through SEO**

To boost a website's visibility on search engines, developers and content creators can apply a variety of SEO techniques. Below are some real-world, practical examples that show how to improve search rankings effectively:

1. Optimize Page Speed

* Example: Compressing high-resolution fashion images using tools like TinyPNG or WebP reduces file size and improves loading time.
* Result: Faster site speeds increase user retention and improve rankings, especially on mobile devices.

2. Use Targeted Keywords in Content

* Example: A men’s fashion sales website includes keywords like “buy men’s slim fit jeans” or “affordable men’s jackets” in product titles, meta descriptions, and headings
* Result: Improves relevance for specific searches, helping pages rank higher for targeted queries.

3. Build Quality Backlinks

* Example: The site collaborates with popular fashion bloggers who link to specific products or blog posts.
* Result: Increases the website’s domain authority and credibility in the eyes of search engines.

4. Improve Mobile Usability

* Example: Redesigning the layout with responsive CSS so users can easily browse categories and products on their phones.
* Result: Meets Google’s mobile-first standards, boosting ranking in mobile searches.

5. Use Descriptive Alt Text for Images

* Example: Instead of using “image1.jpg”, the website uses alt="Men's navy slim fit suit – front view" for product photos.
* Result: Enhances accessibility and helps Google index images for image-based searches.

6. Fix Broken Links and Improve Internal Linking

* Example: Regularly checking for 404 errors and ensuring all products link back to related categories.
* Result: Enhances crawlability and keeps users navigating within the site, reducing bounce rate.

7. Create High-Quality Blog Content

* Example: Posting fashion tips, seasonal style guides, and “How to Wear” articles that include relevant keywords.
* Result: Attracts traffic through long-tail keyword searches and increases time spent on the site.

**CHAPTER 2: Categorize website technologies, tools and software used to develop websites (LO2)**

**2.1 Discuss the capabilities and relationships between front-end and back-end website technologies and explain how these relate to presentation and application layers. (P3)**

**2.1.1 Front-End Technologies: HTML, CSS, JavaScript, ReactJS, AngularJS**

Front-end technologies are responsible for what users see and interact with on a website. These technologies run in the web browser and focus on presentation, layout, and user interaction.

1. HTML (HyperText Markup Language)



Figure 2-1:HTML

* Advantages:
  + Easy to learn and use
  + Essential for structuring web content
  + Supported by all browsers
* Disadvantages:
  + Only handles structure, not style or logic
  + Static – can't create dynamic content on its own

2. CSS (Cascading Style Sheets)



Figure 2-2: CSS

* Advantages:
  + Controls the visual design (colors, layout, fonts)
  + Allows responsive design (e.g., mobile-friendly pages)
  + Separates content from presentation
* Disadvantages:
  + Can become complex in large projects
  + Browser compatibility issues may arise

3. JavaScript (JS)

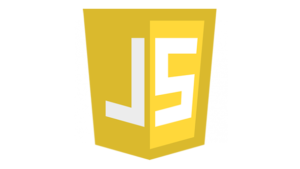


Figure 2-3: Javascript

* Advantages:
  + Enables interactive features (e.g., sliders, dropdowns, form validation)
  + Fast and runs in the browser
  + Large community and libraries available
* Disadvantages:
  + Can be misused and create security vulnerabilities
  + Code may behave differently across browsers

4. ReactJS (JavaScript Library)

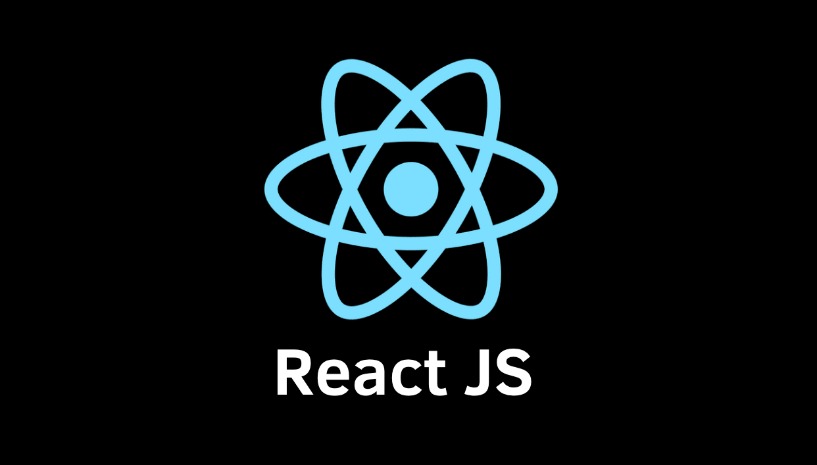


Figure 2-4: ReactJS

* Advantages:
  + Component-based: promotes reusable UI elements
  + Fast performance using virtual DOM
  + Easy integration with other tools/libraries
* Disadvantages:
  + Only covers UI layer — needs additional tools for full app
  + Steep learning curve for beginners

5. AngularJS (JavaScript Framework)



Figure 2-5: AngularJS

* Advantages:
  + Complete front-end framework (includes routing, forms, services)
  + Two-way data binding simplifies dynamic content
  + Backed by Google and well-documented
* Disadvantages:
  + Heavy framework – slower initial load time
  + Complex syntax; not beginner-friendly

**2.1.2 Back-End Technologies: PHP, Java, Node.js**

Back-end technologies handle the server-side logic — including data processing, authentication, database communication, and business rules. This part of the website is not visible to users but is essential for dynamic functionality.

1. PHP (Hypertext Preprocessor)

* Advantages:
  + Open-source and widely used
  + Well-suited for web development, especially with platforms like WordPress
  + Easy to learn for beginners
  + Large community and plenty of documentation
* Disadvantages:
  + Slower compared to newer back-end languages
  + Older codebases can become messy and hard to maintain
  + Less suitable for large-scale enterprise applications

2. Java

* Advantages:
  + Strongly typed and object-oriented — ideal for large, complex systems
  + Platform-independent (write once, run anywhere)
  + Used in enterprise applications, Android apps, and banking systems
  + Secure and scalable
* Disadvantages:
  + More verbose and complex to write
  + Slower development time compared to scripting languages like Python or Node.js
  + Requires more memory and processing resources

3. Node.js

* Advantages:
  + Fast and lightweight — uses non-blocking I/O
  + Based on JavaScript, so front-end and back-end use the same language
  + Large ecosystem of packages (via npm)
  + Real-time data handling (e.g., chat apps)
* Disadvantages:
  + Not ideal for CPU-heavy applications
  + Callback hell in complex code (though modern syntax like async/await helps)
  + Less structured than frameworks like Django or Spring

**2.1.3 How Front-End and Back-End Technologies Relate to Presentation and Application Layers**

To understand this, we connect web development technologies to the OSI model, specifically the Presentation Layer and Application Layer.

Role of the Presentation and Application Layers:

Figure 2-1: Role of the Presentation and Application Layers

| **Layer** | **Purpose** |
| --- | --- |
| **Presentation Layer** | Deals with how data is displayed and formatted for the user (e.g., visuals, UI/UX). |
| **Application Layer** | Interfaces directly with end-user applications and handles the logic of data communication, access, and delivery. |

In web development:

* Front-end technologies relate to the Presentation Layer.
* Back-end technologies relate to the Application Layer.

Front-End & the Presentation Layer

* + Technologies like HTML, CSS, JavaScript, ReactJS, and AngularJS define the look, layout, and interactivity of the website.
  + They transform raw data into a user-friendly interface.
  + Example: When a user views a product page, ReactJS dynamically renders the product’s name, image, and price on the screen.

Back-End & the Application Layer

* + Technologies like PHP, Java, and Node.js handle the core functionality — such as processing login data, retrieving product information, or handling checkout.
  + These technologies ensure data is processed correctly and securely behind the scenes.
  + Example: When a user adds a product to the cart, Node.js communicates with the database to update the cart items.

Summary of the Relationship:

Figure 2-2: Summary of the Relationship

| **Layer** | **Related Technologies** | **Responsibilities** |
| --- | --- | --- |
| Presentation Layer | HTML, CSS, JS, ReactJS, AngularJS | Displaying content, styling, UI interactivity |
| Application Layer | PHP, Java, Node.js, Python, Ruby | Processing logic, interacting with databases/APIs |

**2.2 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).(P4)**

**2.2.1 Definition, advantages and disadvantages of Tools: WordPress, wix.com, Joomla, CoffeeCup**

1. WordPress



Figure 2-6: Wordpress

Definition: WordPress is a content management system (CMS) that allows users to create and manage websites easily. It is open-source and highly customizable, with a vast library of themes and plugins.

* Advantages:
  + User -Friendly: Intuitive interface that is easy for beginners.
  + Flexibility: Highly customizable with thousands of themes and plugins.
  + SEO-Friendly: Built-in features and plugins to enhance search engine optimization.
  + Community Support: Large community for support and resources.
* Disadvantages:
  + Maintenance: Requires regular updates and maintenance.
  + Security Risks: Being popular makes it a target for hackers.
  + Learning Curve: Can be overwhelming for complete beginners due to its extensive features.

2. Wix.com



Figure 2-7: Wix

Definition: Wix.com is a cloud-based website builder that allows users to create HTML5 websites through drag-and-drop tools. It is designed for users with little to no coding experience.

* Advantages:
  + Ease of Use: Drag-and-drop interface makes it simple to design.
  + Templates: Offers a wide range of professionally designed templates.
  + Hosting Included: Provides hosting services, simplifying the process.
* Disadvantages:
  + Limited Customization: Less flexibility compared to open-source platforms like WordPress.
  + Cost: Can become expensive with premium features.
  + SEO Limitations: Some users find SEO capabilities less robust than other platforms.

3. Joomla



Figure 2-8: Joomla

Definition: Joomla is another open-source content management system that is used for building websites and online applications. It is known for its flexibility and powerful features.

* Advantages:
  + Flexibility: Suitable for complex websites and applications.
  + User Management: Advanced user management features.
  + Multilingual Support: Built-in support for multiple languages.
* Disadvantages:
  + Steeper Learning Curve: More complex than WordPress, which can be challenging for beginners.
  + Less Community Support: Smaller community compared to WordPress.
  + Customization Complexity: Requires more technical knowledge for customization.

4. CoffeeCup



Figure 2-9: CoffeeCup

Definition: CoffeeCup is a web design software that allows users to create websites using a visual editor or by writing code. It is more suited for users who have some coding knowledge.

* Advantages:
  + Code Editing: Offers a robust code editor for advanced users.
  + Responsive Design: Tools for creating responsive websites.
  + No Internet Required: Can be used offline.
* Disadvantages:
  + Limited Templates: Fewer pre-designed templates compared to other platforms.
  + Learning Curve: Requires some coding knowledge, which may deter beginners.
  + Cost: Not free; requires a purchase for full features.

**2.2.2 Example**

Steps to Create a Website

**WordPress:**

1. Choose a Hosting Provider & Domain Name

- Pick a reliable hosting provider (like Bluehost, SiteGround, or Hostinger).

- Register a domain name (e.g., www.yourwebsite.com).

2. Install WordPress

- Many hosting providers offer one-click WordPress installation.

- Follow the installation process and set up your admin account.

3. Select & Customize a Theme

- Browse free and premium themes under Appearance > Themes.

- Install a theme and customize it to match your brand.

4. Install Essential Plugins

- Go to Plugins > Add New and install must-have plugins like:

- Yoast SEO (for search optimization)

- Elementor (for easy drag-and-drop design)

- WPForms (for contact forms)

5. Create Key Pages

- Set up essential pages such as Home, About, Services, Blog, and Contact.

- Use the WordPress Block Editor or page builder to design each page.

6. Configure Website Settings

- Under Settings, adjust:

- Site title & tagline

- Permalinks for SEO-friendly URLs

- Reading settings for homepage display

7. Publish & Promote

- Once satisfied, hit Publish to make your website live.

- Share on social media and optimize content for search engines.

**Wix.com**

1. Sign Up & Choose a Plan

- Create an account on Wix.com.

- Pick a free or premium plan depending on your needs.

2. Choose a Website Type & Template

- Wix offers AI-powered design (Wix ADI) or manual editing (Wix Editor).

- Select a pre-made template or let Wix ADI create a site for you.

3. Customize Your Website

- Use the drag-and-drop editor to:

- Change colors, fonts, and layout

- Add images, text, and animations

- Modify header & footer sections

4. Add Essential Features

- Install Wix Apps like:

- Wix Blog (for blogging)

- Wix Stores (for e-commerce)

- Wix Forms (for collecting customer info)

5. Optimize Your Site

- Go to Settings to:

- Set up SEO (Wix SEO Wiz helps you optimize content)

- Connect a custom domain (for premium users)

- Enable mobile responsiveness

6. Publish & Promote

- Click Publish to make your website live.

- Share it on social media and start marketing.

**Joomla**

1. Choose a Hosting Provider & Install Joomla

- Get a Joomla-compatible hosting provider (SiteGround, Hostinger, Bluehost).

- Use cPanel or a one-click installer to set up Joomla.

2. Configure Basic Settings

- Set up your site name, admin user, and database during installation.

- Access the Joomla dashboard at yourdomain.com/administrator.

3. Select & Customize a Template

- Find Joomla templates in Extensions > Templates.

- Install a template and customize its layout, fonts, and colors.

4. Install Essential Extensions

- Go to Extensions > Manage > Install to add functionality like:

- JCE Editor (better content management)

- Akeeba Backup (automatic site backups)

- SP Page Builder (drag-and-drop website editing)

5. Create & Organize Content

- Use Articles to build pages like Home, About, Services, and Contact.

- Organize content using Categories & Menus.

6. Optimize & Secure Your Website

- Configure SEO settings under Global Configuration.

- Install security extensions like Admin Tools to prevent hacks.

7. Publish & Maintain

- Launch your website and promote it.

- Keep Joomla updated for security & performance improvements.

**CoffeeCup**

1. Choose Your CoffeeCup Tool

- HTML Editor: Great for manual coding.

- Responsive Site Designer: Drag-and-drop design without coding.

- Site Mapper & Form Builder: Extra tools for site functionality.

2. Set Up Your Project

- Open CoffeeCup and create a new project.

- Choose a template or start from scratch.

3. Customize Your Website

- Modify fonts, colors, and layouts with the design panel.

- Use the grid-based system for mobile responsiveness.

- Add content like text, images, and videos.

4. Enhance Your Website

- Use CSS styles for custom formatting.

- Integrate JavaScript for animations.

- Build forms with CoffeeCup Form Builder.

5. Preview & Test

- Check responsiveness with Live Preview.

- Test on different devices before publishing.

6. Publish Your Site

- Export your website files.

- Upload to a hosting provider via FTP (using CoffeeCup Direct FTP).

- Make adjustments and keep your site updated.

**2.2.3 Comparison**

Table 2-3: Comparison

| **Feature/Tool** | **WordPress** | **Wix.com** | **Joomla** | **CoffeeCup** |
| --- | --- | --- | --- | --- |
| **Type** | Open-source CMS | Cloud-based website builder | Open-source CMS | Web design software |
| **Ease of Use** | Moderate (requires some learning) | Very easy (drag-and-drop) | Moderate (steeper learning curve) | Moderate (requires coding knowledge) |
| **Customization** | Highly customizable (themes/plugins) | Limited customization | Flexible but complex | Customizable (code editing) |
| **Templates** | Thousands available | Wide range of templates | Fewer templates available | Limited templates |
| **SEO Features** | Strong SEO capabilities | Basic SEO tools | Good SEO features | Basic SEO capabilities |
| **Hosting** | Requires separate hosting | Hosting included | Requires separate hosting | Requires separate hosting |
| **Cost** | Free (with paid options) | Free and premium plans | Free (with paid extensions) | Paid software |
| **Support** | Large community support | Good customer support | Smaller community support | Limited support |
| **Best For** | Blogs, portfolios, e-commerce | Small businesses, personal sites | Complex websites, online applications | Users with coding knowledge |

**2.3 Analyze a range of tools and techniques available to design and develop a custom- built website. (M3)**

**2.3.1 Advantages and disadvantages of IDE Frontend, Backend, database management tool**

1. Frontend IDEs

a. Visual Studio Code

* Advantages:
  + Lightweight and Fast: Quick to load and responsive.
  + Extensive Extensions: A vast library of extensions for added functionality.
  + Integrated Git Support: Built-in version control features.
  + Cross-Platform: Available on Windows, macOS, and Linux.
* Disadvantages:
  + Learning Curve: Can be overwhelming for beginners due to numerous features.
  + Performance Issues: May slow down with too many extensions.

b. Sublime Text

* Advantages:
  + Speed: Extremely fast and responsive.
  + Distraction-Free Mode: Allows for focused coding without distractions.
  + Customizable: Highly customizable with themes and plugins.
* Disadvantages:
  + Cost: The free version has a nag screen; full version requires a purchase.
  + Limited Built-in Features: Lacks some advanced features found in other IDEs.

c. Notepad++

* Advantages:
  + Free and Open Source: No cost and community-supported.
  + Lightweight: Minimal resource usage.
  + Syntax Highlighting: Supports multiple programming languages.
* Disadvantages:
  + Limited Features: Lacks advanced features like debugging and version control.
  + Windows Only: Not available on macOS or Linux.

2. Backend IDEs

a. NetBeans

* Advantages:
  + Multi-Language Support: Supports Java, PHP, and more.
  + Integrated Tools: Built-in tools for debugging and profiling.
  + Free and Open Source: No cost for users.
* Disadvantages:
  + Performance: Can be slow and resource-intensive.
  + Complexity: May be overwhelming for beginners.

b. Visual Studio (C#)

* Advantages:
  + Powerful Features: Advanced debugging, profiling, and testing tools.
  + Integration with Azure: Seamless integration with Microsoft Azure services.
  + Rich Ecosystem: Extensive libraries and frameworks available.
* Disadvantages:
  + Cost: The full version can be expensive; free version has limitations.
  + Windows Only: Primarily designed for Windows development.

c. Eclipse

* Advantages:
  + Extensible: A wide range of plugins available for various languages.
  + Strong Community Support: Large community and resources available.
  + Free and Open Source: No cost for users.
* Disadvantages:
  + Complex Interface: Can be confusing for new users.
  + Performance Issues: May be slow with large projects.

3. Database Management Tools

a. MS SQL Server

* Advantages:
  + Robust Performance: High performance for large databases.
  + Security Features: Advanced security and compliance features.
  + Integration with Microsoft Products: Seamless integration with other Microsoft tools.
* Disadvantages:
  + Cost: Licensing can be expensive.
  + Windows-Centric: Primarily designed for Windows environments.

b. MySQL Workbench

* Advantages:
  + Free and Open Source: No cost for users.
  + User -Friendly Interface: Intuitive graphical interface for database management.
  + Cross-Platform: Available on multiple operating systems.
* Disadvantages:
  + Limited Features: May lack some advanced features found in other database tools.
  + Performance Issues: Can be slow with very large datasets.

c. Oracle Database

* Advantages:
  + High Scalability: Suitable for large enterprise applications.
  + Advanced Features: Offers advanced features like clustering and partitioning.
  + Strong Security: Robust security features for data protection.
* Disadvantages:
  + Cost: Licensing can be very expensive.
  + Complexity: Steeper learning curve for new users.

**2.3.2 Advantages and disadvantages of frameworks, database servers**

1. Frameworks

a. ASP.NET

* Advantages:
  + Performance: High performance due to compiled code and optimized runtime.
  + Security Features: Built-in security features to protect against common vulnerabilities.
  + Integration with Microsoft Ecosystem: Seamless integration with other Microsoft products and services.
* Disadvantages:
  + Windows-Centric: Primarily designed for Windows environments, which may limit deployment options.
  + Learning Curve: Can be complex for beginners, especially with advanced features.

b. Laravel

* Advantages:
  + Elegant Syntax: Clean and expressive syntax that enhances developer productivity.
  + Built-in Features: Comes with many built-in features like authentication, routing, and caching.
  + Strong Community Support: Large community and extensive documentation available.
* Disadvantages:
  + Performance Overhead: May have performance overhead compared to lighter frameworks.
  + Learning Curve: Some concepts may be challenging for beginners.

c. Spring Hibernate

* Advantages:
  + Modular Architecture: Allows for building complex applications with modular components.
  + Integration with Java: Seamless integration with Java applications and libraries.
  + Robust Transaction Management: Strong support for transaction management.
* Disadvantages:
  + Complex Configuration: Can be complex to set up and configure.
  + Steep Learning Curve: Requires a good understanding of Java and Spring concepts.

2. Database Servers

a. MySQL

* Advantages:
  + Open Source: Free to use and widely supported.
  + High Performance: Efficient for read-heavy applications.
  + Strong Community Support: Large community and extensive documentation.
* Disadvantages:
  + Limited Features: May lack some advanced features found in commercial databases.
  + Scalability Issues: Can face challenges with very large datasets or high write loads.

b. SQL Server

* Advantages:
  + Robust Performance: High performance for large-scale applications.
  + Advanced Features: Offers features like data warehousing and business intelligence.
  + Strong Security: Advanced security features for data protection.
* Disadvantages:
  + Cost: Licensing can be expensive.
  + Windows-Centric: Primarily designed for Windows environments.

c. MongoDB

* Advantages:
  + NoSQL Database: Flexible schema design allows for handling unstructured data.
  + Scalability: Easily scalable for large datasets and high traffic.
  + High Performance: Optimized for high write loads and real-time analytics.
* Disadvantages:
  + Learning Curve: Requires a different mindset compared to traditional SQL databases.
  + Data Consistency: Eventual consistency model may not be suitable for all applications.

**REFERENCE**

Soanes, C. and Stevenson, A., 2008. Concise Oxford English Dictionary. 11th ed. Oxford: Oxford University Press.

Shelly, G.B. and Vermaat, M.E., 2012. Discovering Computers: Complete. 1st ed. Boston: Cengage Learning, p.83.

Fielding, R.T. (2000). Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine.

Silberschatz, A., Galvin, P.B. and Gagne, G. (2018). Operating System Concepts. 10th edn. Hoboken, NJ: Wiley.

Koch, C. (2017). Apache, Nginx, and IIS: A Comparative Study of Web Server Performance. 1st edn. New York: Springer.

Clarke, A., 2010. Search Engine Optimization: Your Visual Blueprint for Effective Internet Marketing. 2nd ed. Indianapolis: Wiley, p. 184.