

Higher Nationals – Summative Assignment Feedback Form

Student Name/ID	MOHAMMED AASHIK /E230667		
Unit Title	Unit 13: Web Design and Development		
Assignment Number	1	Assessor	
Submission Date	11/01/2025	Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	

Assessor Feedback:

LO1 Explain server technologies and management services associated with hosting and managing websites

Pass, Merit & Distinction Descripts	P1 <input type="checkbox"/>	P2 <input type="checkbox"/>	M1 <input type="checkbox"/>	M2 <input type="checkbox"/>	D1 <input type="checkbox"/>
-------------------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

LO2 Categorise website technologies, tools and software used to develop websites

Pass, Merit & Distinction Descripts	P3 <input type="checkbox"/>	P4 <input type="checkbox"/>	M3 <input type="checkbox"/>
-------------------------------------	-----------------------------	-----------------------------	-----------------------------

LO3 Utilise website technologies, tools and techniques with good design principles to create a multipage website

Pass, Merit & Distinction Descripts	P5 <input type="checkbox"/>	P6 <input type="checkbox"/>	M4 <input type="checkbox"/>	D2 <input type="checkbox"/>
-------------------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

LO4 Create and use a Test Plan to review the performance and design of a multipage website

Pass, Merit & Distinction Descripts	P7 <input type="checkbox"/>	M5 <input type="checkbox"/>	D3 <input type="checkbox"/>
-------------------------------------	-----------------------------	-----------------------------	-----------------------------

* Please note that grade decisions are provisional. They are only confirmed once internal and external moderation has taken place and grades decisions have been agreed at the assessment board.

Assessor Feedback:

Grade:	Assessor Signature:	Date:
---------------	----------------------------	--------------

Resubmission Feedback:

- Please note resubmission feedback is focussed only on the resubmitted work

Grade:	Assessor Signature:	Date:
---------------	----------------------------	--------------

Internal Verifier's Comments:**Signature & Date:****Important Points:**

1. It is strictly prohibited to use textboxes to add texts in the assignments, except for the compulsory information. eg: Figures, tables of comparison etc. Adding text boxes in the body except for the before mentioned compulsory information will result in rejection of your work.
2. Avoid using page borders in your assignment body.
3. Carefully check the hand in date and the instructions given in the assignment. Late submissions will not be accepted.
4. Ensure that you give yourself enough time to complete the assignment by the due date.
5. Excuses of any nature will not be accepted for failure to hand in the work on time.
6. You must take responsibility for managing your own time effectively.

7. If you are unable to hand in your assignment on time and have valid reasons such as illness, you may apply (in writing) for an extension.
8. Failure to achieve at least PASS criteria will result in a REFERRAL grade.
9. Non-submission of work without valid reasons will lead to an automatic RE FERRAL. You will then be asked to complete an alternative assignment.
10. If you use other people's work or ideas in your assignment, reference them properly using HARVARD referencing system to avoid plagiarism. You have to provide both in-text citation and a reference list.
11. If you are proven to be guilty of plagiarism or any academic misconduct, your grade could be reduced to A REFERRAL or at worst you could be expelled from the course
12. Use word processing application spell check and grammar check function to help editing your assignment.
13. Use **footer function in the word processor to insert Your Name, Subject, Assignment No, and Page Number on each page.** This is useful if individual sheets become detached for any reason.

STUDENT ASSESSMENT SUBMISSION AND DECLARATION

When submitting evidence for assessment, each student must sign a declaration confirming that the work is their own.

Student name: MOHAMMED AASHIK		Assessor name:
Issue date:	Submission date: 11/01/2025	Submitted on: 11/01/2025
Programme: HND in Computing		
Unit 13: Web Design and Development		
Assignment number and title: 1 Web application for Velvet Vogue clothing store		

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 penalized. 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

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 signature: E230667

Date: 30.12.2024

Assignment Brief

Student Name/ID Number	MOHAMMED MAHROOF MOHAMMED AASHIK/E230667
Unit Number and Title	Unit 13: Web Design and Development
Academic Year	2024/2025
Unit Tutor	
Assignment Title	Web application for Velvet Vogue clothing store
Issue Date	
Submission Date	11/01/2025
Submission Format	
<p>A written portfolio that explores and highlights key information on the technologies that could be utilised to develop, host and manage the website for Velvet Vogue.</p> <p>The portfolio should consist of the following components. (Activity 2)</p>	

- **Solution design documents** produced to meet the requirements of a given problem, including requirements specifications, visual and technical design documentation details of additional considerations
- **Development documentation**, including records of test plan and review processes, and refinements made to the website throughout development.

The recommended word limit is 2000–2,500 words, although you will not be penalised for exceeding the total word limit.

An Evaluative report to evaluate the design and development process of the website. (Activity 1 and 3)

The recommended word limit is 1,500–2,000 words, although you will not be penalised for exceeding the total word limit.

You are required to make use of headings, paragraphs and subsections as appropriate, and all work must be supported with research and referenced using the Harvard referencing system.

Unit Learning Outcomes

LO1 Explain server technologies and management services associated with hosting and managing websites

LO2 Categorise website technologies, tools and software used to develop websites

LO3 Utilise website technologies, tools and techniques with good design principles to create a multipage website

LO4 Create and use a Test Plan to review the performance and design of a multipage website.

Transferable skills and competencies developed

Computing-related cognitive skills :

- Demonstrate knowledge and understanding of essential facts, concepts, principles and theories relating to computing and computer applications
- Recognise and analyse criteria and specifications appropriate to specific problems, and plan strategies for their solutions

- Critical evaluation and testing: analyse the extent to which a computer-based system meets the criteria defined for its current use and future development
- Methods and tools: deploy appropriate theory, practices and tools for the design, implementation and evaluation of computer-based systems.

Computing-related practical skills :

- The ability to specify, design and construct reliable, secure and usable computer-based systems
- The ability to evaluate systems in terms of quality attributes and possible trade-offs presented within the given problem
- The ability to deploy effectively the tools used for the construction and documentation of computer applications, with particular emphasis on understanding the whole process involved in the effective deployment of computers to solve practical problems.

Generic skills for employability :

- Intellectual skills: critical thinking; making a case; numeracy and literacy
- Contextual awareness, e.g. the ability to understand and meet the needs of individuals, business and the community, and to understand how workplaces and organisations are governed).

Assignment activity and guidance:

Activity 1 – Evaluative report

John Finlo is an ambitious entrepreneur who has recently started a clothing store called "Velvet Vogue". Specializing in trendy casualwear and formal wear, that offers a vast selection of clothing and accessories for young adults who want to express their identity through their style.

John is looking for new ways to promote his business by enhancing the online presence of his brand. So he has approached you to develop a multi-page (initially between four and six pages) website that will act as a showcase of his products categories, allow customers to make transactions, inquiries and provide a seamless shopping experience. The website will need to provide the following features:

- A visually appealing home page that showcases the products, promotions, and new arrivals. The home page should include clear navigation to different sections of the website. (Product categories, the shopping cart, customer support/inquiry page etc.)
- Product pages that display detailed information about each item, including high-quality images, descriptions, available sizes, colors, and pricing.
- A shopping cart that includes add, review, adjust quantities and proceed payments of the products they select.
- Product categories and filters that allow users to search items based on criteria such as gender, clothing type, size, price range and etc.
- User accounts and profiles that allow customers to create personal accounts, admin accounts to add new products to the web site.
- Responsive design that ensures the website accessibility across different devices

Your task is to design and develop an e-commerce website that reflects his brand identity and satisfy the preferences and needs of its target audience. Develop a web-based solution for the above scenario and produce a report covering the following tasks.

Write an evaluative report that explores the technologies that could be utilized to develop, host and manage the website for Velvet Vogue. The report should include the following.

- Identification of the purpose and different types of DNS
- An explanation of how domain names are organized and managed
- An explanation of the relationships between hardware and software used to design, publish and access the Velvet Vogue website, with consideration of:
 - communication protocols (e.g. HTTPS, FTP)
 - server hardware
 - operating systems
 - Web server software (e.g. Apache, TomCat, IIS, Nginx).
- An analysis of how web development technologies and frameworks (e.g. HTML5, JSON, JQuery, REACT, Angular, Node.js) have had an impact on web design, functionality and management
- A review of the influence of search engines on website performance

- A review, supported by clear, well-documented evidence, of how to improve the index value of a website through SEO
- A discussion of the capabilities and relationships of front-end and back-end website technologies and how they relate to the presentation and application layers
- An analysis of the impact of different tools and techniques available on custom-built website design and development
- A discussion of the differences between online website creation tools and custom-built sites with consideration of:
 - design flexibility
 - performance
 - functionality
 - User Experience (UX)
 - User Interface (UI).

Activity 2 -portfolio

Design, Develop and Test the Website

You are to design a multi-page website solution that will meet the needs of the Velvet Vogue website.

Your design documents should include:

- a full set of client and user requirements, e.g. functional and non-functional requirements, user acceptance criteria
- wireframes
- interface designs
- details of:
 - networking and hosting considerations
 - security features
 - how accessibility and inclusivity guidelines will be implemented.

You should then use your design documents to implement a multi-page website that will meet the needs of Velvet Vogue.

Your implementation should include the following.

- A branded, multi-page website for Velvet Vogue, supported with realistic content from your additional research
- The creation of a suitable test plan for the site. The test plan should identify key areas that cover both the functionality and performance of the developed website
- An explanation of the web design principles, standards and guidelines used
- A justification of the implementation of the multi-page website decisions against the planned design decisions, noting any similarities or differences
- An implementation of the test plan against the final developed multi-page website

- Analysis of the QA process, including a review of how it was implemented during the design and development stages of the multi-page website for Velvet Vogue.

Activity 3 – Evaluative report contd.

Evaluation

Produce an evaluative report that explores the design and development of the website, and how well the final website meets the needs of Velvet Vogue.

The report should include the following.

- Justification of the chosen management services, tools and software from the range of options presented in Activity 1 that were used ultimately in the development of the Velvet Vogue website
- An evaluation of the design and development process of making the Velvet Vogue website, compared to the original design document
- An evaluation of any specific technical challenges faced creating the Velvet Vogue website
- An evaluation of the results of the Test Plan and the overall success of the website
- A list of recommended improvements for the Velvet Vogue website; this could be based on client feedback and the results of testing.

Recommended resources.

Weblinks:

<https://developer.chrome.com/> (n.d.) *Chrome DevTools* [online] Available at: <https://developer.chrome.com/docs/devtools/> [Accessed 1 August 2022]

<https://developers.google.com/> (2022) *Search Engine Optimization (SEO) Starter Guide* [online] Available at: <https://developers.google.com/search/docs/beginner/seo-starter-guide> [Accessed 1 August 2022]

<https://getbootstrap.com/> (n.d.) *Build fast, responsive sites with Bootstrap* [online] Available at: <https://getbootstrap.com/> [Accessed 1 August 2022]

Learning Outcomes and Assessment Criteria

<https://ibrandstudio.com/> (n.d.) *Design Theory Principles That Work When Building A Website* [online] Available at: <https://ibrandstudio.com/articles/building-website-design-theory-principles> [Accessed 1 August 2022]

<https://wishdesk.com/> (2020) *QUALITY ASSURANCE FOR YOUR WEB PROJECT: STAGES, METHODS, & PRACTICES* [online] Available at: <https://wishdesk.com/blog/quality-assurance-stages-steps-tools> [Accessed 1 August 2022]

<https://www.comentum.com/> (2022) *Guide to Web Application Development* [online] Available at: <https://www.comentum.com/guide-to-web-application-development.html> [Accessed 1 August 2022]

<https://www.monocubed.com/> (2022) *List of 10 Best Web Frameworks for Web App Development in 2022* [online] Available at: <https://www.monocubed.com/blog/most-popular-web-frameworks/> [Accessed 1 August 2022]

Reading:

Duckett, J (2011) *HTML and CSS: Design and Build Websites*. 1st edn, Wiley

Duckett, J (2014) *JavaScript and jQuery: Interactive Front End Development*. 1st edn, Wiley

Duckett, J (2022) *PHP & MySQL: Server-Side Web Development*. 1st edn, Wiley

Plekhanova, J. (2009) *Evaluating web development frameworks: Django, Ruby on Rails and CakePHP*, Available at: <https://ibit.temple.edu/wpcontent/uploads/2011/03/IBITWebframeworks.pdf>

Journals:

Divyam. G. Pulkit J. and Bharat, B. (2020) Enhancement of Security using Various Web Development Frameworks. *Proceedings of the International Conference on Innovative Computing & Communications (ICICC) 2020*, SSRN: <https://ssrn.com/abstract=3600740> or <http://dx.doi.org/10.2139/ssrn.3600740>

K. Lei, Y. Ma and Z. Tan, (2014) Performance Comparison and Evaluation of Web Development Technologies in PHP, Python, and Node.js. 2014 IEEE 17th International Conference on Computational Science and Engineering, 2014, pp. 661-668, doi: 10.1109/CSE.2014.142.

Salas-Zárate, M. et al. (2015) Analyzing best practices on Web development frameworks: The lift approach. *Science of Computer Programming*, Volume 102, pp.1-19. doi: <https://doi.org/10.1016/j.scico.2014.12.004>

Visser, E.B. & Weideman, M. (2011) An empirical study on website usability elements and how they affect search engine optimisation. *SA Journal of Information Management* 13(1), Art. #428, 9 pages. doi: 10.4102/sajim.v13i1.42

Wilson, R. et al. (2006) Search engine optimisation: A primer on keyword strategies. *Journal of Direct, Data and Digital Marketing Practice*. doi:10.1057/PALGRAVE.DDDMP.4340563

Pass	Merit	Distinction
LO1 Explain server technologies and management services associated with hosting and managing websites	LO1 & LO2	

<p>P1 Identify the purpose and types of DNS, including explanations on how domain names are organised and managed.</p> <p>P2 Explain the purpose and relationships between communication protocols, server hardware, operating systems and web server software with regard to designing, publishing and accessing a website.</p>	<p>M1 Analyse the impact of common web development technologies and frameworks with regard to website design, functionality and management.</p> <p>M2 Review the influence of search engines on website performance and provide evidence-based support for improving a site's index value and rank through search engine optimisation.</p>	<p>D1 Justify the technologies, management services, tools and software chosen to realise a custom-built website.</p>
<p>LO2 Categorise website technologies, tools and software used to develop websites</p>		
<p>P3 Discuss the capabilities and relationships between front-end and back-end website technologies and explain how they relate to presentation and application layers.</p> <p>P4 Discuss the differences between online website creation tools and</p>	<p>M3 Analyse a range of tools and techniques available to design and develop a custom-built website.</p>	

<p>custom-built sites with regard to design flexibility, performance, functionality, User Experience (UX) and User Interface (UI).</p>		
<p>LO3 Utilise website technologies, tools and techniques with good design principles to create a multipage website</p>		<p>D2 Evaluate the design and development process of the multipage website against the design document including any technical challenges faced.</p>
<p>P5 Create a design document for a branded, multipage website, supported with medium fidelity wireframes and a full set of client and user requirements.</p>	<p>M4 Justify the multipage website implementation decisions against the design document.</p>	
<p>P6 Use the design document with appropriate principles, standards and guidelines to produce a branded, multipage website supported with realistic content.</p>		
<p>LO4 Create and use a Test Plan to review the performance and design of a multipage website.</p>		<p>D3 Evaluate the results of the Test Plan and the overall success of the multipage website with</p>
<p>P7 Create a suitable test plan, identifying key</p>	<p>M5 Analyse the Quality Assurance (QA) process and review how it was implemented during the multipage</p>	

performance areas to review the functionality and performance of the multipage website developed.	website design and development stages.	recommendations for improvement.
---	--	----------------------------------

ACKNOWLEDGEMENT

I am deeply grateful for the assistance and guidance I received from numerous esteemed individuals, which was instrumental in the successful completion of my task. I would like to express my sincere appreciation to ESOFT for providing a conducive workspace that facilitated the completion of my task. I am delighted to announce the successful completion of the assignment. I am particularly indebted to **Mrs. Dilishi** for his invaluable guidance throughout my Second semester assignments. Lastly, I extend my heartfelt gratitude to my family members and classmates whose unwavering support greatly contributed to the timely completion of this project. Thank you all for your immense contribution!

THANK YOU!

M.M.M AASHIK

E230667

Content

Contents

Plagiarism.....	3
Activity 01	
27.12.2024	16
Identification of the Purpose and Different Types of DNS	16
Explanation of How Domain Names Are Organized and Managed	21
1. OSI Model (open system interconnection).....	35
Analysis of the Impact of Web Development Technologies and Frameworks on Web Design, Functionality, and Management	67
A Review of How to Improve the Index Value of a Website through SEO	83
Key Areas of Influence	84

Discussion of the Capabilities and Relationships of Front-End and Back-End Website Technologies and Their Relation to Presentation and Application Layers	88
Front End (Presentation Layer)	88
Back End (Application Layer)	95
Server-Side Scripting Languages:	97
Databases	100
Back-End Designing Tools	100
Online Builders	127
Offline Builders	128
Design Document for a Branded, Multipage Website	136
1. Project Overview	136
2. Client Requirements	137
Performance	140
Scalability	140
Reliability	140
Security	140
Accessibility	140
SEO Optimization	140
Maintainability	141
Usability	141
Localization	141
Integration	141
Energy Efficiency	141
Analytics and Monitoring	141
3. User Requirements	141
Home page wireframe	143
Login wireframe	144
Principles, Standards, and Guidelines Followed	149
Justification of Multipage Website Implementation Decisions against the Design Document	152
1. Alignment with Client and User Requirements	152
2. Adherence to Principles, Standards, and Guidelines	154
Technical challenges faced During development	186

While connecting to a remo	187
Test Plan for Velvet Vogue Multipage Website	188
1. Test Plan Overview	188
3. Execution Plan.....	191
4. Reporting and Metrics.....	192
5. Expected Outcomes.....	192
1. Quality Assurance Process Overview.....	207
2. QA Process Phases During Website Design and Development.....	207
3. Review of QA Implementation during Website Development	211
1. Evaluation of Test Plan Results	212
2. Evaluation of Overall Website Success	215
Success:	216
Issues Found:.....	217
Recommendations:.....	218
Success:	218
Issues Found:.....	219
Recommendation:	219
Success:	219
Issues Found:.....	219
Recommendation:	220
3. Recommendations for Improvement.....	220
Improve Search Functionality:	220
2. Optimize Performance:	221
3. Increase Security Measures:	221
4. Refine UX and UI:.....	221
5. Enhance Compatibility:.....	222
6. Address Accessibility Issues:.....	222
7. Continuous Testing and Monitoring:.....	223
References	227

Activity 01

Evaluative report for Velvet Vogue...

INTRODUCTION

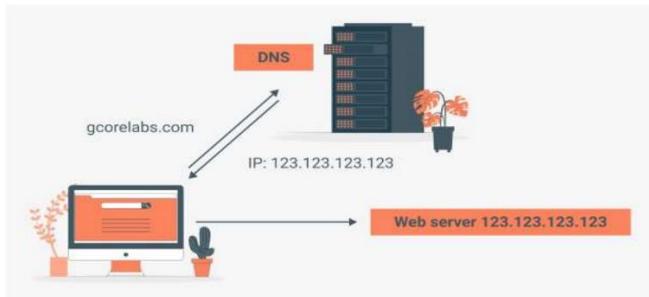
The Velvet Vogue multipage website is an innovative e-commerce platform crafted to highlight a carefully curated collection of trendy casualwear and formalwear for young adults. Combining modern design with robust functionality, the platform delivers an engaging and seamless shopping experience. With user-centric features such as dynamic product filtering, secure checkout, and personalized user profiles, Velvet Vogue caters to both style-conscious shoppers and tech-savvy users seeking convenience and reliability.

The website emphasizes responsive design and accessibility, ensuring optimal performance across devices and providing an inclusive shopping experience for a global audience. Its visually appealing interface incorporates intuitive navigation and engaging elements like dynamic carousels to feature promotions, new arrivals, and popular categories. On product pages, high-quality images, detailed descriptions, and interactive filters allow users to sort items by size, price, and category with ease.

Leveraging advanced technologies, Velvet Vogue combines the power of HTML5 and CSS3 for a clean, responsive front end, React.js for dynamic features like real-time updates, and Node.js for server-side logic. MongoDB efficiently manages product catalogues, user accounts, and order histories, while hosting on AWS ensures scalability and reliability. With a focus on functionality, branding, and user satisfaction, Velvet Vogue sets a benchmark for modern e-commerce platforms.

Identification of the Purpose and Different Types of DNS

DNS (Domain Name Service)



What is DNS?

The Domain Name System (DNS) acts as the "phonebook" of the internet, translating user-friendly domain names into machine-readable IP addresses. When you enter a URL into your web browser, DNS converts this domain name into an IP address that computers use to locate and connect with each other over the internet. Each device connected to the internet has a unique IP address, and DNS simplifies our online experience by removing the need to memorize these numeric addresses.

How DNS Works

DNS Query Process

- ⊕ **Recursive Query:** In a recursive query, the DNS server takes full responsibility for resolving the request. It queries other DNS servers on behalf of the user until it finds the IP address for the domain name. The user receives the resolved IP address directly from the server.
- ⊕ **Iterative Query:** In an iterative query, the DNS server returns a referral to another DNS server if it doesn't have the requested IP address. The user may need to query multiple servers to get the final IP address.
- ⊕ **Non-Recursive Query:** In a non-recursive query, the DNS resolver already knows the answer. It either retrieves the information from its local cache or queries the authoritative name server for a quick response.

Purpose of DNS

Key purposes of DNS

1. **Simplified Access:** Converts domain names into IP addresses to make internet navigation user-friendly.

2. **Decentralization:** Distributes domain name resolution tasks across multiple servers to ensure reliability and reduce traffic bottlenecks.
3. **Scalability:** Supports the growing number of domain names and IP addresses globally.
4. **Redundancy and Reliability:** Provides fallback options to prevent service interruptions in case of server failures.
5. **Load Balancing:** Helps distribute traffic across multiple servers for optimized website performance

Why we need DNS

The **Domain Name System (DNS)** is a critical component of the internet that translates human-readable domain names (like `www.example.com`) into machine-readable IP addresses (such as `192.168.1.1`). Without DNS, the internet would be far less user-friendly and significantly more complex to navigate. Here's why DNS is essential:

1. **Human-Friendly Access:** Remembering numerical IP addresses for each website would be impractical. DNS simplifies this by allowing users to type easy-to-remember domain names instead of long, complex numbers.
2. **Scalability:** The internet consists of billions of websites and devices. DNS organizes and manages this vast amount of information, making it possible to locate and access resources quickly, even as the internet grows.
3. **Decentralization:** DNS operates as a distributed system, meaning no single entity controls all DNS data. This decentralization ensures resilience, reliability, and faster resolution of queries by distributing the workload across multiple servers worldwide.
4. **Dynamic Mapping:** Websites and services can move between servers or change IP addresses. DNS enables these transitions seamlessly, ensuring uninterrupted access for users without requiring them to remember new IP addresses.
5. **Global Connectivity:** DNS bridges the gap between users and servers across the globe, enabling seamless communication and access to websites regardless of geographical location.

6. **Support for Advanced Features:** DNS also supports load balancing, redundancy, and content delivery networks (CDNs), optimizing performance and ensuring website availability.

In essence, DNS acts as the "phonebook" of the internet, providing a scalable, efficient, and user-friendly way to navigate and access the vast resources available online. It's a foundational technology that ensures the internet operates smoothly and effectively.

Types of DNS Servers

➤ The primary server

The main server is an authoritative server for a zone. All zone-related administrative operations, such as adding subdomains and other similar duties, must be completed on the master server. Furthermore, any changes to the zone files, as well as any updates or additions to the RRs in the zone files, must be performed on the primary server. Except when Active Directory and Microsoft DNS Server are linked, each zone has a single main server.

➤ Backup Servers

The primary DNS server is backed up by a secondary DNS server. Zone transfers move all zone files to the secondary server. A zone can have as many secondary servers as it needs for load balancing, fault tolerance, and traffic reduction. Furthermore, one DNS server can serve many zones at the same time. Other DNS server functions, such as cache servers and forwarders, may be employed in addition to primary and secondary DNS servers, depending on the DNS infrastructure.

➤ Caching Servers

Caching servers, often known as caching-only servers, serve only cached requests for DNS replies. They do not retain zone files, like other secondary servers do; instead, they make queries, cache the replies, and provide the results to the requesting client, rather than storing zone files. Because caching servers do not maintain zone files or transfer zones as needed, they create a lot of network traffic as compared to secondary servers that do

➤ Recursive DNS

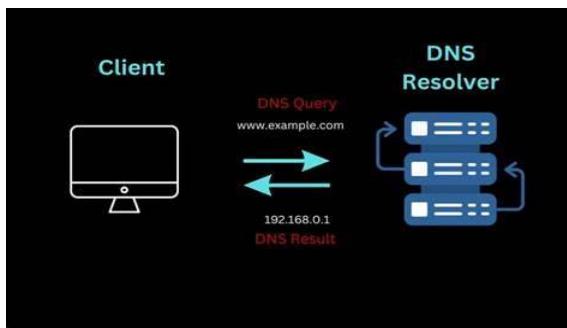
Handles queries for users by finding the appropriate IP address. Acts as an intermediary between the user and other DNS servers. Queries other servers to resolve domain names and delivers the result to the user. Your ISP's DNS resolver.

➤ **Forwarding DNS**

Forwards queries to specific external DNS servers for resolution. Redirects queries to another DNS server for resolution. Often used by organizations to streamline query management.

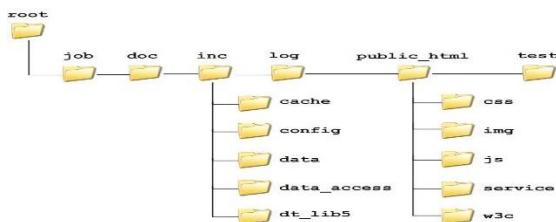
➤ **DNS Resolver**

A recursive resolver is a DNS server that takes queries from user applications to convert domain names (like www.example.com) into IP addresses. It performs the lookup by querying other DNS servers in a hierarchical manner, ensuring the user receives the correct IP address to access the desired resource.



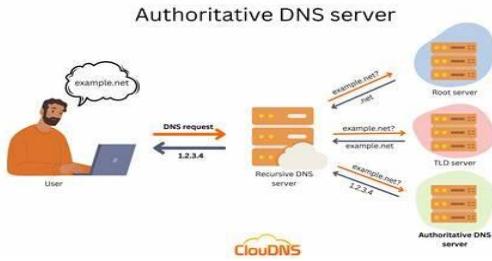
➤ **DNS Root Server**

Root servers operate in the root zone of DNS and store records to answer queries or cache results. There are 13 root servers, which handle initial requests to locate authoritative servers. Directs queries to appropriate TLD (Top-Level Domain) servers. Serves as the starting point for DNS lookups. Root servers managing (Dot) entries.



➤ **Authoritative Name Server**

This server holds up-to-date information about domain names and provides definitive answers to queries about those domains. Stores and provides answers for specific domain names. Holds the records (e.g., A, MX, CNAME) for a particular domain and answers queries directly. The DNS server for `velvetvogue.com`



For Velvet Vogue

- Recursive DNS ensures quick query resolutions for users accessing the website.
- An authoritative DNS stores records like A (IP address) and CNAME (aliases) for `www.velvetvogue.com`.
- Caching DNS improves user experience by reducing query time for repeat visitors.

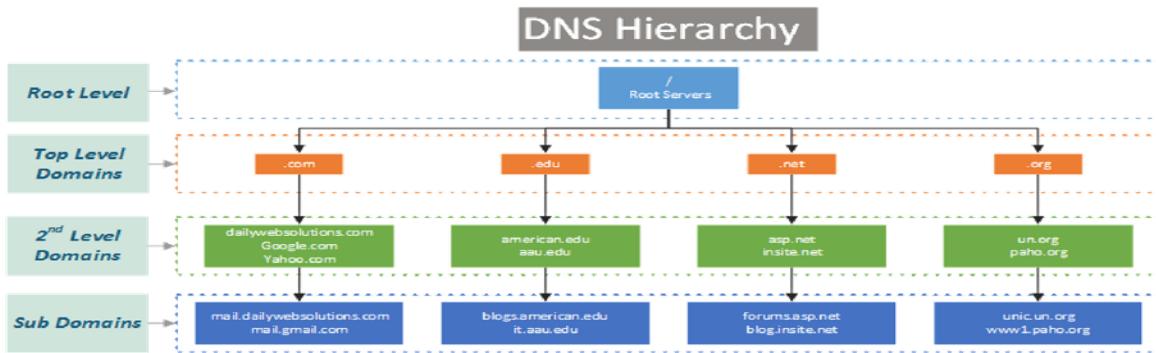
This ensures smooth accessibility and reliability of Velvet Vogue's online store globally.

Explanation of How Domain Names Are Organized and Managed

Organization of Domain Names

Domain names are organized hierarchically within the **Domain Name System (DNS)**, following a tree-like structure with different levels:

DNS Hierarchy



1. Zone servers / root level servers

It is the top level of the DNS hierarchy. The root servers are the initial stage in resolving the domain name. It gives a list of top-level domain (TLD) authoritative name servers. All top level domains (TLDs) are listed on these servers. There are 13 root DNS servers worldwide. These root servers are referred to as A-root servers, B-root servers, and so on, all the way up to K-root servers. Root servers store information about each TLD, such as where it is located and what its IP address is.

2. TLDs (Top Level Domains)

TLD is the second highest level domain in the Internet's hierarchical Domain Name System. The top level domains are .com, .org, .edu, and so on. If you look at the DNS hierarchy diagram, it will make more sense.

Examples

- **Generic TLDs**- .com, .org, .net, .edu
- **Country Code TLDs**- .uk, .au, .in
- **Sponsored TLD**- Managed by specific organizations, e.g., .gov for government use.

3. Domain at the Second Level

Individual or organization with a variable-length name. Depending on the type of organization or geographic area, these names are always based on a suitable top-level domain.

Example

In `velvetvogue.com`, `velvetvogue` is the second-level domain.

4. Subdomain

Additional names that are derived from the registered second-level domain name can be created by an organization. These are names that have been added to an organization's DNS tree in order to partition it into departments or geographic areas.

Examples of subdomains might include

- en.example.org for English content
- es.example.org for Spanish content
- it.example.org for Italian content
- de.example.org for German content

In practice, subdomains like "www" are less common today as they are often not necessary for visiting a site.

However, subdomains remain useful for organizing content and services.

Technically, all domains below the top-level domain are subdomains of the preceding one, but they are not always referred to in this way. The structure is hierarchical

- The domain name (SLD) is a subdomain of the TLD.
- A third-level domain is a subdomain of the SLD.
- Additional levels continue in a similar pattern.

While theoretically, the number of domain levels can be unlimited, practical use typically limits domain names to three levels for ease of memory and usability... (Anon., 2021)

Uses of DNS

- Domain Name Resolution: DNS servers convert user-friendly domain names (e.g., www.example.com) into IP addresses (e.g., 192.168.1.1) that computers use to locate and interact with one another on the internet.

- Website Hosting: DNS enables websites to utilize human-readable domain names rather than numeric IP addresses. When a user types a domain name into a web browser, DNS is responsible for routing that request to the IP address of the relevant web server.
- Translates human-readable domain names (e.g., velvetvogue.com) into IP addresses required by servers to locate and deliver content.
- Email Delivery: DNS is used to locate the mail servers connected with a domain, allowing email messages to be sent correctly. Mail servers utilize DNS to search for Mail Exchange (MX) records for a domain.
- Load Balancing: DNS may be used to distribute incoming network traffic among several servers for load balancing. This improves performance and availability by optimizing resource utilization.
- Content Delivery Networks (CDNs): CDNs employ DNS to route user requests to servers that are geographically nearby, lowering latency and enhancing content delivery speed and performance.
- Configuration of a Dynamic Host: DNS is frequently used in combination with DHCP to dynamically allocate IP addresses to network devices. DNS records may be updated by DHCP servers with freshly issued IP addresses.
- Virtual Private Networks (VPNs): DNS is used in VPN deployments to convert domain names into IP addresses, allowing users to safely access resources on a private network.
- Internet of Things (IoT): DNS may be used to correlate domain names with IoT devices, making it easier to manage and identify these devices on a network.

Management of Domain Names

Domain names are managed by a collaborative system involving international organizations, registrars, and domain owners:

1. **ICANN (Internet Corporation for Assigned Names and Numbers)**
 - A global organization responsible for overseeing the DNS system, including the allocation of IP addresses and domain name management.
2. **Domain Registrars**
 - Companies authorized by ICANN to sell domain names to the public.

- Examples: Go Daddy, Name cheap, Google Domains.
- Services include domain registration, DNS configuration, and renewal management.

3. Domain Management Activities

- **Registration:** - Domain owners purchase and register a domain through a registrar.
- **Renewal:** - Domains must be renewed periodically (annually or longer).
- **Transfer:** - Domains can be transferred between registrars.
- **DNS Configuration:** - Adjust DNS settings to point the domain to specific servers for website hosting, email, etc.

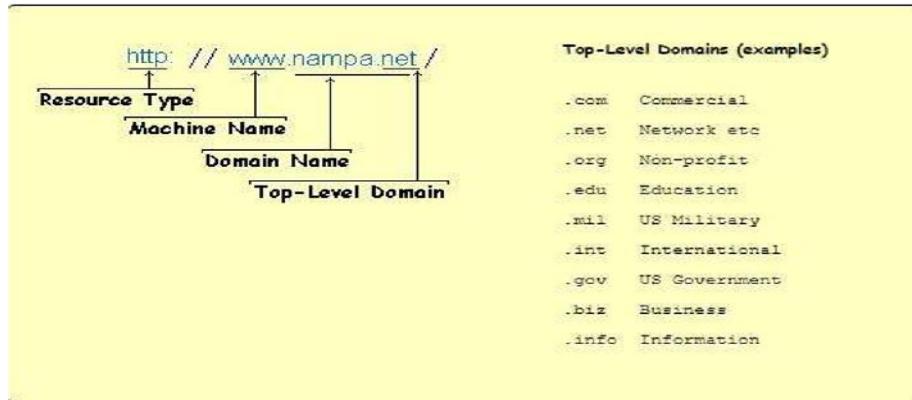
DNS Records

- **A Record (Address Mapping Record):** Maps a domain name to an IPv4 address, helping to resolve a domain to its corresponding IP address.
- **AAAA Record (IPv6 Address Record):** Maps a domain name to an IPv6 address, accommodating the newer IP version.
- **CNAME Record (Canonical Name Record):** Maps one domain name to another, allowing multiple domain names to point to the same IP address.
- **MX Record (Mail Exchanger Record):** Specifies the mail server responsible for handling emails for a domain.
- **NS Record (Name Server Record):** Indicates the authoritative name servers for the domain, which handle DNS queries for that domain.
- **PTR Record (Reverse-lookup Pointer Record):** Provides a domain name associated with an IP address, used for reverse DNS lookups.
- **CERT Record (Certificate Record):** Stores cryptographic certificates for security purposes, used to verify domain ownership and secure communications. (educba.com, 2024)

By leveraging this hierarchical organization and management system, Velvet Vogue ensures that its website is accessible, secure, and reliable for users worldwide.

Structure of Domain Names

The following diagram shows the structure of a domain name. The top-level domain includes .com, .net or any of the standard country top-level domains like .us (United States) or .ca (Canada). New top-level domains (TLD) are being created all the time.



1. Resource type

The type of page. Typical web pages are HTTP, an acronym for 'Hyper Text Transfer Protocol'. HTTPS is secure HTTP. Other resources include, FTP, GOPHER, and TELNET etc.

2. Machine Name

This can also be called a sub-domain. Most web pages start with WWW. This isn't an absolute rule. We could call it MYWEBSITE instead. Historically, people have chosen WWW (World Wide Web). You don't need to use this part on most web pages if the DNS is set up properly.

3. Top-Level Domain

The top level domain is a huge group of similar types of domain names. Some are restricted to only certain people, like .gov or .int.

4. Domain Name

The Domain Name, which includes the top-level domain name is the complete name, owned by someone (like you)!

Explanation of the Purpose and Relationships between Communication Protocols, Server Hardware, Operating Systems, and Web Server Software

Introduction to communication protocols

Web development



Web development involves creating websites for hosting on the internet or intranet. This multifaceted process includes web designing, content creation, scripting, network security, and more. Web development can range from simple text pages to complex websites.

The most common languages used by web developers—professionals who create and maintain websites—are HTML (Hypertext Mark-up Language), CSS (Cascading Style Sheets), and JavaScript. Web developers are typically categorized into two groups:

1. **Front-End Developers:** Focus on the visual aspects of the website, ensuring it looks good and is user-friendly. They use HTML, CSS, and JavaScript to create the layout, design, and interactivity of a site.
2. **Back-End Developers:** Handle the server-side aspects of web development. They manage databases, server logic, and application integration to ensure the site functions correctly and efficiently.

The combination of these processes and the collaboration of front-end and back-end developers result in a successful web development project. (Sriharan, 2023)

Creating the perfect website that meets the client's approval is a complex task that can take months. Following a specific hierarchy can provide a clear path for the developer, guiding

the project from start to finish. This structured approach, known as the web development process, includes a series of steps that help ensure the project's success.

While these steps may vary among developers, the following default process is widely effective:

1. Information Gathering:

- Researching and collecting information is the most crucial initial step. Without it, the developer would be working blindly. This stage involves understanding the client's needs and objectives, which helps the developer choose the right approach. Effective communication between the client and developer is essential for making informed decisions.
- Techniques such as interviews, questionnaires, and thorough research are employed to gather necessary information. Each website is unique, so the developer must be confident in the resources collected to proceed effectively.

2. Planning:

- Once the necessary information is gathered, the planning phase begins. The developer creates a project plan, including timelines and project variables. Planning is as vital as any other stage in the process.
- An essential component of planning is the creation of a sitemap, which visually represents the entire website in a single chart, serving as a blueprint for the development phase.

3. Design:

- The user interface (UI) is critical as it forms the first impression for visitors. A successful first impression keeps users engaged with the site. The design must align with the site's purpose; for instance, entertainment sites differ significantly from financial sites.
- The site should have a cohesive theme and well-chosen color schemes to enhance user engagement and overall aesthetic appeal.

4. Development:

- This is the phase where the actual website is built, divided into front-end and back-end development.
- The front end involves HTML, CSS, and various frameworks and libraries, while the back end includes server-side applications such as databases.

5. Testing and Delivery:

- After development, the website undergoes rigorous testing, including usability tests, unit tests, and compatibility tests, among others. This phase ensures that the website is free of errors and bugs before going live.

6. Maintenance:

- Equally important as the other stages, maintenance involves making the website live and accessible to users. It includes exchanging user guides, source codes, and project documents with the client, along with providing post-deployment support and maintenance.

Adhering to these steps can lead to an effective and successful project. This comprehensive and universal hierarchy ensures that the developer can trust the process and work systematically.

Web Technology: Web technology encompasses various tools and techniques used to communicate with devices across the internet. The following topics will be explored in depth for our project:

- **Communication Protocols**
- **Server Hardware**
- **Operating System**
- **Web Server Software**

By following these guidelines and understanding the relevant web technologies, the web development process can be executed efficiently, leading to a high-quality website that meets client expectations.

Communication protocols

Communication protocols are standardized rules that enable devices to communicate over a network. They ensure reliable data transmission by defining how data is formatted, transmitted, and received. Protocols operate at various layers of the OSI model, each addressing specific functions, such as routing, error handling, and data delivery. Key protocols include HTTP/HTTPS for web communication, SMTP/IMAP for email, TCP/IP for internet connectivity, FTP for file transfers, and DNS for domain resolution. Protocols ensure interoperability, security, and scalability across diverse systems and devices. They are essential for enabling seamless communication in applications like web browsing, file sharing, and online transactions.

Introduction about IEEE

The Institute of Electrical and Electronics Engineers (IEEE), a professional organization, advances technical innovation and excellence for the good of humanity. The IEEE, which was established in 1963, is the largest technical professional association in the world, with more than 400,000 members spread over more than 160 nations. Its goal is to assist engineering research and development across a range of disciplines and to foster technological excellence and advancement by fostering the creation and use of standards for technology. It also offers its members resources and opportunities for professional growth.

The IEEE standards in computer networks and their purpose are listed below.

IEEE 802 -LAN/MAN Overview and architecture

IEEE 802.1 -LAN/MAN Bridging and management

IEEE 802.1s -Multiple spanning tree

IEEE 802.1 w -Rapid reconfiguration of spanning tree

IEEE 802.1x -Port-based network access control

IEEE 802.2 -Logical Link Control (LLC)

IEEE 802.3- CSMA/CD access method (Ethernet)

IEEE 802.3ae -10 Gigabit Ethernet

IEEE 802.4 -Token passing bus access method and Physical layer specifications

IEEE 802.5 -Token Ring access method and Physical layer specifications

IEEE 802.6 -Distributed Queue Dual Bus (DQDB) access method and Physical layer specifications (MAN)

IEEE 802.7 -Broadband LAN

IEEE 802.8 -Fiber Optic

IEEE 802.9 -Isochronous LANs (standard withdrawn)

IEEE 802.10- Interoperable LAN/MAN Security

IEEE 802.11 -Wireless LAN MAC and Physical layer specifications

IEEE 802.12- Demand-priority access method, physical layer and repeater specifications

IEEE 802.13- Not used

IEEE 802.14 -Cable modems (proposed standard was withdrawn)

IEEE 802.15 -Wireless Personal Area Network (WPAN)

IEEE 802.16 -Wireless Metropolitan Area Network (Wireless MAN)

IEEE 802.17 -Resilient Packet Ring (RPR) Access

IEEE 802

IEEE 802 is an Institute of Electrical and Electronics Engineers (IEEE) standard set that includes

The physical and data link levels of the Open Systems Interconnection (OSI) paradigm. It defines

Standards and protocols for wired local area networks (WLAN), metropolitan area networks

(MAN), and wireless networks; defines characteristics, operating procedures, protocols, and

Services for networks that carry variable sized packets; and specifies the development and
M.M.M AASHIK/E230667

Handling of compatible devices and equipment.

IEEE 802.3

IEEE 802.3 is a working group of standard standards maintained by the Institute of Electrical and

Electronics Engineers (IEEE) for Ethernet, a form of packet-based physical communication in a

Local area network. It specifies a physical layer and a data link layer for a wired, fast Ethernet

Network connection's media access control, or MAC address. These physical connections are

Formed by copper or fiber cables between nodes or equipment like as routers, switches, and hubs.

In general, IEEE 802.3 standards define the physical medium and operation of Ethernet.

However, there are other variants of this standard in use today.

IEEE 802.6

DQDB (Distributed Queue Dual Bus), an IEEE 802.6 standard, is a MAN (Metropolitan Area

Network) protocol. It is a high-speed shared medium access control protocol that operates on a

Bus network. It features two unidirectional buses for control reasons, and the bus may transfer

Data, video, and audio through a network with bandwidth assigned according to time periods.

The paired bus has the advantage of being utilized to address failure configuration. It has a range

of up to 30 miles at 34-55 Mbps

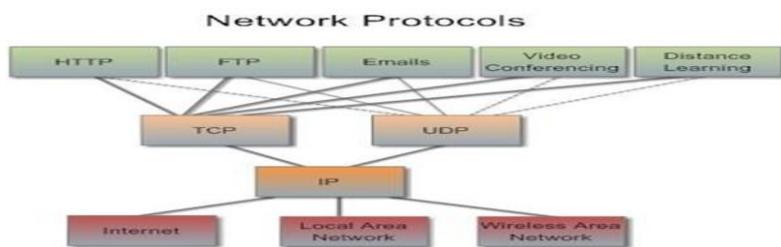
IEEE 802.11

IEEE standard 802.11 describes Wireless Local Area Network (WLAN) or WiFi. It includes all

WLAN series items. It is designed for a range of about 100 meters. It does not offer the service

Throughout the coverage region, preventing continuous connectivity. In terms of usability, this

802.11 standard offers less scalability



HTTP (Hyper Text Transfer Protocols)

HTTP (Hyper Text Transfer Protocol) is a fundamental protocol used to transfer data over the World Wide Web, enabling communication between a client (typically a web browser) and a server. It operates at the application layer of the internet protocol suite and follows a request-response model. When a user accesses a web page, their browser (the client) sends an HTTP request to the server hosting the website. The server processes this request and responds with the requested data, such as HTML documents, images, or multimedia files, which the browser renders into a user-friendly web page.

HTTP supports several request methods, such as GET and POST, which define the nature of the interaction between the client and server:

1. GET: This method retrieves data from the server. For example, when a user visits the homepage of the Velvet Vogue e-commerce platform, the browser sends a GET request to fetch the page's content. GET requests are used to retrieve publicly available data and do not include sensitive information in the request body.
2. POST: This method submits data to the server, often used for actions like creating a new user account or processing a payment. For instance, when a customer submits login

credentials or payment details on Velvet Vogue, a POST request securely transmits this sensitive information to the server.

To enhance security, HTTP works alongside TLS/SSL certificates in its secure variant, HTTPS. This encrypts communications between the client and server, protecting sensitive data like login credentials and payment details from interception or tampering. For e-commerce platforms like Velvet Vogue, HTTPS ensures that customers can confidently browse, shop, and complete transactions without compromising their personal information.



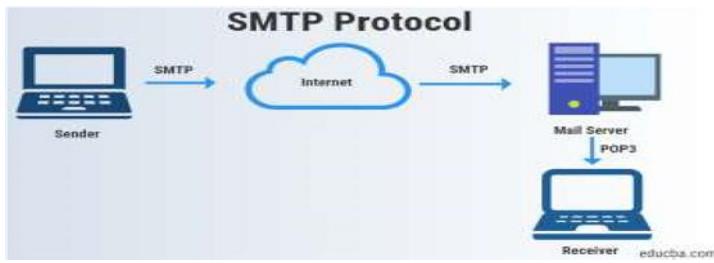
FTP (File Transfer Protocol)

FTP is a client/server protocol used for transferring files between computers over a TCP/IP network. It supports authenticated access using usernames and passwords, as well as anonymous access. FTP is commonly used for downloading files from servers and transferring web pages to servers.



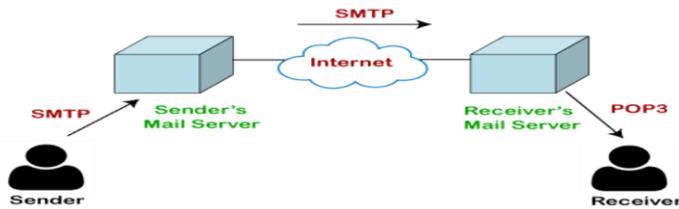
Simple Mail Transfer Protocol (SMTP)

The Simple Mail Transfer Protocol (SMTP) is a technological standard for sending email across a network. SMTP, like other networking protocols, allows PCs and servers to communicate data independently of the hardware or software on which they are running. Just as a defined method of addressing an envelope allows the postal service to function, SMTP standardizes the way email goes from sender to receiver, allowing for broad email delivery. Sends a confirmation email to Velvet Vogue's customers after a purchase.



POP protocol

Post Office Protocol is abbreviated as POP. As we all know, SMTP is a message transport agent. When a message is sent, SMTP is used to transfer it from the client to the server and then to the destination server. However, the message is transferred from the receiving server to the real server via the Message Access Agent.



TCP/IP (Transmission control protocol / internet protocol)

End-to-end communication via the internet and other computer networks is made possible by the TCP/IP (Transmission Control Protocol/Internet Protocol) family of network protocols. It consists of two main protocols: Transmission Control Protocol (TCP), which provides stable, ordered, and error-checked data flow between applications, and Internet Protocol (IP), which handles the routing and addressing responsibilities necessary for network device connections. Two other protocols that are a part of TCP/IP are the Internet Control Message Protocol (ICMP) and the User Datagram Protocol (UDP). It is the foundation of the modern internet and the main method of communication for most networked devices.

1. OSI Model (open system interconnection)

A theoretical framework that explains the communication within a computer network is called the OSI (Open Systems Interconnection) model. Each of the seven tiers specifies a certain network communication function. They are:

- i. **Physical Layer:** The physical layer defines the network's physical components, including its connections, cables, and communication methods.
- ii. **Data Link Layer:** The data link layer allows data to be transmitted between two devices that are linked to the same physical network. It also handles the identification and correction of errors.
- iii. **Network Layer:** At the network layer, data routing between various networks is carried out. It also provides logical addressing, such as IP addresses.
- iv. **Transport Layer:** The transport layer provides safe end-to-end communication between two devices. It ensures that data is delivered in the correct order and manages congestion.
- v. **Session Layer:** The session layer is in charge of starting, maintaining, and ending communication sessions between two devices.
- vi. **Presentation Layer:** The presentation layer determines how data is delivered between programs and how it is organized. It oversees tasks including data compression and encryption.
- vii. **Application Layer:** The way user apps communicate with the network is defined by the application layer. It is made up of protocols such as HTTP, FTP, and SMTP.

IP (Internet Protocol: As an addressing protocol, IP is purpose-built. Mostly TCP is used with it. As messages travel through a network of nodes to their final destination, the IP addresses in the packets help in their routing. Network connections are made using the most widely used protocol, TCP/IP.

There are four layers in all, and each one serves a particular purpose in communication.

- i. **Network Interface Layer:** A device is connected to a physical network via the network interface layer, which also provides services for data transmission across the network. It is made up of Wi-Fi and Ethernet protocols.
- ii. **Internet Layer:** The Internet Layer is used to route data packets logically between many networks. Internet Protocol (IP), a component of the Internet, is used to route data.
- iii. **Transport Layer:** The transport layer provides safe end-to-end communication between two devices. It is composed of TCP and UDP protocols.

- iv. **Application Layer:** The way user apps communicate with the network is defined by the application layer. It has several protocols that enable data exchange and network resource access, such as SMTP, HTTP, and FTP.

DHCP (Dynamic Host Configuration Protocol)

Network administrators can automate IP address distribution in a network by using the DHCP communication protocol. Each device utilizing an IP network must have a distinct IP in order to connect to the internet. DHCP enables network administrators to centrally distribute IP addresses and automatically deliver a new IP address whenever a device connects from a different location on the network. DHCP utilizes a client-server architecture

Difference between SMTP & POP

Feature	SMTP (Simple Mail Transfer Protocol)	POP (Post Office Protocol)
Role	Sending emails from the client to the server.	Retrieving emails from the server to the client.
Functionality	Sending emails from the client to the server.	Incoming mail server.
Connection Type	Sending emails from the client to the server.	Connection-oriented (temporary connection).
Usage	Used for sending emails.	Used for retrieving and downloading emails.

Difference between HTTP & IP

Features	HTTP (Hypertext Transfer Protocol)	IP (Internet Protocol)
Layer in OSI Model	Application layer (Layer 7).	Network layer (Layer 3).

Functionality	Transmission of hypertext content.	Addressing and routing of data packets.
Connection Type	Stateless protocol.	Connectionless protocol.
Examples	Web browsing, fetching web pages.	Routing data packets across the internet.

Difference between FTP & SMTP

Feature	FTP (File Transfer Protocol)	SMTP (Simple Mail Transfer Protocol)
Purpose	Transferring files between systems.	Sending and receiving emails.
Connection Type	Connection-oriented.	Connection-oriented.
Authentication	Supports various authentication methods.	Primarily relies on username and password.
Example	Uploading and downloading files.	Sending emails between mail servers.

Introduction to server

Servers

A server is a computer system or gadget that joins a network and gives other computers or gadgets access to resources, services, or data. It is designed to manage several client demands while delivering services in a trustworthy and effective way. Large-scale data centers that provide cloud computing, storage, and virtualization can be found among servers, as can smaller systems that provide standard functions like file sharing and printing.

Usually, servers are built to run constantly and manage several client requests at once. They can handle more complicated and demanding applications because they have more memory, computing power, and storage than typical client PCs. They are usually kept in safe, climate-controlled spaces with backup power supply to guarantee their availability.



Servers are the digital workhorses that power much of our modern world. To understand their significance, it's essential to grasp the wide array of tasks they can perform:

1. Data Storage and Retrieval:

- Servers provide a centralized location for storing and managing data, including files, documents, databases, and media content. This data is accessible to authorized users from various locations.

2. Website Hosting:

- Web servers store website files and deliver them to web browsers when users request a page. They handle text, images, videos, and interactive features.

3. File Sharing:

- File servers store and share files within a network, ensuring accessibility to authorized users and incorporating security features to protect sensitive data.

4. Database Management:

- Database servers manage and organize structured data, ensuring data consistency, reliability, and efficient querying for applications that rely on databases.

5. Communication Services:

Servers enable various forms of communication, such as instant messaging, voice-over-IP (VoIP), and videoconferencing, facilitating real-time interactions and data exchange between users.

6. Backup and Recovery:

- Backup servers automate the process of creating copies of data for disaster recovery and data loss prevention, storing data redundantly and enabling rapid restoration in case of system failures or data corruption.

7. Security Services:

- Security servers, including firewalls and intrusion detection systems (IDS), safeguard networks and data from cyber threats by monitoring network traffic, filtering malicious activity, and enforcing security policies.

These diverse functions highlight the crucial role servers play in modern computing, providing the backbone for data storage, communication, and security across various networks and applications.

Server software

Server software is a critical component of server functionality and management. It encompasses a range of applications and systems that enable servers to perform specific tasks, communicate with clients, and efficiently manage hardware resources. The choice of server software depends on the intended purpose and requirements of the server.

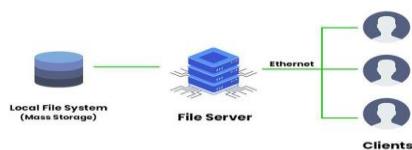
Types of servers

- File server
- Web server
- Email server
- Database server
- Application server
- Virtualization server
- Proxy Server
- DHCP Server

- DNS Server
- **Software server**

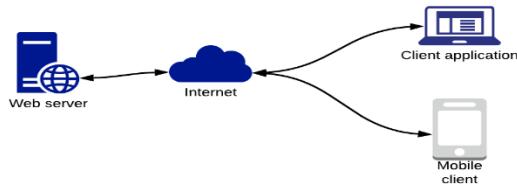
File server

A dedicated server used for managing and storing network-shared data files is called a file server. It can be used to store financial data by the Accounting and Finance Department and administrative files and documents by the Administration Department. File sharing protocols including SMB, NFS, and FTP can be supported by the file server because of its fast CPU and big storage capacity.



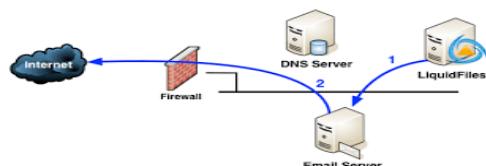
Web server

Host websites and respond to HTTP requests from web browsers, delivering web pages, images, videos, and other content. Examples: Apache, Nginx, Microsoft Internet Information Services (IIS).



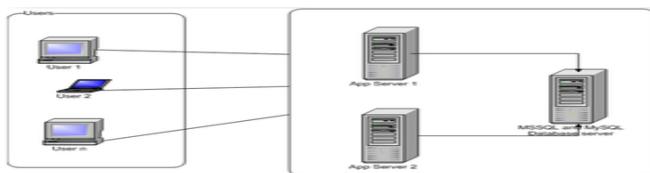
Email server

An email server is a specialized computer that controls email traffic and keeps email messages. It can be used by the Customer Service Department and HR Department to handle email correspondence with clients and staff. The email server should support email protocols like SMTP, POP3, and IMAP and have a fast processor and lots of storage space.



Database server

Large storage areas are served by database servers, which are used and accessed by businesses to run a number of different programs. Depending on the database structures, a database server can operate.



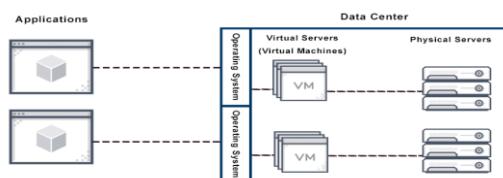
Application server

These servers use virtual server connections to link clients to software programs. Users can browse applications without having to download data on their hardware thanks to this. Application servers are perfect for organizations because they can efficiently host substantial volumes of application data to several users simultaneously.



Virtual machine (VM)

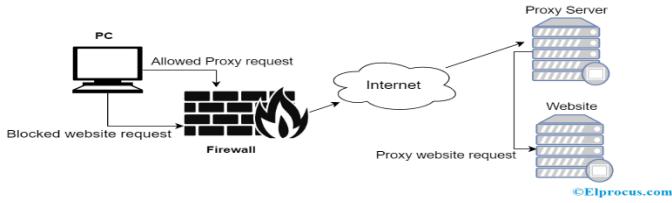
Virtual machines only store and connect data in virtual space. IT teams use software called a hypervisor, commonly referred to as a virtual machine monitor (VMM), to build virtual machines. A hypervisor can run hundreds of virtual machines on a single piece of physical hardware. As they are the cheapest sort of server to run, this server virtualization technique is frequently utilized for data transfer and storage.



Proxy Server

An intermediary between a client and the internet is a proxy server. A web page request made by a client is initially sent via the proxy server. The proxy server then assesses the request, sends it to the internet, and gives the client the response. Software used as proxy servers includes Squid, Nginx, and Apache HTTP Server instances. The exact proxy

server software and specs needed by Alliance Health will depend on their business requirements, security requirements, and content filtering needs.



- **Hardware server**

Key Components of Server Hardware:

1. Processor (CPU):

- The main component determining the server's performance. Choices range from entry-level CPUs for small businesses to mid-level and high-end CPUs for more demanding applications.

2. Memory (RAM):

- RAM allows the server to read and write data quickly, reducing the need to access slower storage devices. More RAM improves performance by enabling the CPU to handle larger workloads efficiently.

3. Storage Devices:

- Servers use HDDs or SSDs for data storage. RAID configurations are common for redundancy and improved data integrity.

4. Motherboard:

- The main board that connects all other hardware components. Server motherboards support multiple processors, higher RAM capacities, and additional expansion slots.

5. Network Interface Cards (NICs):

- Provide network connectivity, often with multiple NICs for redundancy and load balancing. Commonly use Gigabit or 10 Gigabit Ethernet interfaces.

6. Power Supply Unit (PSU):

- Reliable power supplies with redundant configurations ensure continuous operation, reducing the risk of downtime due to power failures.

7. Cooling Systems:

- Advanced cooling solutions, including fans and heat sinks, maintain optimal operating temperatures.

Some servers use liquid cooling for more efficient heat dissipation.

8. Form Factor:

- Server hardware comes in various forms, such as rack-mounted, tower, or blade servers. Rack-mounted servers are common in data centers for efficient space utilization.

9. Remote Management:

- Allows administrators to monitor, configure, and troubleshoot servers remotely using technologies like Intelligent Platform Management Interface (IPMI).

10. Redundancy and Reliability:

- Features like redundant power supplies, fans, and storage enhance server reliability, ensuring continuous operation and high availability.

In summary, both server software and hardware are critical in ensuring the efficient and reliable performance of servers, which are essential for data storage, communication, security, and numerous other functions in modern computing environments.

The most well-known producers of hardware servers are Lenovo, Dell, HP, and IBM. Hardware server selection will be based on a number of criteria, including their budget and unique requirements.

There are 3 main types of Hardware Servers:

- Tower Server
- Rack Server
- Blade Server

Types of hardware servers

Tower servers



Tower servers are a type of server hardware designed in a vertical, standalone form factor resembling a desktop computer. They are commonly used in small to medium-sized businesses, remote offices, or individual workstations due to their flexible placement options, as they can easily fit on or under desks in office environments. Although less dense than rack or blade servers, tower servers offer good scalability to accommodate growing business needs. They are designed for ease of hardware maintenance and upgrades, with components easily accessible for repairs or enhancements. Tower servers also have effective cooling mechanisms due to their standalone configuration, ensuring reliable operation. While they offer the advantage of flexible placement, scalability, cost-effectiveness, and quiet operation, they do consume more physical space compared to the compact design of rack or blade servers. Additionally, while they can be upgraded, tower servers may not scale as efficiently for very large deployments, and their limited density makes them less ideal for organizations with significant server requirements. Overall, tower servers provide a versatile and accessible solution for businesses with moderate server needs, balancing performance, scalability, and ease of maintenance.

Rack servers



Rack servers are specifically designed to be mounted in standard server racks or cabinets, making them ideal for data centers and enterprise environments where space efficiency and scalability are critical. These servers are prized for their high density, allowing organizations to maximize rack space and streamline their data center footprint. By vertically stacking servers within racks, data centers can optimize floor space, leading to a more organized and efficient layout. Rack servers also facilitate centralized management, making it easier to monitor and maintain a large number of servers from a single location. They often come with integrated cable management systems, which enhance overall tidiness and simplify maintenance tasks. This design approach helps data centers achieve a compact, scalable infrastructure that supports growth while maintaining a clean and manageable environment.

Blade servers



Blade servers are a specialized type of server hardware designed to fit into a chassis, also known as a blade enclosure or blade chassis. This design offers a highly dense and modular approach to server deployment. Blade servers are compact, allowing a significant number of servers to be housed in a single enclosure, thereby optimizing data center space. Their modular nature makes them particularly well-suited for large-scale virtualization and cloud computing deployments. One of their key features is hot-swappability, which allows individual blades to be replaced or upgraded without disrupting other components in the enclosure. Additionally, blade servers often share power, cooling, and networking infrastructure within the chassis, reducing overall cabling requirements and contributing to a more streamlined data center environment.

Feature	Tower Server	Rack Server	Blade Server
Form Factor	Vertical Housing resembling a desktop tower.	Designed for mounting in a server rack.	Compact units that fit into a chassis.
Scalability	Limited scalability for smaller setups.	More scalable, easy to add or replace.	Highly scalable, modular architecture.
Space Efficiency	Takes up more floor space.	Optimized for space efficiency.	High-density computing, reduced footprint.
Use Cases	Small businesses, limited scalability.	Data centers, higher scalability.	Large-scale data centers, resource consolidation.
Placement	Standalone, suitable for desk or corner.	Mounted in a server rack.	Modular design, fits into a housing.

Server software VS Server Hardware

Feature	Server Hardware	Server Software
Definition	Physical components that make up the server.	Software programs that enable server functionality.
Components	CPU, RAM, Storage (HDD/SSD), Network interfaces, Power Supply,	Operating System (OS), Server applications,

	Cooling Systems, Motherboard, etc.	Middleware, Drivers, Utilities, etc.
Nature	Tangible and physical.	Intangible and programmatic.
Function	Provides the infrastructure for computing power, storage, and networking.	Manages and controls hardware resources, facilitates specific server functions.
Role	Hardware provides the platform for software to run.	Software directs and coordinates hardware operations to fulfill specific tasks.
Customization	Customizable through hardware upgrades and additions.	Customizable through software configuration, installation of applications, and customization of settings.
Examples	Dell PowerEdge, HP ProLiant, IBM System x servers	Windows Server, Linux distributions, Apache HTTP Server, Microsoft SQL Server
Cost Considerations	Costs involve purchasing and maintaining physical components.	Costs involve licensing, support, and development (if applicable).
Upgrades and Maintenance	Upgrades involve physical components (e.g., adding RAM, replacing hard drives).	Upgrades involve updating software, applying patches, and configuring settings.
Management	Hardware management involves monitoring physical components.	Software management involves configuring settings, monitoring processes, and handling user requests.

--	--	--

Justification how they are related for Velvet Vogue's Website

For the scenario of developing a website for Velvet Vogue, an e-commerce platform focused on showcasing trendy casualwear and formalwear, shared hosting is a suitable server hardware option. Below are the justifications based on designing, publishing, and assessing the website

A. Designing

1. Cost-Effectiveness

Shared hosting is significantly more affordable than dedicated hosting or virtual private servers. This cost advantage is critical for Velvet Vogue, especially during its initial stages as a fashion e-commerce website. The cost savings from shared hosting allow Velvet Vogue to allocate more of its budget toward product inventory, marketing efforts, and customer acquisition, instead of expensive server infrastructure. This makes shared hosting a practical choice for maintaining a professional online presence without straining financial resources.

2. Ease of Use

Shared hosting environments are designed to be beginner-friendly and accessible, even for those with limited technical expertise. This ease of use is particularly beneficial for Velvet Vogue, as it may not have a dedicated IT team to manage complex server configurations. Shared hosting providers typically offer intuitive control panels (e.g., cPanel) and straightforward management tools, enabling Velvet Vogue to efficiently update product pages, promotions, and customer inquiry forms with minimal technical effort.

3. Limited Configuration Needs

The shared hosting environment is pre-configured by the hosting provider, simplifying server management. Velvet Vogue, primarily focused on displaying product images, offering a shopping cart, and processing inquiries, does not require complex server setups. Shared hosting provides all the necessary features and resources to support the website's core functionality, such as uploading new collections and managing basic e-commerce operations, without the burden of managing intricate server settings.

B. Publishing

1. Quick Deployment

Shared hosting providers streamline the website setup process, enabling quick deployment. This allows Velvet Vogue to launch its website and showcase its collections to potential customers without unnecessary delays. The ability to go live quickly ensures that Velvet Vogue can establish its online presence promptly, attracting visitors and starting sales operations as soon as possible.

2. Managed Services

One of the key advantages of shared hosting is the inclusion of managed services. Hosting providers handle routine server maintenance, security updates, and backups. This alleviates the burden on Velvet Vogue, allowing the team to focus on running the fashion business and curating collections. Managed services ensure that critical tasks such as monitoring server performance, ensuring security, and performing regular backups are handled professionally, safeguarding the website's data and functionality.

C. Assessing

1. Traffic Handling

Shared hosting is well-suited for small to medium-sized websites with moderate traffic levels. Given that Velvet Vogue's website is primarily an e-commerce platform for showcasing collections and enabling purchases rather than a high-traffic enterprise platform, shared hosting provides adequate resources to handle the anticipated traffic. It ensures smooth performance and reliable access for users browsing Velvet Vogue's product catalog and completing transactions.

2. Scalability

While shared hosting is ideal for Velvet Vogue's current needs, it may have limitations in handling very high traffic volumes or extensive growth. However, many shared hosting providers offer scalable solutions, allowing Velvet Vogue to upgrade to more advanced hosting options such as virtual private servers or dedicated hosting if its traffic and sales demands increase. This flexibility ensures that the hosting solution can evolve alongside Velvet Vogue's business growth.

3. Support

Shared hosting services often include reliable customer support, which is invaluable for Velvet Vogue in case of technical issues or website management queries. Support teams can assist with troubleshooting problems, optimizing website performance, or answering questions about hosting configurations. This ensures that Velvet Vogue can maintain a functional and secure online presence without extensive in-house technical expertise.

Summary

Shared hosting is a highly suitable server hardware option for Velvet Vogue's website due to its cost-effectiveness, ease of use, quick deployment capabilities, and managed services. It aligns well with the requirements of a fashion e-commerce platform, providing adequate resources for displaying collections, handling customer inquiries, and processing transactions. Shared hosting offers a practical, affordable, and scalable solution for Velvet Vogue to establish and grow its online presence.

Advantages of tower server

Installation and maintenance simplicity

Tower servers are simple to set up and maintain since they don't require any specific racks or cabinets. As a result, they are appropriate for tiny firms or offices with limited space and infrastructure.

Scalability:

Tower servers are frequently more scalable than desktop PCs. They can house extra hardware components like hard drives, RAM, and expansion cards, allowing for future upgrades as the server's needs develop.

Small Businesses Can Save Money:

Tower servers are a low-cost option for small firms or organizations with minimal IT resources. They deliver server-grade performance without requiring a dedicated server room.

Silent operation:

Tower servers are often quieter to operate than rack servers. This can be useful in places where noise levels must be kept low, such as tiny offices.

Advantages of Rack server

-  Efficiency of Space:

Rack servers save space and enable high-density server configurations. Multiple servers can be stacked vertically in a single rack, maximizing data center space use.

-  Scalability:

Rack servers, like tower servers, are scalable and can be easily expanded by adding more servers to the rack. This scalability is critical for firms with expanding IT requirements.

-  Cable management centralized:

Rack servers help to keep cable management cleaner and more orderly. Cables may be neatly stored within the rack, making management and troubleshooting easy.

Advantages of Blade server

-  Extreme Density:

Blade servers are distinguished by their great density. Multiple blade servers can be installed in a single chassis, providing a large boost in processing capability while maintaining a compact physical footprint.

-  Management Has Been Simplified:

Blade servers frequently include integrated management tools that make administration jobs easier. This can feature centralized resource management, simple deployment of new blades, and expedited firmware upgrades.

-  Power Consumption is Reduced:

Blade servers are energy-efficient in design. When compared to traditional servers, the shared infrastructure within the chassis, including power supplies and cooling, can result in reduced total power usage.

Operating System

An operating system (OS) is essential software that enables a user to run various applications on a computing device by managing the hardware and software resources of the system. The OS performs a crucial role in overseeing and coordinating the use of hardware components such as input devices (e.g., keyboards and mice), output devices (e.g., monitors, printers, and

scanners), network devices (e.g., modems, routers, and network connections), and storage devices (e.g., internal and external drives).

The OS facilitates the efficient execution and management of software applications by allocating and managing resources such as memory, CPU, and storage. For instance, when multiple applications are running simultaneously. Such as a web browser, firewall, and antivirus. The OS ensures that each program receives the necessary resources to operate effectively.

By providing a consistent and standardized framework, the OS simplifies application development, allowing software developers to focus on their programs without needing to address specific hardware details. This approach leverages common libraries and interfaces, promoting compatibility and efficiency across different hardware configurations



Server Operating System

In a client-server architecture, a server operating system (server OS) operates on a server and delivers Different services to client workstations over the network. It serves as the software backbone for a variety of Programs and applications. A server OS provides enhanced features for running, managing, monitoring, and Controlling programs, processes, and client devices such as web servers, file servers, application servers, mail servers, database servers, and other servers.

Traditional operating systems like Windows, Linux, and macOS are suited for personal and professional computing tasks, but server operating systems are engineered to handle the intensive requirements of managing network services and resources. These OSs are

optimized for performance, security, and reliability, ensuring that servers can effectively support high-demand environments.

Server operating systems are equipped with features that support various server roles, such as file servers, web servers, mail servers, application servers, and database servers. They facilitate efficient resource management by handling multiple simultaneous requests from client computers, managing network traffic, and ensuring data security and integrity.

Windows Server



Microsoft's Windows Server operating systems are built for server use. It includes functionality like Active Directory, DNS, DHCP, and others. It is ideal for a wide range of applications and environments since it supports several server roles. A graphical user interface (GUI) and a command-line interface (CLI) are available in Windows Server.

Windows Server is extremely successful in contexts where interaction with Microsoft technologies is critical. Its Active Directory services are especially useful for managing people and resources in a Windows centric network. The operating system offers administrators a familiar interface and supports a wide range of applications. However, its license prices can be an issue, and some users may choose more lightweight or open-source options for specific server jobs.

Linux (Different Distributions):



Ubuntu Server is a version of the popular Linux distribution Ubuntu designed primarily for server settings. It is built on the Debian architecture and is well-known for its stability, security, and usability. Ubuntu Server is a popular web server, database, and cloud computing platform. It includes the APT package management system, which allows for simple program installation and updates.

Ubuntu Server is extremely effective for a wide variety of server applications. Because of its open-source nature, it provides for greater flexibility and customization, making it a popular choice for web servers, databases, and cloud settings. The APT package management system makes it easier to install and update applications. Ubuntu Server is well-known for its reliability and security, and the lack of license fees makes it an affordable alternative for many businesses. Frequently used for hosting PHP-based websites, which Velvet Vogue might use for e-commerce functionality



Red Hat Enterprise Linux is an open-source Linux operating system distribution. It is intended for corporate deployments and provides stability, security, and support. RHEL manages software via the Red Hat Package Manager (RPM). It's frequently seen in mission-critical applications, data centers, and cloud settings.

Red Hat Enterprise Linux is well-known for its dependability and support, making it an excellent choice for enterprise-level installations. It thrives in mission-sensitive applications and situations where dependability is vital. The incorporation of Red Hat's support services, as well as technologies such as the RPM package management system, improves RHEL's efficacy. However, in budget-constrained settings, the subscription-based approach may be worth considering.

AIX (UNIX)



IBM's AIX (Advanced Interactive executive) operating system was designed for its System p and System i server hardware. AIX is a Unix-based system that offers great dependability, scalability, and performance. It is well-known for its ability to handle huge workloads, making it a popular choice for enterprise-level applications.

AIX is extremely successful in settings that require extreme scalability and performance. AIX is a server operating system developed by IBM that is particularly well-suited for managing large workloads and enterprise-level applications. Its toughness and dependability make it an excellent choice for crucial systems. However, because of its proprietary nature and specialized concentration on IBM hardware, AIX may be less popular than Linux or Windows Server.

FreeBSD



FreeBSD is a Unix-like operating system recognized for its speed, security, and powerful networking capabilities. It is based on the UNIX distribution Berkeley Software Distribution (BSD). Web servers, file servers, and network infrastructure are all common uses for FreeBSD. It is compatible with the Ports Collection, which is a method for building and installing third-party applications.

MacOS Server



MacOS Server is an extension of the macOS operating system, tailored specifically for Apple environments. It integrates seamlessly with other Apple devices and services, offering a user-friendly and intuitive design that aligns with the simplicity of macOS. This server operating system is particularly well-suited for hosting websites and managing multimedia content. It supports a range of services including file sharing, Time Machine backups, and multimedia streaming.

Criteria	Windows Server2022	macOS	Unix (Linux)	Unix (Generic)
Kernel Type	Monolithic	Hybrid	Monolithic	Monolithic/Hybrid

User Interface	GUI (Windows Interface)	GUI (Aqua)	Typically Command Line	Varies (Command Line/GUI)
Open Source	Partially (Certain components)	No	Yes	Yes
Cost	Licensing fees	Included in hardware	Often free.	Often free
Security	Focus on security features	Generally secure	Depends on distribution	Depends on distribution
Customization	Limited	Limited	Highly customizable	Highly customizable
Compatibility	Broad range of software support	Limited	Extensive support	Varies

Justification

Designing:

1. Visual Studio Integration:

Windows Server 2022 is fully compatible with Visual Studio, Microsoft's integrated development environment (IDE). This integration streamlines the development process by providing a seamless environment for coding, debugging, and deploying web applications. Developers benefit from enhanced productivity tools and features that are directly supported by the operating system, which simplifies the design and implementation of complex web solutions.

2. IIS (Internet Information Services):

Windows Server 2022 includes Internet Information Services (IIS), a robust and user-friendly web server software. IIS supports a wide range of web technologies and provides an intuitive graphical user interface for configuring and managing web applications. This built-in web server solution is highly compatible with various web development frameworks, ensuring efficient deployment and management of web content.

Publishing:

M.M.M AASHIK/E230667

WDD

1. IIS Web Server:

The inclusion of IIS in Windows Server 2022 ensures optimal performance for hosting web applications, particularly those developed using Microsoft technologies. IIS supports modern web standards and offers features such as security enhancements, scalability options, and advanced configuration settings that are crucial for maintaining high-performance web services.

2. .NET Framework:

For websites developed using ASP.NET or other Microsoft technologies, Windows Server 2022 provides a dedicated environment that fully supports the .NET Framework. This compatibility ensures that web applications perform optimally, leveraging the full capabilities of the .NET ecosystem for dynamic content delivery and advanced functionalities.

Assessing:

1. Integrated Monitoring Tools:

Windows Server 2022 is equipped with built-in monitoring tools and performance counters that facilitate thorough assessment and management of server health. These tools allow administrators to track system performance, identify potential issues, and ensure that the server operates efficiently, contributing to a stable and reliable web hosting environment.

2. Active Directory Integration:

The seamless integration with Active Directory allows for secure user authentication and access control. This feature is essential for managing user permissions and ensuring secure access to web applications. It also simplifies the administration of user accounts and roles, enhancing the overall security of the web server.

Summary:

Windows Server 2022 is a highly suitable choice for hosting a website, especially when the development stack involves Microsoft technologies. Its integration with Visual Studio, IIS, and the .NET Framework makes it an ideal platform for designing, publishing, and assessing web applications. The operating system's user-friendly interface, robust support from Microsoft, and built-in monitoring and security features contribute to a reliable and scalable web hosting solution.

Web Server

A web server is software and hardware that responds to client requests via the World Wide Web using HTTP (Hypertext Transfer Protocol) and other protocols. A web server's primary function is to show website content by storing, processing, and distributing webpages to users. Web servers, in addition to HTTP, support SMTP (Simple Mail transmission Protocol) and FTP (File Transfer Protocol), which are used for email, file transmission, and storage.



Web Server software

The software on a web server governs how the server receives and displays information over the internet. The program takes browser requests, locates the required files, and then provides them over Hypertext Transfer Protocol (HTTP). In essence, the program translates the facts on your website into understandable information for your users. Today, many web servers use HTTPS for more secure transactions. Other protocols, such as Simple Mail Transfer Protocol (SMTP) and File Transfer Protocol (FTP), are used by many web server software alternatives. Furthermore, separate software is created for different Operating Systems (OS), such as Windows or Linux.

Application Software

Application software, often abbreviated as "apps," refers to a category of computer programs designed to execute specific tasks directly for end-users. Unlike system software, which manages computer hardware and provides a platform for running applications, application software is tailored for user interaction and serves a wide array of purposes. It is the software with which users interact to accomplish various tasks on their devices, including computers, smartphones, and tablets. Here's a closer look at some key types of application software:

1. Content Management Systems (CMS):

Content Management Systems are applications that enable users to create, modify, and manage digital content on a website without requiring extensive technical expertise. These systems provide a user-friendly interface for adding, editing, and organizing text, images, and multimedia content. For instance, a marketing team can use a CMS to update website content, publish blog posts, and manage media assets efficiently, all without needing to involve developers. Popular CMS examples include WordPress, Joomla, and Drupal.

2. E-commerce Platforms:

E-commerce applications facilitate online transactions by providing features such as product catalogs, shopping carts, and secure payment processing. These platforms allow users to browse products, add items to their cart, proceed to checkout, and complete secure online payments. An online retailer might use an e-commerce platform to showcase products, manage inventory, and handle customer transactions. Notable e-commerce platforms include Shopify, Magento, and WooCommerce.

3. Web Frameworks:

Web frameworks are software frameworks designed to streamline the development of web applications by providing a structured foundation for building dynamic and interactive websites. These frameworks offer tools and libraries to manage data, handle user interactions, and build web pages efficiently. Developers leverage web frameworks to accelerate development and maintain code consistency. For example, a developer might use Django (Python) or Ruby on Rails (Ruby) to create a feature-rich web application that includes authentication, database connectivity, and dynamic content generation.

Types of web server software

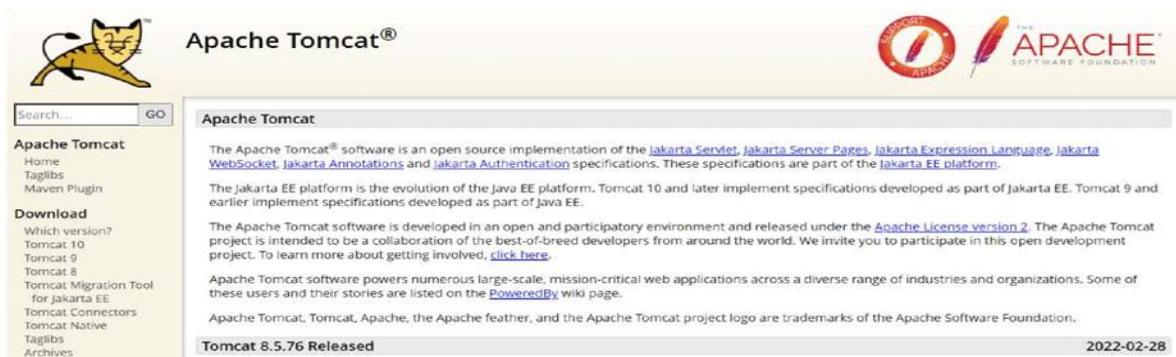
Apache



The Apache HTTP Server Project homepage features a header with the Apache logo and "HTTP SERVER PROJECT". A banner at the top reads "The Number One HTTP Server On The Internet". Below the banner, there's a sidebar for "APACHECON 2021" and sections for "Essentials" (with links to Download, About, License, FAQ, and Security Reports), "Source Repositories" (with links to General Information, Trunk, and 2.4), and "Documentation" (with a link to Version 2.4). The main content area includes a news item about "Apache httpd 2.4.52 Released" on 2021-12-20, stating it's the best available version of Apache HTTP Server. It also mentions that Apache HTTP Server version 2.4.43 or newer is required for TLS 1.3 support.

Apache is the second most widely used web server software, accounting for 31.5 percent of all known websites. It's an open-source project that runs on a variety of operating systems, including Windows and Linux. Customizability is one of Apache's finest characteristics. This server software is made up of multiple modules that may be configured to handle various requests. Furthermore, Apache supports a variety of programming languages and has a vibrant development community. Because it also supports PHP, it is a popular web server for WordPress users. Widely used for PHP and MySQL-based applications like Velvet Vogue's e-commerce site.

Tomcat



The Apache Tomcat® homepage features a header with the Apache logo and "APACHE SOFTWARE FOUNDATION". The main content area includes a search bar, a sidebar with links for "Apache Tomcat" (Home, Taglibs, Maven Plugin) and "Download" (Which version?, Tomcat 10, Tomcat 9, Tomcat 8, Tomcat Migration Tool for Jakarta EE, Tomcat Connectors, Tomcat Native, Taglibs, Archives), and a central panel for "Apache Tomcat". It states that Tomcat is an open source implementation of Jakarta EE specifications and is developed under the Apache License version 2. The panel also highlights that Tomcat powers many mission-critical web applications. A news item at the bottom right indicates "Tomcat 8.5.76 Released" on 2022-02-28.

Tomcat is an excellent web server software choice for Java applications. It makes use of several Java specs in an open-source environment that is geared for speed and performance. As a result, it's a great solution for dynamic content like animations and interactive webpages. Furthermore, Tomcat is produced by the same firm as Apache. As a result, these two server software alternatives are designed to complement one another. Furthermore, Tomcat is compatible with a variety of operating systems, including Windows, Linux, and macOS.

Designed for Java-based web applications, less common for e-commerce sites like Velvet Vogue.

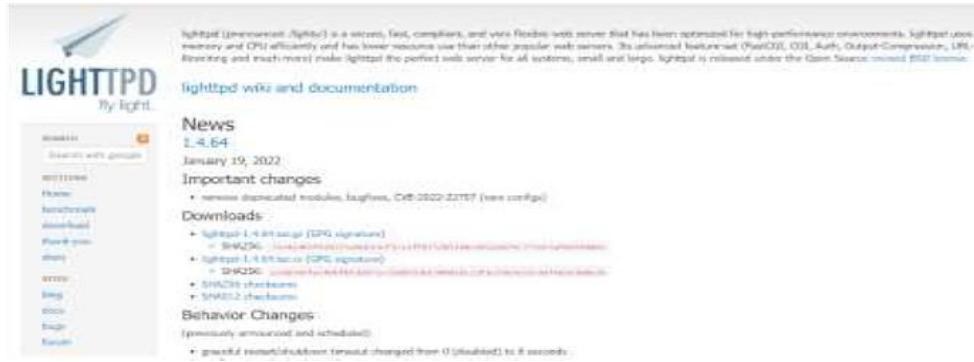
Microsoft IIS



The screenshot shows the Microsoft IIS website. At the top, there's a navigation bar with links for Home, Manage, Downloads, Learn, Reference, Solutions, Blogs, and Forums. A search bar and a sign-in link are also present. The main content area features a dark blue banner with white text: "A flexible & easy-to-manage web server...". Below the banner, a paragraph describes IIS as a flexible, secure, and manageable Web server for hosting anything on the Web. It highlights its scalability and open architecture. Two buttons are visible: "Get Started with IIS" and "Manage IIS". To the right of the text is a large, stylized graphic of a globe with a gear-like pattern. Below the banner, there are three promotional cards: one for "LeanServer" (struggling with IIS performance issues), one for "Signtech" (simplifying contracts and hires with e-signatures), and one for "Bold BI" (embedding dashboards in Angular applications).

Microsoft IIS is an exceptional server software solution created exclusively for Windows. As a result, several native Windows security capabilities, such as Azure Active Directory, are included. It also has integrated website and server administration features. Microsoft IIS is another excellent alternative for developers. The program features native support for dynamic ASP.NET applications in CSS, JavaScript, and HTML. It's also quite simple to use for both novice and intermediate users.

Lighttpd



The screenshot shows the Lighttpd official website. The header includes the Lighttpd logo and a "By light." tagline. Below the header, there's a brief introduction about Lighttpd being a secure, fast, compliant, and very flexible web server. A "lighttpd wiki and documentation" link is provided. The main content area is divided into sections: "News" (listing version 1.4.64 released on January 19, 2022), "Important changes" (listing various module updates and signature improvements), "Downloads" (links to different versions of Lighttpd), and "Behavior Changes" (listing changes related to timeouts and graceful restarts).

Lighttpd (pronounced "Lighty") is one of the greatest low-resource web server software alternatives. It requires less memory and provides performance enhancements to relieve strain on the physical server. Lighttpd may also deliver dynamic and static information independently to improve speed on database sites. Lighttpd is also designed to work on Linux. It is also compatible with Windows, albeit native support is lacking. Furthermore, Lighttpd supports a number of programming languages, including PHP, Python, Lua, and Ruby.

NGINX



NGINX. It just surpassed Apache and is now utilized by more than 33% of all known websites. NGINX is also compatible with both Linux and Windows. NGINX is likely the most well-known for its high-performance capabilities. The web server software is appropriate for high-traffic sites and demanding conditions since it is built to manage many connections at the same time. With NGINX, you'll enjoy faster loading times while using less server resources.

Xampp Server



XAMPP is a cross-platform, free, and open-source web server solution designed to simplify the process of setting up a local server environment. The name XAMPP stands for **Cross-Platform**, **Apache**, **MySQL**, **PHP**, and **Perl**. Created by Apache Friends, XAMPP provides a comprehensive stack of software components that allow developers to develop, test, and deploy web applications locally before moving them to a live server.

XAMPP is a versatile and user-friendly tool that provides a complete local server environment for web development and testing. Its inclusion of Apache, Maria DB, PHP, Perl, and additional tools like phpMyAdmin and Tomcat makes it a powerful solution for building and managing web applications. XAMPP's ease of installation and cross-platform support allow developers to quickly set up a testing environment and transition smoothly from local development to live deployment..

Comparison of server Software

Feature	Apache	Tomcat	Xampp
Primary Use	Web server	Servlet container	Web development environment with Apache, MySQL, PHP, and more.
Functionality	Serves static content, handles HTTP requests	Executes Java servlets and JSP pages	Bundled package for web development, including Apache, MySQL, PHP, etc..
Programming Language Support	Primarily for serving content written in various programming languages	Specifically designed for Java applications	Supports multiple programming languages, including PHP and Perl
Database Support	Can be configured to work with various databases	Can connect to databases but is not a database server itself	Includes MariaDB (formerly MySQL) for database functionality
Ease of Setup	Generally straightforward	Requires Java knowledge for configuration	All-in-one package, relatively easy to set up
			compared to individual components

Versatility	Can handle various types of content	Specialized for Java applications	All-in-one solution for web development with multiple components
Common Use Cases	Hosting websites, serving static content	Running Java web applications	Local web development, testing, and hosting environments
Community Support	Large and active community	Active Java community	XAMPP has a supportive community, but individual components have their communities
Portability	Portable across different operating systems	Portable but requires Java installation	Portable, designed to work on various operating systems

Designing:

- **Extensive Configuration Options:** Apache provides a highly customizable server environment, allowing developers to fine-tune server settings to meet specific requirements during the design phase. This flexibility ensures that the server can be tailored to various use cases and performance needs.
- **Support for Dynamic Content:** Apache excels in hosting dynamic websites, thanks to its support for a wide range of technologies, including PHP, Python, and more. This makes it an ideal choice for developing complex web applications that require dynamic content generation.

Publishing:

- **Stability and Reliability:** Apache is renowned for its stability and reliability, which are crucial for publishing websites. Its robust performance ensures that websites remain accessible and functional, even under high traffic conditions.
- **Compatibility with Web Development Stacks:** Apache supports various web development stacks and frameworks, making it compatible with a diverse array of

technologies. This compatibility allows for seamless integration with different development tools and applications.

Assessing:

- **Detailed Logging Capabilities:** Apache offers comprehensive logging features, enabling detailed tracking of server activity and user interactions. This data is invaluable for assessing server performance, diagnosing issues, and understanding user behavior.
- **Performance Monitoring and Optimization:** Apache provides numerous modules for monitoring and optimizing server performance. These tools help administrators to manage server resources effectively and ensure optimal operation.

Additional Benefits:

- **Scalability:** Apache's scalable architecture allows it to handle increased traffic and resource demands efficiently. This scalability makes it suitable for both small-scale and enterprise-level applications.
- **Robust Security Model:** Apache has a strong security framework, and regular updates address potential vulnerabilities promptly. The server supports SSL/TLS encryption, ensuring secure data transmission between clients and the server.
- **Active Community Support:** Apache benefits from a large and active community that offers extensive support for troubleshooting and optimizing server configurations. This community-driven support enhances the server's reliability and ease of use.

In summary, Apache stands out as a versatile and dependable web server solution for designing, publishing, and assessing websites. Its flexibility, stability, and extensive feature set make it particularly well-suited for websites requiring dynamic content and diverse technological integrations. Apache's robust performance, scalability, and security features further reinforce its suitability for a wide range of web hosting needs.

	Designing	Publishing	Accessing
Communication Protocols	<ul style="list-style-type: none"> Choose between HTTP and HTTPS based on the sensitivity of the data Consider the security needs of the website 	<ul style="list-style-type: none"> Ensure secure data transfer during file uploads using HTTPS Implement proper communication protocols for data integrity 	<ul style="list-style-type: none"> Users access the website using HTTP/HTTPS protocols. HTTPS is crucial for secure and encrypted communication.
Server Hardware	<ul style="list-style-type: none"> Understand the expected traffic and resource needs to appropriately size the server. Decide between shared hosting for cost-effectiveness or dedicated hosting for high performance. 	<ul style="list-style-type: none"> Optimize server hardware for efficient handling of website content. Consider scalability for future growth. 	<ul style="list-style-type: none"> Better user experience and performance with appropriately configured server hardware
Operating Systems	<ul style="list-style-type: none"> Choose an operating system based on compatibility with web development technologies. Consider the security features and a simple development. 	<ul style="list-style-type: none"> Ensure that the selected operating system supports the required server software. Consider system stability and security during deployment. 	<ul style="list-style-type: none"> The choice of the operating system on the server may impact user experience
Web Server Software	<ul style="list-style-type: none"> Choose web server software based on 	<ul style="list-style-type: none"> Configure the selected web server software to 	<ul style="list-style-type: none"> Different web server software may have

Analysis of the Impact of Web Development Technologies and Frameworks on Web Design, Functionality, and Management

Web development technologies

Web development technologies relate to the various programming languages and tools used to create dynamic and feature-rich websites and applications. Client-side technologies are those that are used to construct and display everything with which the end user interacts. We have competence in a variety of web development technologies, which allows us to maintain and improve our high-quality online projects.



Front-end (client-side) technology.

Front-end technologies are used on the "client side" of your website or application. They are used to create the interactive components of your website and the items that people view and interact with. This covers text colors and styles, graphics, buttons, and navigation menus.



Back-end (server-side) technologies.

Back-end technologies are for the "server side" of your website or application. They're for developing the technical foundation. They store and arrange data and make sure everything on the front-end works. For example, when a user provides login credentials to a social media application, back-end technologies are used to check if those credentials are accurate. Once the credentials are verified, the server will send back the profile name, picture, and other associated information.



Impact on Web Design

HTML



HTML is an abbreviation for Hyper Text Markup Language. It is a crucial technology necessary for web development. The backbone of web content structure, supporting multimedia, semantic elements, and responsive design. It serves as the foundation for a web page. HTML coding guarantees that all of a website's content is correctly structured. This is done so that your Internet browser can show the material correctly. A browser could not display text or load photos or other content without HTML.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          </head>
5      <body>
6          <h1>My First Page</h1>
7          <p>This is my first page.</p>
8          <h2>A secondary header.</h2>
9          <p>Some more text.</p>
10         </body>
11     </html>
```

CSS (Cascading Style Sheets)



Cascading Style Sheets, abbreviated as CSS, define the style and aesthetics of a web page. While HTML is used to structure a web page, CSS specifies the appearance of that structure. This includes page layouts, colours, fonts and element positioning. If HTML is the bones of the web page, CSS is the skin. It makes the Internet, and your website, look good. Enables websites to adapt seamlessly to mobile, tablet, and desktop views, improving user experience.

```
redTitle {
  color: red;
}
greenTitle {
  color: green;
}
blueTitle {
  color: blue;
}
centerTitle {
  text-align: center;
}
rightTitle {
  text-align: right;
}
leftTitle {
  text-align: left;
}
giantTitle {
  font-size: 50px;
}
bigTitle {
  font-size: 30px;
}
normalTitle {
  font-size: 15px;
}
smallTitle {
  font-size: 5px;
}
tinyTitle {
  font-size: 1px;
}
```

JavaScript



For several reasons, JavaScript is the most popular programming language among entrepreneurs. It may be used for both front-end and back-end development. When compared to other languages, JavaScript is very simple to learn (though all languages have certain obstacles), and it is widely utilized in bespoke web application development. It is also being regularly updated and enhanced with new features Create dynamic and interactive interfaces. Enhance user engagement with features like real-time updates and smooth navigation transitions.

```
web ▶ src ▶ views ▶ JS ToastView.js ▶ 🐛 ToastView ▶ ⏺ render
1 // @ts-check
2
3 import React from "react";
4 import { Paper, Button } from "@material-ui/core";
5
6 export class ToastView extends React.Component {
7   render() {
8     return (
9       <Paper>
10      <div>
11        <Button>Delete</Button>
12      </div>
13    )
14  }
15}
16
17
18 }
```

Programming Languages

A programming language is a computer language used by programmers to create software programs, scripts, or other sets of instructions that computers can execute. Despite the fact M.M.M AASHIK/E230667

WDD

that many languages have commonalities, each has its unique syntax. After learning the language's rules, grammar, and structure, a programmer writes the source code in a text editor or integrated development environment (IDE). The programmer will then frequently convert the code into machine language that the computer can understand. Scripting languages that do not require a compiler run the script using an interpreter.

Examples of popular programming languages include Python, Java, C++, JavaScript, and Ruby. Each language has its own strengths and weaknesses and is suited for different types of projects.

Some of the key features of programming languages include:

- **Syntax:** The specific rules and structure used to write code in a programming language.
- **Data Types:** The type of values that can be stored in a program, such as numbers, strings, and booleans.
- **Variables:** Named memory locations that can store values.
- **Operators:** Symbols used to perform operations on values, such as addition, subtraction, and comparison.
- **Control Structures:** Statements used to control the flow of a program, such as if-else statements, loops, and function calls.
- **Libraries and Frameworks:** Collections of pre-written code that can be used to perform common tasks and speed up development. (SanghpriyaGautam, 2023)

As we saw in the above topic, the language used to code the front end of the website. These included HTML, CSS and Java script. The languages used in the front end cannot be used in the back end too. JavaScript is an exception because it plays the role of both front and back end language. Programming language used in server side is mainly focused on choosing which content to show in the browser as responds to requests. The server side codes also handles job's such as validating requests and responds, linking database to store and retrieve data and sending the correct respond to the made requests. There are many languages used to code the server side of the website. Some of them are:-

Database



Databases are crucial for managing and storing user information and data efficiently. They facilitate the creation, access, and modification of customer details, ensuring that data is organized, easily retrievable, and systematically linked across various applications.

Dynamic websites often rely on databases to deliver content to users without altering the original HTML pages. This approach allows for real-time updates and interactions while maintaining a consistent and stable web interface. Databases generally fall into two categories:

SQL Databases: These databases use a structured query language (SQL) for defining and manipulating data. They offer a rigid schema with predefined data structures and support data validation and integrity. Common SQL databases include:

MySQL: A popular, open-source SQL database known for its reliability and ease of use.

Oracle: An enterprise-level SQL database providing robust features for large-scale applications.

SQL Server: A Microsoft-managed SQL database that integrates well with other Microsoft products and services.

NoSQL Databases: These databases offer more flexibility in data storage and retrieval, making them suitable for applications with varied and dynamic data requirements. They do not use a fixed schema and can handle unstructured data effectively. Notable NoSQL databases include:

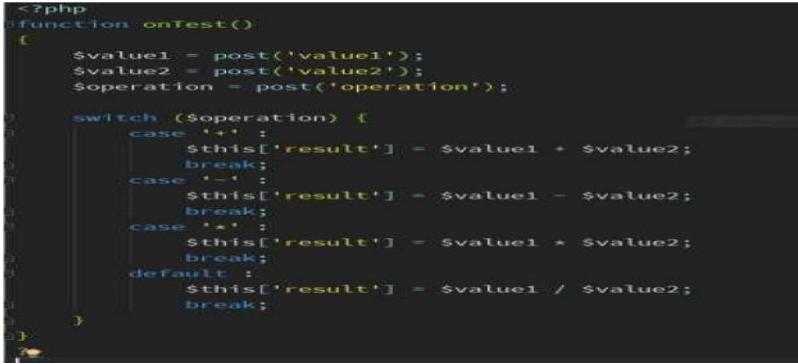
MongoDB: An open-source NoSQL database that stores data in flexible, JSON-like documents, allowing for dynamic and scalable data models.

By utilizing these database technologies, developers can efficiently manage and deliver content, ensuring that applications perform optimally and provide a seamless user experience.

PHP



PHP is the most widely used programming language in web development. PHP is the foundation of WordPress, the world's most popular content management system (CMS). Finding a decent developer may be challenging regardless of the language, but using a popular one, such as PHP, can provide you with a wider pool of developers to pick from and work with. In addition, PHP includes substantial online documentation and tutorials for troubleshooting and support



```
<?php
function onTest()
{
    $value1 = post('value1');
    $value2 = post('value2');
    $operation = post('operation');

    switch ($operation) {
        case '+':
            $this['result'] = $value1 + $value2;
            break;
        case '-':
            $this['result'] = $value1 - $value2;
            break;
        case '*':
            $this['result'] = $value1 * $value2;
            break;
        default:
            $this['result'] = $value1 / $value2;
            break;
    }
}
```

Difference between Libraries and Frameworks

Even though libraries and frameworks have the same function which is to provide a set of codes for the purpose of reusing it still has its differences. Frameworks can be anything such as a library, a set of library or a collection of scripts. On the other hand library can be a collection of objects, functions and methods. However there is a main technical difference between frameworks and libraries.

Inversion of control is a term which represents the main technical difference of frameworks and libraries. In library the developer has the freedom to choose when and where the library has to be called, but in framework, the framework itself takes care of that job and chooses the place to plug in the code.

With this different the developer has to choose frameworks or libraries based on their practice and interest.

Data Formats

As all other sectors in IT, web development too has its data formats. These data formats are used to communicate with each other. Therefore only a limited amount of data formats are used because it has to be general in order to the browsers to understand. The most common formats used in modern APIs are JSON (JavaScript Object Notation) and XML (Extensible Markup Language).

Web Development Frameworks



Web development encompasses the creation and maintenance of websites and web applications, ranging from simple static pages to complex dynamic platforms. It involves various technologies and frameworks that aid in the development process, providing structured methods for delivering web resources, services, and APIs. Web development frameworks are categorized into two main types: **Front-end frameworks** and **Back-end frameworks**

Frontend frameworks

Frontend frameworks are the fundamental building blocks of the software development process. But there are so many alternatives when it comes to creating visually stunning apps with a great user experience. To assist you, we have compiled a list of the finest frontend frameworks for 2024.

1. Angular



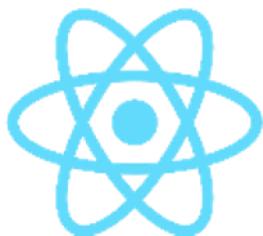
Angular is a well-known open-source framework for developing dynamic web apps. Google created and maintains Angular, which utilizes TypeScript as its core programming language. It adheres to the Model View-Controller (MVC) design, which assists developers in developing modular, manageable, and scalable applications. Angular has a number of capabilities, like as two-way data binding, dependency injection, and a sophisticated templating engine, which make it useful for developing single-page apps with a rich user experience.

2. Bootstrap



Bootstrap is a free and open-source front-end framework developed by Twitter. It is intended to make it easier to create responsive and mobile-first websites. Bootstrap comes with a library of pre-designed HTML, CSS, and JavaScript components such as navigation bars, buttons, and forms that can be quickly changed and incorporated into online applications. Bootstrap streamlines the process of creating visually beautiful and functioning websites by delivering a uniform and adaptable design across multiple devices and screen sizes. This eliminates the need for substantial custom style or scripting.

3. React

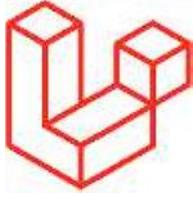


one of the simplest frameworks to learn, was created at Facebook to address code maintainability difficulties caused by the app's rapid addition of new features. React, currently an open-source framework, distinguishes out due to its virtual Document Object Model (DOM), which provides extraordinary capabilities. An excellent framework for people that expect huge traffic and want a reliable base to handle it.

Backend frameworks

Backend frameworks serve as the foundation for building powerful and scalable online applications, as well as handling server-side operations and database management. Choosing the proper backend framework may improve your development efficiency and the overall performance of your online apps dramatically. Let's take a look at some of the best backend frameworks to assist you make an educated pick.

1. Laravel



PHP is used as a server-side programming language on 76.8% of websites. PHP and its frameworks have been the favored choice for web developers due to its shorter turnaround time and increased security. If you've been looking for a PHP backend framework, Laravel is one of the best choices you can make. This open-source is intended for symphony-based web apps. Laravel is the best choice if you want to focus on back-end development

2. Ruby on Rails



The Ruby on Rails framework is well-known for its full-stack capabilities as well as its ease of use. The framework's goal is to make creating web apps easier than ever before by removing a lot of the guesswork. The framework offers the infrastructure (components), allowing you to concentrate on what is important: your application's code. Even though it is not a commonly used programming language, it is a popular backend framework since it assists developers during the development process with ease, automated testing, and libraries.

3. ASP.NET Core



ASP.NET Core is an open-source, cross-platform framework for developing .NET applications. This framework is a lightweight implementation of the Microsoft server stack that allows you to design and operate applications across several platforms and environments. Many programming languages, including C#, F#, Visual Basic, NodeJS, and JavaScript, may be used to create web apps and API servers

Impact of Web Development Technologies and Frameworks on Website Design, Functionality, and Management

The impact of common web development technologies and frameworks is profound, significantly influencing various facets of website design, functionality, and management. HTML, CSS, and JavaScript, the foundational trio of web development, collectively shape the design of a website. HTML provides the structural framework, CSS contributes to the visual aesthetics, and JavaScript enhances interactivity, resulting in a seamless and engaging user experience. This combination ensures a well-designed and visually appealing website that caters to the expectations of modern internet users. Frameworks, such as Bootstrap, further augment the impact on both design and functionality. Bootstrap, a responsive front-end framework, simplifies and accelerates the design process by offering a responsive grid system and pre-designed components. This not only ensures consistency in design but also enhances the website's responsiveness across different devices and screen sizes. The impact on functionality is significant, with Bootstrap providing features like modals or carousels that contribute to a dynamic and user-friendly interface. This framework streamlines development, resulting in a website that is both visually cohesive and functionally robust. In terms of website management, the use of server-side scripting languages like PHP plays a crucial role. PHP facilitates dynamic content generation, database interaction, and the overall processing of data. It enhances the website's functionality by enabling features like user authentication, form handling, and data storage. Additionally, the utilization of frameworks for back-end development, such as Laravel or Django, further streamlines management tasks by providing structured architectures, security features, and reusable components. These frameworks contribute to efficient coding practices and effective website management. The incorporation of Content Management Systems (CMS) is another aspect that impacts website management. CMS like Word Press or Joomla simplify content creation, editing, and publication, empowering non-technical users to manage website content easily. They offer features like user roles, version control, and plugin integrations, contributing to efficient content management and overall website administration. In conclusion, the impact of common web development technologies and frameworks is multi-faceted, influencing the design, functionality, and management of a website. The combination of HTML, CSS, and JavaScript shapes the user interface, while frameworks like Bootstrap enhance design

consistency and functionality. Server-side scripting languages and back-end frameworks contribute to dynamic functionality and efficient management. Additionally, the integration of CMS streamlines content management, making the website more accessible and manageable for both developers and content creators. The strategic selection and utilization of these technologies are crucial for creating websites that not only meet aesthetic standards but also excel in functionality and ease of management.

A Review of the Influence of Search Engines on Website Performance

Search Engines



A search engine is a sophisticated software system designed to explore and retrieve information from a vast database based on user queries, presenting a list of results that best match the user's search intent. Among the pioneering search engines, Archie, created for locating FTP files, and Veronica, the first text-based search engine, laid the groundwork for the field. Today, Google reigns as the most prominent and widely-used search engine, known for its efficiency and comprehensive search capabilities. Other notable search engines include AOL (America Online), Ask.com, Baidu, Bing, DuckDuckGo, and Yahoo, each offering unique functionalities and user experiences.

The performance of search engines profoundly affects multiple facets of the digital landscape. User experience is directly impacted by speed and responsiveness; slow search results can frustrate users and lead to abandonment. User satisfaction hinges on the relevance and accuracy of search results, as users rely on search engines to quickly locate information matching their queries. The effectiveness of search engine ranking algorithms influences web traffic distribution, directing visitors to high-quality, relevant content. For search engines that monetize through advertising, performance

affects ad effectiveness, with advertisers favouring platforms with a large, engaged user base and precise ad targeting.

Search engine performance also determines the efficiency of information retrieval, impacting search accuracy through effective indexing, quick retrieval, and relevant ranking. Trust and credibility are bolstered by a search engine's commitment to security and privacy, essential for user confidence.

(computerhope, 2024)

Popular search engines

- **Google**



Google leads the search engine rankings by an astounding 88.57% over second-place Bing. Google was founded 25 years ago (1998) and soon rose to prominence as the leading search engine due to its capacity to provide accurate responses to users' searches in the shortest amount of time. According to Statista and Stat counter data, Google controls the market in all countries across all devices (desktop, mobile, and tablet). Google's desktop market share is 83.84%, while its mobile market share is 94.94%.

- **Bing**



Microsoft Bing is the greatest Google alternative search engine. Bing's search engine share has fluctuated between 2.83% and 12.31% over the years. Bing is Microsoft's attempt to compete with Google in search, however despite their efforts, they were unable to persuade people that their search engine could be as dependable as Google.

- **Yahoo**



Yahoo is one of the most popular email providers, and its web search engine holds third place in search with an average of 1.24% market share. From October 2011 to October 2015, Yahoo search was powered exclusively by Bing. In October 2015, Yahoo agreed with Google to provide search related services, and until October 2018, the results of Yahoo were powered by Google. As of October 2019, Yahoo! Search is once again provided exclusively by Microsoft Bing.

- **Baidu**



Baidu has a global market share ranging from 0.68% to 11.26%. Baidu, China's most popular search engine, was created in 2000. According to Wikipedia, Baidu serves billions of search queries every month, and its market share is rapidly expanding. It is presently placed sixth in Wikipedia's Website Rankings.

- **Yandex**



Russia's most popular search engine, controls between 0.5% and 1.16% of the worldwide market. According to Wikipedia, Yandex.ru is one of the top ten most popular websites on the Internet, ranking eighth in Russian. Yandex positions itself as a technology firm that creates intelligent products and services based on machine learning.

Effects of search engines on website performance

Search engines have a big influence on website performance, impacting things like visibility, traffic, and overall user engagement. Websites that are successfully indexed and ranked by search engines are more likely to receive organic traffic, which improves their online visibility. Search engine optimization (SEO) strategies are critical in aligning a site's content and structure with search engine algorithms, which directly affects its ranking. Poor SEO, on the other hand, might result in decreased exposure and traffic. Furthermore, user experience is prioritized by search engine algorithms, so websites with faster loading speeds, mobile responsiveness, and high-quality content tend to do higher. In conclusion, search engines have a substantial impact on website performance, stressing the need of smart SEO efforts for online success.

Search Engine Optimization (SEO)



Search Engine Optimization (SEO) is the process of enhancing a website's visibility to users through their search queries. With numerous search engines available, each utilizing its own algorithms to rank websites, SEO becomes crucial for web developers aiming to achieve a higher ranking in search results. The primary goal is to attract more visitors by ensuring that a website appears prominently when users search for relevant terms.

SEO works by increasing both the quality and quantity of organic traffic to a website. To effectively optimize a site, it's important to understand the rules and regulations of the search engine algorithms in use. For this discussion, we'll focus on Google's algorithm due to its prominence in the industry, though similar principles apply to other search engines.

Key Processes in Search Engine Optimization

- Crawling:** This is the initial step where search engine bots, also known as crawlers or spiders, discover web pages on the internet. These crawlers navigate websites to collect data, including newly updated content from existing pages, pages discovered through links, and new pages submitted via sitemaps. It is crucial for crawlers to understand the content on your site accurately. Ensuring that all web

pages are linked to the homepage helps crawlers access and index every page effectively.

2. **Indexing:** After crawling, the collected data is stored in an index. This process involves analyzing the content of web pages, including text, images, and videos, to understand their relevance and context. This indexed information is then stored on servers, making it available for retrieval during searches.

3. **Ranking:** In this stage, the search engine ranks websites based on various factors, including the relevance of content, user location, language, and device used. When a keyword is entered, the search engine queries its index and ranks the results according to its algorithm, presenting the most relevant pages at the top of the search results.

Effective SEO Strategies

- **Content Optimization:** Incorporate relevant keywords throughout your webpages to enhance visibility. High-quality, informative content that answers user queries is key.
- **Content Updates:** Regularly update your website with fresh and relevant content. Search engines favor sites that offer up-to-date information, which can improve rankings.
- **Meta Data:** Utilize meta tags effectively. The meta title, description, and keyword tags within the HTML head section help search engines understand the content of your pages and improve search visibility.
- **Alt Tags:** Include alt text for images to describe their content. This helps search engines index your images and provides additional context.

While there are many other SEO techniques available, focusing on these fundamental strategies can significantly enhance your website's ranking and visibility. An optimized website not only performs better in search engine rankings but also offers a better user experience, leading to increased traffic and engagement.

Google users make billions of searches for information and goods every day. It's no surprise that search engines are one of the most important sources of traffic for websites. Search engine optimization is essential for increasing your ranking places. More traffic equals higher ranks. More traffic equals more consumers and greater brand recognition.

A Review of How to Improve the Index Value of a Website through SEO

Site Index

A site index, often known as a sitemap, is a systematic list or map of the pages on a website. It acts as a guide for both visitors and search engines, explaining a site's content arrangement and hierarchy. This tool facilitates navigation by assisting users in easily locating certain information or pages of interest. Furthermore, search engines employ site indexes to more effectively crawl and index web pages, increasing the site's exposure in search results. Overall, a well-designed site index adds to a better user experience and improved search engine optimization for a website.

Improving site index and value rank

- Create an XML sitemap outlining your website's structure and submit it to search engines like Google. This helps search engines comprehend how your site is organized
- Make certain that your material is of good quality, relevant, and helpful to your intended audience. To make your material more search engine friendly, use correct headers, Meta tags, and keyword optimization.
- Make sure your website is mobile-friendly, as search engines like responsive sites. A mobile-friendly design enhances the user experience and can boost your search rankings.
- Optimize the loading speed of your website, as slow-loading pages can have a detrimental impact on user experience and search rankings. To boost performance, compress pictures, use browser caching, and consider a content delivery network (CDN).
- To ensure a safe connection, make sure your website uses HTTPS. Secure sites may be prioritized by search engines, and it also installs trust in your users.
- Include social media sharing buttons on your website. Social signals may add to the overall authority and worth of your site.

- Create a solid internal connection framework. To help people and search engines navigate your site, link pertinent sites inside your content. This aids in the distribution of authority and improves indexation.

Key Areas of Influence

Ways to Improve Site Indexing and Value Rank

A. On-Page SEO

1. **Quality Content:** Quality, authoritative content is crucial for improving search engine rankings. Well-crafted content tailored to your audience increases site traffic, enhances your site's authority, and boosts relevance. Aim to present yourself as an expert in your topic.
2. **Keyword Optimization:** Effective keyword optimization helps search engines understand the relevance of your content. Conduct keyword research and strategically place relevant keywords in your content, Meta tags, and headers. For example, use specific keyword phrases like "online masters in engineering management" or "how to apply for scholarships." High-quality, original content helps Velvet Vogue rank higher for relevant searches like "trendy casualwear."
3. **Optimize Meta Tags:** Meta tags, including titles and descriptions, are important for search engine visibility. Craft concise, compelling, and keyword-rich Meta tags to provide valuable information about your pages, improving click-through rates from search results.
4. **Image Optimization:** Optimize images with descriptive alt text and file names to enhance accessibility and provide additional context for search engines. This also improves site speed and overall SEO.
5. **Internal Linking:** Internal linking connects different pages on your site, distributing page authority and aiding navigation. This practice helps search engine crawlers discover and index content, improving user experience and SEO.

6. **Page Load Speed:** Fast page load speeds enhance user satisfaction and reduce bounce rates, which positively impacts search engine rankings. Ensure your pages load quickly to provide a better user experience.

B. Off-Page SEO

1. **Backlink Building:** Acquiring high-quality backlinks from reputable sources enhances your site's authority and improves search rankings. Backlinks signal to search engines that your content is valuable and authoritative.
2. **Social Signals:** Social signals, such as likes, shares, and comments, reflect content popularity and can influence search rankings. Integrating social media into your SEO strategy can boost visibility and drive traffic. Social media mentions and shares can indirectly improve visibility.
3. **Local SEO:** For businesses targeting local audiences, optimizing for local searches and managing local business listings are essential. Positive local reviews and accurate location-based information improve visibility in local search results.
4. **User Experience (UX):** A positive user experience, including easy navigation and clear design, encourages engagement and can indirectly impact search rankings. A well-designed UX leads to lower bounce rates and higher rankings.

C. Technical SEO

1. **XML Sitemap:** A well-structured XML Sitemap helps search engines crawl and index your site efficiently by providing a roadmap of your pages.
2. **Robots.txt:** The robots.txt file guides search engine crawlers on which pages to crawl or avoid, improving crawl efficiency and focusing on essential content.
3. **Crawl Errors:** Identifying and fixing crawl errors is crucial for maintaining a healthy website. Unresolved errors can hinder search engine crawlers and impact overall visibility.
4. **SSL Certificate (HTTPS):** Securing your site with an SSL certificate is both a security measure and a ranking factor. Search engines prioritize secure sites, providing a safer browsing experience and boosting rankings.

Techniques to Improve SEO

1. **Crawl Budget Optimization:** Improve crawl efficiency by minimizing duplicate content, fixing broken links, and ensuring a logical site structure.
2. **Internal Linking:** Use internal links to connect relevant pages, aiding in content discovery and indexing by search engines.
3. **Mobile-Friendly Design:** Optimize your site for mobile devices, as Google primarily uses the mobile version for indexing.
4. **Page Load Speed:** Ensure fast loading times, as page speed is a ranking factor and influences user experience.
5. **Quality Content:** Regularly publish high-quality, relevant content that provides value to users.
6. **Keyword Optimization:** Incorporate relevant keywords naturally into content, meta tags, and headers.
7. **User Experience (UX):** Enhance UX with a clean, intuitive design to improve engagement and reduce bounce rates.
8. **SSL Certificate (HTTPS):** Secure your site with an SSL certificate to enhance security, user trust, and SEO rankings.

SEO Tools

Google Search Console: Monitors indexing issues and provides performance insights.
SEMrush/Ahrefs: Analyzes keywords, backlinks, and competitors' SEO strategies.

Performance Optimization

Using a Content Delivery Network (CDN) improves load times globally. Compressing images and leveraging browser caching reduces server load.

Mobile Optimization

M.M.M AASHIK/E230667

WDD

Responsive design ensures smooth navigation across devices. AMP (Accelerated Mobile Pages) speeds up mobile page load times.

Website Speed and Performance

Search engines prioritize fast-loading websites to ensure a good user experience.

- Google Core Web Vitals

Metrics like Largest Contentful Paint (LCP) and First Input Delay (FID) measure user experience. Websites with slow load times are penalized with lower rankings, affecting Velvet Vogue's traffic.

- Impact on Bounce Rate

A slow website leads to higher bounce rates, signaling poor performance to search engines. Conversely, fast-loading pages retain users, improving engagement and search rankings.

Mobile-Friendliness

Search engines prioritize mobile-responsive websites due to the growing number of mobile users.

- Mobile-First Indexing

Google primarily uses the mobile version of a website for indexing and ranking. Ensuring that Velvet Vogue's e-commerce site is responsive and mobile-friendly boosts rankings and accessibility.

Content Quality and Relevance

Search engines favor websites with relevant, high-quality content.

- Freshness Algorithm

Regularly updated content, like product descriptions or blog posts, helps maintain visibility. Relevant keywords aligned with user intent improve search rankings.

- Rich Snippets

Use of structured data (e.g., JSON-LD) enhances search engine understanding and improves click-through rates with rich results, like product reviews and pricing.

User Behavior and Engagement

Search engines analyze user behavior to assess a website's quality.

- Click-Through Rate (CTR)

A higher CTR for search results improves rankings. Optimized Meta descriptions and titles attract more clicks.

- Dwell Time

The time users spend on a website indicates its relevance and quality. Engaging product pages and interactive elements on Velvet Vogue can improve this metric.

- Bounce Rate

High bounce rates negatively impact rankings. A seamless shopping experience reduces this risk.

Discussion of the Capabilities and Relationships of Front-End and Back-End Website Technologies and Their Relation to Presentation and Application Layers

Front End (Presentation Layer)

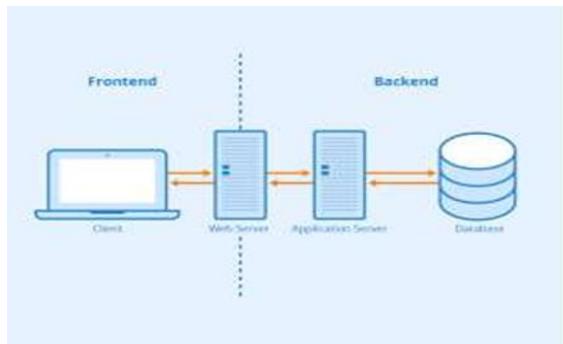
The **front end of a web application** is the part that users see and directly interact with. It encompasses all visual and interactive components, such as buttons, menus, forms, animations, and multimedia content. Essentially, the front end creates a seamless, visually appealing, and user-friendly experience that bridges the gap between users and the application's back-end processes. It is responsible for ensuring smooth communication, intuitive navigation, and an engaging interface that enhances user satisfaction.

At its core, the front end is built using three main technologies: **HTML**, **CSS**, and **JavaScript**. **HTML (HyperText Markup Language)** provides the structural foundation of the application, organizing content into logical sections such as headings, images, and links. **CSS (Cascading Style Sheets)** handles the styling and layout, defining colors, fonts, and page arrangements. It also ensures responsiveness, allowing the web application to adapt seamlessly across different devices and screen sizes. Meanwhile, **JavaScript** introduces interactivity, enabling features such as real-time updates, dynamic animations, and form validations, ensuring that the application reacts promptly to user inputs.

Modern front-end development often leverages frameworks like **React.js**, **Angular**, or **Vue.js**, which simplify the creation of dynamic and modular user interfaces. These tools allow developers to build reusable components, speeding up development and ensuring scalability. Additionally, CSS preprocessors like **SASS** and bundlers like **Webpack** enhance styling and asset management, streamlining workflows.

A key aspect of the front end is the balance between **graphic design** and **user interface functionality**. Graphic design ensures the website is visually appealing and aligned with the brand's identity, while the user interface focuses on usability and seamless interaction. Both elements work together to provide an intuitive and engaging user experience. Collaboration between designers and developers is essential to translate creative designs into functional and interactive interfaces.

In conclusion, the front end serves as the gateway to the web application, shaping the first impression and overall user experience. By integrating design aesthetics, responsive layouts, and dynamic features, the front end ensures users can easily navigate and interact with the application, creating a lasting impact on engagement and satisfaction.



Front-End Web Technologies:

Front-end web technologies are the tools, languages, and frameworks used to construct the user interface and user experience of a website or web application. These technologies are centered on client-side development, dealing with what users see and interact with directly in their web browsers.

- **HTML (Hypertext Mark-up Language)**: Provides the structural backbone of web pages, defining the content and layout.

```

1 <div id="navWrap">
2   <nav class="navbar navbar-expand-md p-3">
3     <div class="container-fluid">
4       <a class="navbar-brand" href="#">
5         
6       </a>
7       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navb
8         <span class="navbar-toggler-icon"></span>
9       </button>
10      <div class="ms-auto">
11        <i class="fas fa-arrow-right" id="toggleRightMenu"></i>
12      </div>
13    </div>
14  </nav>

```

- **CSS (Cascading Style Sheets)**: Adds aesthetic styling, controlling the presentation and formatting of elements to achieve a visually pleasing design.

```

1 .nav {
2   margin-right: 22px;
3 }
4 .verticalNav {
5   width: 28px;
6   position: fixed;
7   height: 100vh;
8   top: 0;
9   right: 0;
10  writing-mode: vertical-rl;
11  text-orientation: upright;
12  line-height: 40px;
13  text-transform: uppercase;
14  cursor: pointer;
15  transition: all 300ms ease-in;
16  border-right: 1px solid #222222;
17 }

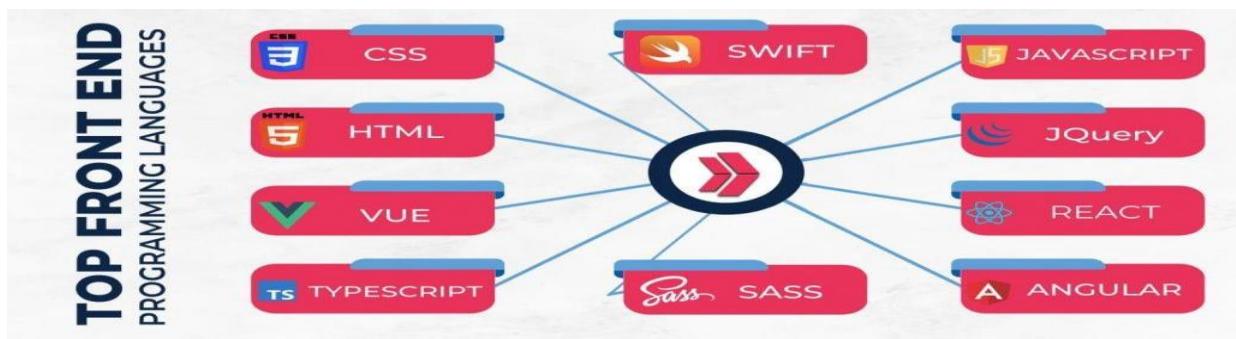
```

- **JavaScript**: Introduces interactivity, enabling dynamic behaviour, real-time updates, and interactive features. Together, these technologies form the foundation

of front-end development. HTML structures the content, CSS manages the layout and appearance, and JavaScript enables dynamic content and interactivity. This synergy creates engaging and user-friendly interfaces, essential for a positive user experience.

```
1 const LOCALE = globalThis.navigator.language
2
3 const div = document.body.appendChild(document.createElement('div'))
4 const list = div.appendChild(document.createElement('ol'))
5
6 const dayNames = new Map()
7
8 for (let i = 0; i < 7; ++i) {
9   const d = Temporal.PlainDate.from({
10     year: Temporal.Now.plainDateISO().year,
11     month: 1,
12     day: i + 1,
13   })
14
15   dayNames.set(d.dayOfWeek, d.toLocaleString(LOCALE, { weekday: 'long' }))
16 }
17
18 for (const num of [...dayNames.keys()].sort((a, b) => a - b)) {
19   list.appendChild(Object.assign(
20     document.createElement('li'),
21     { textContent: dayNames.get(num) },
22   ))
23 }
```

Front End Languages



Front-end languages are programming languages specifically employed in the creation and development of the user interface (UI) and user experience (UX) elements of a website or web application. These languages contribute to shaping the structure, presentation, and interactive features that users directly engage with on the client side.

HTML (Hypertext Markup Language) forms the foundation, defining the structure and content of web pages. CSS (Cascading Style Sheets) complements HTML by styling and formatting the visual presentation of these elements. JavaScript, a dynamic scripting language, adds interactivity and behaviour to the front end, facilitating tasks such as form validation, animations, and real-time updates. Collectively, these front-end languages

empower developers to create visually appealing, responsive, and interactive web interfaces that enhance the overall user experience.

HTML, CSS, and JavaScript are foundational languages for front end development. HTML establishes the document structure, CSS styles it, and JavaScript adds behaviour. Additionally, TypeScript, a superset of JavaScript, introduces static typing for enhanced code quality.

Front End Frameworks



Front-End Frameworks and Libraries

➤ AngularJS

AngularJS is an open-source JavaScript front-end framework that is mostly used to create single-page web applications (SPAs). It is a framework that is always evolving and extending to give better Methods for constructing online apps. It transforms static HTML into dynamic HTML. It is an opensource project that may or may not be free. It uses Directives to expand HTML attributes and HTML to bind data.

➤ React.js

React is a JavaScript library for creating user interfaces that is declarative, fast, and customizable. ReactJS is an open-source, component-based front-end library that is only responsible for the application's display layer. Facebook is in charge of its upkeep. Furthermore, React Js simplifies front-end development.

➤ JQuery

JQuery is a JavaScript library that is open source that streamlines interactions between an HTML/CSS page, or more properly the page Object Model (DOM), and JavaScript. JQuery facilitates HTML document traversal and manipulation,

browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript programming, to name a few features.

➤ **SASS**

It is the most trustworthy, mature, and stable CSS extension language. It is used to easily enhance the functionality of an existing CSS of a site, including variables, inheritance, and nesting.

➤ **Flutter**

Flutter is a Google-managed open-source UI developer SDK. Dart is the programming language that powers it. From a single code base, it generates performant and aesthetically pleasing natively generated apps for mobile (iOS, Android), web, and desktop. The major selling point of flutter is that UI and native performance make flat development easier, more expressive, and more adaptable. Flutter 2 will be released in March 2021, and it will improve flutter to develop release apps for the web and the desktop.

Front End Designing Tools



• Visual Studio Code

(VS Code) Microsoft's VS Code is a source code editor that works on Linux, Windows, and macOS. It uses intelligence methods to aid in syntax highlighting, debugging, and code completion.

• Bootstrap

Bootstrap is a free and open-source CSS framework. Its purpose is to produce front-end web development that is responsive, simple, and mobile-first, and it includes design

templates for various interface components such as forms, typography, buttons, and navigation.

- Atom

Atom is a text and source code editor developed by GitHub that is free, open-source, and community-built. Atom is available for Linus, macOS, and Microsoft Windows and is written in Less, Coffee Script, JavaScript, and HTML. It is very versatile and adaptable, with built-in Git Control and multi-cursor plugins written in JavaScript, and is based on the Electron framework, which allows creating cross platform desktop programs.

- JQuery

JQuery, a JavaScript library licensed under the MIT license, was created to simplify HTML DOM tree navigation and manipulation. It is also used to handle events, Ajax, and CSS animation.

- Git

Git is the most used distributed version control system for tracking changes. It is open-source and written in Shell, C, Perl, and Tcl. Its design is based on the essential principles of simplicity, speed, complete distribution for non-linear processes, integrity, and support for numerous parallel branches.

Advantages of front-end

- User Experience Enhancement

Front-end development is concerned with designing a user-friendly and engaging interface in order to improve the overall user experience. Increased customer happiness and retention can be attributed to a well-designed front end.

- Visual Appeal

Front-end developers are in charge of a website's visual components, such as layout, colors, typography, and pictures. A visually beautiful design may entice customers, provide a favorable first impression, and improve brand perception.

- **Interactivity and Engagement**

Front-end development involves the implementation of interactive features such as forms, buttons, sliders, and animations. These elements help to boost user engagement while also making the website more dynamic and intriguing.

- **Cross-Browser Compatibility**

Front-end developers guarantee that a website works properly across several web browsers. This helps to reach a larger audience and offers a consistent experience independent of the visitor's browser or device.

- **Responsiveness**

Front-end development entails designing responsive designs that can be adapted to different screen sizes and devices. This guarantees that customers accessing the website from PCs, tablets, or smartphones get a consistent experience.

- **Fast Load Times**

Front-end developers reduce load times by optimizing assets such as pictures, scripts, and stylesheets. Faster load times improve the user experience and can improve search engine rankings.

- **Improved SEO Performance**

A well-optimized front end aids in search engine optimization (SEO). Search engines like websites that load quickly, are mobile friendly, and have clean code, all of which are factors impacted by front-end development.

Back End (Application Layer)



The **back end** of a web application, also known as the server-side, is the hidden infrastructure that powers the visible front end. It encompasses the server, database, and application logic, working together to process, manage, and deliver data in response to user requests. Unlike the front end, which focuses on user interaction, the back end operates behind the scenes, handling tasks such as data storage, retrieval, authentication, and business logic execution. Essentially, the back end is the "brain" of a web application, ensuring that everything functions seamlessly and reliably.

At its core, the back end manages server-side operations. This includes processing requests received from the client-side (front end), running scripts to perform specific tasks, and fetching or storing data in a database. For example, when a user logs into an account, the back end verifies the user credentials, retrieves relevant data from the database, and sends it back to the front end for display. It also handles complex business logic, such as calculating totals in e-commerce platforms or managing workflows in enterprise applications.

A typical back-end system consists of two main types of servers: an **application server** and a **database server**. The application server processes business logic, handles APIs, and manages communication between the client and the database. The database server, on the other hand, stores and organizes all the application's data, such as user information, product catalogs, and transaction records. Popular databases include relational ones like MySQL and PostgreSQL and NoSQL databases like MongoDB.

Back-end developers focus on building the components that drive the application. They write code to implement functionality, manage server operations, and ensure efficient data handling. This includes creating and optimizing APIs, integrating third-party services, managing user authentication and authorization, and building tools or libraries to support the application. They also ensure the system is secure, scalable, and capable of handling high volumes of user traffic.

Unlike static websites, which consist of fixed content that rarely changes, dynamic web applications rely heavily on back-end components to provide interactive and personalized experiences. For instance, when users search for products on an e-commerce website, the back end processes the query, retrieves matching results from the database, and sends them back to the front end. Without a robust back end, such functionality would not be possible. In conclusion, the back end is an essential part of any web application, enabling the functionality and data management that users rely on. By working in harmony with the front end, the back end ensures a smooth and efficient user experience, powering modern, dynamic, and data-driven applications. It serves as the foundation of the digital ecosystem, providing the necessary infrastructure for innovation and scalability.

Back End Web Technologies



Back-end web technologies encompass the tools, languages, and frameworks used to build and manage the serverside components of a web application. These technologies are crucial for handling data processing, managing databases, and executing the application's business logic. They facilitate communication between the front end and databases, ensuring efficient data retrieval and storage. Additionally, back-end technologies provide the necessary infrastructure for deploying and scaling web applications, significantly contributing to the overall reliability and efficiency of the system.

Server-Side Scripting Languages:

1. PHP:

PHP stands for hypertext preprocessor. This is a scripting language, which means that the code you write in it is read by a computer rather than a compiler, where it would be translated into machine-readable code. PHP is used by developers for server-side functionality, and it

facilitates deploying, or submitting and confirming, code using deployment tools and capabilities. PHP is beneficial in a variety of development domains because PHP code can be embedded in HTML, which is a front-end tool for developing online apps and websites.

```
<?php
$res = mysql_query("SELECT id, name FROM table");
while ($row = mysql_fetch_array($res)) {
    echo "id: ".$row["id"]."<br>\n";
    echo "name: ".$row["name"]."<br>\n";
}
?>
```

2. Python:

Python is an object-oriented programming language that focuses on manipulating data-contained objects. This is a basic language with a simple syntax that is easy to comprehend and debug. Web developers using Python may utilize the open-source framework Django to construct scalable, readily updated, or side-graded software for the web.

3. Ruby:

Ruby is a computer language that may be used with Ruby on Rails (RoR), a Ruby-specific framework that allows you to build and perform tasks with less code. Frameworks are frequently excellent tools for software architects, who use the framework's code libraries, built-in tools, and specific methods to ease the development process.

```
class CreateRelationships < ActiveRecord::Migration[5.0]
  def change
    create_table :relationships do |t|
      t.integer :follower_id
      t.integer :followed_id
      t.timestamps
    end
    add_index :relationships, :follower_id
    add_index :relationships, :followed_id
    add_index :relationships, [:follower_id, :followed_id], unique: true
  end
end
```

Back-End Frameworks



1. Laravel (PHP):

- **Purpose:** Laravel is a PHP framework that provides an elegant syntax and a wide range of features for web application development. It emphasizes ease of use and follows the MVC (Model-View-Controller) architectural pattern.
- **Role:** Laravel simplifies tasks such as routing, authentication, and database management. It enhances developer productivity by offering built-in tools and libraries.

2. Django (Python):

- **Purpose:** Django is a high-level Python web framework that promotes rapid development and clean, pragmatic design. It adheres to the MTV (Model-Template-View) architecture.
- **Role:** Django streamlines tasks like database interaction, authentication, and URL routing. It includes an admin interface for managing application data and ensures security with built-in features.

3. Express.js (Node.js):

- **Purpose:** Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features for building web applications and APIs.

- **Role:** Express.js simplifies server-side development by handling routing, middleware, and HTTP requests. It is often used in conjunction with Node.js to create scalable server-side applications.

Databases

1. MySQL

SQL (structured query language) is a computer language that employs queries, or data requests, to obtain information from databases. SQL may be used to get useful server-side information from databases and then utilize that information to conduct actions that allow the client side of an application or software to function effectively. This programming language is particularly handy for updating internal corporate databases or retrieving data for clients

2. MongoDB

- **Purpose:** MongoDB is a NoSQL database that uses a document-oriented data model. It stores data in JSON-like format, making it suitable for handling unstructured data.
- **Role:** MongoDB provides flexibility in data storage and retrieval, supporting scalability and performance for modern web applications.

3. PostgreSQL

- **Purpose:** PostgreSQL is an open-source relational database known for its advanced features and standards compliance.
- **Role:** PostgreSQL supports complex queries, transactions, and data integrity, making it a powerful choice for applications requiring robust data management.

Back-End Designing Tools

• MongoDB

MongoDB is a popular free and open-source document-oriented database among web developers. This NoSQL database stores and retrieves data using JSON-like documents with optional schemas, providing much-needed scalability and flexibility. Ad-hoc queries, Indexing, Replication, Load Balancing, File Storage, Aggregation, Transactions, and more functionalities are available in MongoDB. Furthermore, the security and data recovery components of this database are far superior than those of conventional database management systems. Aside from that, Mongo DB is compatible with a wide range of mainstream programming languages, including C/C++, Java, PHP, Python, Ruby, and others.

• MySQL

MySQL is another popular open-source relational database management system for web-based applications. It is a high-performance database with improved scalability, usability, and stability.

Furthermore, MySQL supports cross-platform compatibility, robust indexing, SSL for secure connections, powerful data encryption and accuracy, built-in replication support, and a variety of additional capabilities. Meanwhile, MySQL can run on a variety of different operating systems and is compatible with a wide range of popular programming languages such as PHP, Java, and others. Let us also mention that this particular foundation, MySQL, is utilized by several well-known websites like as Flickr, Twitter, Facebook, Drupal, Joomla, and many more.

• Apache

The Apache, often known as the Apache HTTP Server, is a cross-platform open-source web server created by the Apache Software Foundation. It is one of the most popular web servers, as evidenced by the fact that Apache powers about half of all websites worldwide. Apache includes various useful features such as loadable dynamic modules, multiple request processing modes, CGI support, user and session tracking, real-time status displays, XML & FTP support, and many more. Furthermore, the Apache Web Server is very interoperable with practically all operating systems, including Linux, macOS, and Windows.

• Docker

Docker is, without a doubt, the most recommended tool for all developers. It is a containerization platform that allows you to easily design, deploy, and execute applications

using containers. Docker provides you with a consistent and isolated environment and is in charge of isolating and segregating your apps and resources so that each container may access all essential resources without interfering with or relying on another container. Other significant benefits of utilizing Docker include rapid application deployment, scalability and flexibility, an in-built version control system, security and cost-effectiveness, and many more

Advantages of back-end

- Data Management: Backend developers are in charge of data storage and retrieval. They create and administer databases to ensure effective data storage, retrieval, and modification. This is critical for programs that handle massive volumes of data.
- Server-Side Logic: Backend development entails creating the server-side logic that governs the functioning of the program. This comprises business logic, authentication, authorization, and other server-side activities.
- Security: Backend development prioritizes security. Backend developers put in place safeguards to secure sensitive data, restrict unwanted access, and assure the application's overall security. This covers systems for encryption, authentication, and authorization.
- Scalability: Backend development is critical to ensure that programs can manage rising demands and scale efficiently. To support rising user bases, backend developers create scalable systems and integrate services such as load balancing.
- API Creation: APIs (Program Programming Interfaces) are created by backend developers to allow various portions of a program to communicate with one another. APIs promote communication between the frontend and backend by allowing the integration of multiple services.
- Third-Party Service Integration: Backend developers work with third-party services such as payment gateways, social networking platforms, and external APIs to connect apps. This integration improves the application's functioning and adds new features.
- Database Administration: Backend development includes selecting and managing databases that best meet the demands of the application. This covers the creation of database schemas, the optimization of queries, and the maintenance of data consistency.

- Server Configuration and Administration: Server configuration, deployment, and management are handled by backend developers. They make certain that the server environment is optimized for speed, security, and scalability.
- Performance Enhancement: Backend developers improve server speed by tweaking code, database queries, and server parameters. This results in faster response times and better overall system performance.
- Dealing with Business Logic: Backend development is in charge of putting the business logic that determines how the program works into action. This comprises the processing of user inputs, the execution of sophisticated algorithms, and the management of workflow processes

Relationship between Frontend and Backend

The collaboration between front-end and back-end technologies is crucial for developing fully functional and dynamic web applications. Each plays a distinct but complementary role in creating a seamless user experience.

Front-End Development:

Front-end development is concerned with the user interface (UI) and user experience (UX) of a website. It includes everything users see and interact with directly in their web browsers. This involves:

- **HTML (Hypertext Mark-up Language):** Provides the structural foundation of web pages, defining the content and layout.
- **CSS (Cascading Style Sheets):** Adds styling and visual appeal, controlling the presentation and design elements.
- **JavaScript:** Introduces interactivity and dynamic features, enhancing user engagement through animations, form validations, and real-time updates.

Front-end development focuses on creating visually appealing, responsive, and user-friendly interfaces. Developers use these technologies to ensure that websites are accessible across various devices and screen sizes. Effective frontend development engages users and creates a positive impression of the website.

Back-End Development:

Back-end development handles the server-side operations, data processing, and business logic of a web application. It manages:

- **Server-Side Logic:** Executes the application's core functionality, processes user requests, and performs calculations.
- **Database Management:** Handles data storage, retrieval, and manipulation. Common databases include MySQL, PostGre SQL, and Mongo DB.
- **APIs (Application Programming Interfaces):** Facilitate communication between the front-end and backend, enabling data exchange and integration with external services.

Back-end development is critical for ensuring that a website operates efficiently and securely. It involves working with server-side languages like PHP, Python, Ruby, and frameworks such as Django, Ruby on Rails, and Laravel to build robust and scalable applications.

Integration and Collaboration:

The front-end and back-end must work in harmony to create a cohesive web application. Here's how their collaboration unfolds:

1. **Data Exchange:** Front-end components send requests to the back-end via APIs to fetch or submit data. The back-end processes these requests, interacts with the database, and returns the necessary information to the front end.
2. **Seamless User Experience:** The front end presents the data and functionalities provided by the back end in a user-friendly manner. For instance, a user may fill out a form (front end), which is then processed and stored in the database by the back end.
3. **Error Handling:** Both front-end and back-end developers need to coordinate to handle errors effectively. For example, if the back end encounters an issue, it should return meaningful error messages that the front end can display to users.
4. **Security:** Both sides must work together to ensure data security. While the back end handles authentication and authorization, the front end must ensure that sensitive information is managed securely.

Front-End vs. Back-End:

- **Front-End:** Focuses on the visual elements and user interactions. It aims to create an engaging and intuitive interface.
- **Back-End:** Deals with server-side operations, data management, and application logic. It ensures the website's functionality and performance.

Both front-end and back-end development are integral to building a successful web application. The front end captures user attention and provides an interactive experience, while the back end ensures the application's functionality and reliability. Their effective collaboration is essential for delivering a dynamic, efficient, and user-friendly website.

The front-end and back-end of a web application are two distinct components that work together to deliver a complete and functional user experience. They are closely related to the presentation and application layers of a web application architecture. The front end and back end of a software application are two distinct components that work together to deliver a seamless user experience. The front end, also known as the client side, is the user interface that users interact with directly. It includes elements such as the graphical user interface (GUI), buttons, forms, and other components that users see and interact with. Front-end technologies often involve HTML, CSS, and JavaScript, and the development focuses on creating a visually appealing and user-friendly interface. Front-end development is concerned with ensuring that the application looks good and functions well from the user's perspective. The back end, or server side, on the other hand, is in charge of maintaining and processing data as well as executing different duties behind the scenes. Server-side programming, databases, and server settings are all involved. The back end ensures that the program runs smoothly by handling duties like data storage, retrieval, and user request processing. It is in charge of the application's logic and business rules, as well as database management and server-side activities. APIs (Application Programming Interfaces) connect the back end and front end, allowing data interchange and coordination between the user interface and the server. A client-server architecture is commonly used to define the relationship between the front end and the back end. The front end asks information or actions from the back end, and the back end processes these requests, obtains or

manipulates data, and returns the results to the front end for user presentation. This separation of responsibilities improves software development scalability, maintainability, and cooperation. It also allows for specialization, with developers focusing on either front-end or back-end technologies, enabling effective cooperation in the development of sophisticated and feature-rich applications

Presentation Layer and Application layer

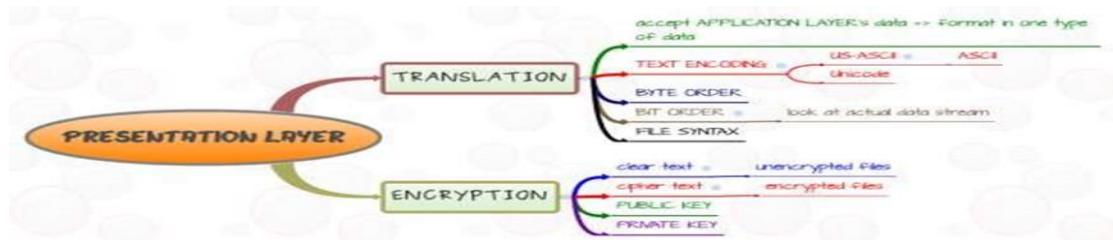
- Presentation Layer**

The presentation layer is accessed by users through a browser and comprises of user interface components and UI process components that allow system interaction. It is built using three main technologies: HTML, CSS, and JavaScript. While HTML specifies what your website will include, CSS dictates how it will look. JavaScript and its frameworks make your website interactive, allowing it to respond to user activities. To make the information on the page dynamic, developers employ JavaScript frameworks such as Angular and React. The presentation layer ensures users interact with a visually appealing and responsive interface. Front-end technologies create and manage the visual components that users see and interact with. The application layer processes data and handles logic required to meet user requests from the front-end. Back-end technologies are responsible for ensuring secure and efficient communication between the server, database, and user interface

- Application Layer**

The OSI and TCP/IP layered models have the application layer at the top. This layer is included in both tiered Models because of the significance of connecting with users and user applications. Apps that are part of a communication system use this layer. The user may or may not engage with the applications directly. The application layer initiates and reflects the real communication. Because it resides at the top of the stack, it does not serve any further levels. The Application layer and all levels below it use Transport to connect with or transfer data to a remote host. The Application Layer includes various protocols that customers frequently require. HTTP (Hyper Text Transfer

Protocol), the World Wide Web's basis, is one of the most widely used application protocols. When a browser requests a web page, it sends its name to the server over HTTP. The server then returns the page to the client



Web Application Architecture.

Web application architecture serves as a foundational framework that describes the establishment of connections between clients and servers. It explains how the diverse components within an application seamlessly communicate with one another. Regardless of the application's size or complexity, they adhere to the same fundamental principles, albeit with variations in specific details.

In more technical terms, when a user initiates a request on a website, multiple facets of the application come into play. This includes components such as user interfaces, middleware systems, databases, servers, and the web browser itself. Web application architecture plays the role of an essential framework, organizing and upholding the intricate interplay among these components. It ensures that the interaction between these elements remains organized and effective.

How Web Application Architecture Works

All web applications operate on both the client-side and the server-side. When a user makes a request, there are mainly two programs running on both sides:

- **Client-Side Code:** Runs in the browser based on the user's inputs. It is responsible for rendering the user interface and providing interactive elements.
- **Server-Side Code:** Responds to HTTP requests. It handles data processing, generates web pages, and manages backend functionalities.

When developing a web application, developers make critical decisions regarding the responsibilities of server-side and client-side code. They define how these two components

will interact. Server-side code can be written in a variety of languages, including Python, JavaScript (Node.js), C#, PHP, Ruby on Rails, and more. Any code that can respond to HTTP requests can potentially run on the server. Server-side code plays a key role in generating the web pages that users request and managing various types of data, such as user profiles, tweets, and pages. Notably, server-side code remains hidden from end-users, except in rare malfunction scenarios.

Client-side development, on the other hand, leverages a combination of HTML, CSS, and JavaScript. This code is executed by the user's web browser and is both visible and editable by the user. Client-side code communicates with the server exclusively through HTTP requests and lacks the capability to access server files directly.

In summary, web developers must make thoughtful decisions about how server-side and client-side code work together to deliver a seamless and interactive web experience. Server-side code handles data processing and page generation, remaining hidden from users, while client-side code is executed in the user's browser, offering visibility and interactivity. These two components collaborate through HTTP requests to create the web applications we use daily.

Web Application Three-Tier Architecture Layers

Web applications use a specific developing system to separate their main functions into layers and tiers. This helps developers to easily upgrade and replace each layer independently without harming other layers. This architecture is called Multi-Tier or Three-Tier Architecture.

1. Presentation Layer

This layer is accessible to the client via a browser and includes user interface components and UI process components. As discussed, these UI components are built with HTML, CSS, and JavaScript (and its frameworks or libraries), where each of them plays a different role in building the user interface.

- **Focus:** User interaction, data input and display of information.

- **Role:** This is the UI/UX that users see and interact. It gathers input, validates data, and displays the results returned from the other layers.
- **Technologies/Tools:** HTML, CSS, JavaScript, front-end frameworks (React, Angular, etc.), Mobile Development technologies (Swift, Kotlin, etc.).

2. Business Layer

Also referred to as Business Logic, Domain Logic, or Application Layer, this layer accepts the user's request from the browser, processes it, and regulates the routes through which the data will be accessed. The entire workflow is encoded in this layer. For instance, when booking a hotel on a website, a traveller will go through a sequence of events, and the whole workflow will be managed by the business logic.

- **Focus:** Core functionality and business logic.
- **Role:** This tier is the brain of the application. It processes data, implements business rules and logic, further validates input, and coordinates interactions between the presentation and data layers.
- **Technologies:**
 - Programming Languages: PHP, ASP.NET, Java, Python, JavaScript, C#, Ruby, etc.
 - Web Frameworks: Spring, Django, Laravel, Ruby on Rails, etc.
 - App Server/Web Server

3. Persistence Layer

Also known as the Storage or Data Access Layer, this layer handles all data calls and provides access to the persistent storage of an application. The business layer is closely attached to the persistence layer, so the logic knows which database to talk to, and the

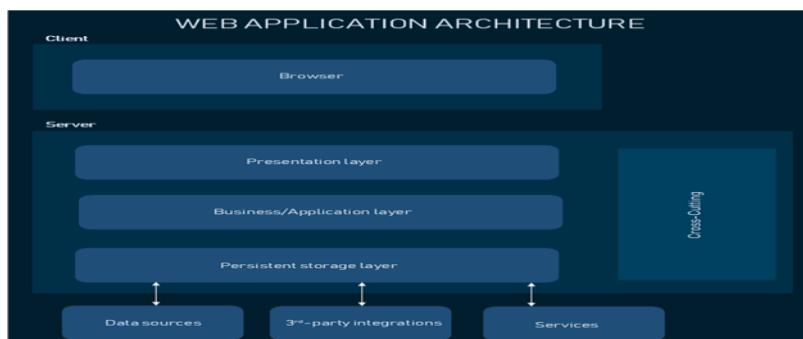
process of retrieving data becomes more optimized. A server and a database management system software exist in data storage infrastructure, which is used to communicate with the database itself, applications, and user interfaces to retrieve and parse data. Data can be stored in hardware servers or in the cloud.

The presentation layer is built using the trinity of front-end development technologies—HTML, CSS, and JavaScript. Regardless of the type of architecture, the presentation layer always incorporates these front-end technologies. The backend of the web app consists of the application layer and the persistence layer. The application layer refers to the workflow and code, while the persistence layer pertains to the database and storage. The combination of these two layers constitutes the back end. The application layer includes all the coding of the workflow and acts as the brain of the web application, handling all logical functionalities. The persistence layer assists in retrieving and storing data based on the instructions provided.

- **Focus:** Persistent storage (i.e. database) and management of data.
- **Role:** This layer reliably stores the application's data and handles all database queries. It protects data integrity and ensures consistency.
- **Technologies:**
 - Database servers and database management systems : (Oracle, MySQL, PostgreSQL, Microsoft SQL Server) or NoSQL systems (MongoDB, Cassandra).

In conclusion, the front end of the web application relates to the presentation layer, and the back end corresponds to the application and persistence layers. The stability of both these layers is crucial for the success of the web application.

Relation of Frontend and Backend to Presentation and Application Layers



In a web application architecture, the connection between frontend and backend development is Analogous to the separation of display and application layers. The display layer is mostly represented by the frontend, often known as client-side development. It is concerned with what people directly engage with—the user interface, visual design, and user experience. Frontend developers employ HTML, CSS, and JavaScript to create a visually appealing and responsive interface that consumers interact with directly in their browsers. This layer is responsible for delivering material and interactions to end users, hence defining their experience with the application

The backend, or server-side development, on the other hand, is in sync with the application layer. It includes the server, database, and server-side logic that is in charge of processing requests, maintaining data, and performing the application's basic capabilities. Backend developers employ programming languages such as Python, Java, or Ruby to administer server-side activities and store data. The application layer is the workhorse behind the scenes, processing data and performing application functions without the user seeing the precise intricacies.

In essence, the frontend and backend work together to produce a seamless online application. The frontend is responsible for the user interface and experience (presentation layer), whereas the backend is in charge of server-side logic and data processing (application layer). These layers' smooth integration results in a wellrounded, functioning, and user-friendly online application. The separation of duties allows for specialization at each layer, promoting the effective creation and maintenance of complex online solutions. (palaksinghal9903, 2024)

Analysis of Tools and Techniques for Designing and Developing a Custom-Built Website

1. Tools for Designing a Custom-Built Website

1.1 Wire framing and Prototyping Tools

Figma: Figma is a cloud-based design tool that facilitates **collaborative wire framing** and **prototyping**, enabling designers to create visually accurate mock-ups. Its main advantage lies

in its ability to **easily share** designs and collect **real-time feedback** from multiple stakeholders, making it ideal for team-based work. The tool also supports **cross-platform** use, ensuring that team members can access and edit designs on various devices. A use case for Velvet Vogue's website could be **visualizing the homepage layout** and **navigation flow**, allowing designers and stakeholders to interact with the design before development.

Adobe XD: Adobe XD provides **advanced tools for designing user interfaces (UI)** and creating **interactive prototypes**, helping designers focus on how the website will function visually and interactively. Its integration with **Adobe Creative Suite** enhances the design process by allowing seamless collaboration with other design tools like Photoshop and Illustrator. This feature would support Velvet Vogue in creating high-fidelity visual designs for the website while maintaining a cohesive design workflow.

Balsamiq: Balsamiq is a tool focused on **low-fidelity wire framing**, ideal for quickly sketching out the website's basic structure and layout without getting bogged down by detailed visual elements. Its simplicity makes it useful in the early stages of design, providing an opportunity to focus on the functionality and **layout of the site**. For Velvet Vogue, Balsamiq could help to outline **the basic structure** of key pages, allowing quick iteration and feedback before progressing to more detailed design work.

1.2 Graphic Design Tools

Adobe Photoshop: Adobe Photoshop is a **professional-grade design tool** ideal for **creating high-quality visuals**, including **product images**, **banners**, and other detailed graphics for the website. Its **advanced editing capabilities** allow for **precise control** over image manipulation, such as color correction, retouching, and compositing, making it suitable for high-end visuals. For Velvet Vogue, Photoshop would be useful in creating **detailed product images** and **promotional banners** that need to maintain a high level of visual quality.

Canva: Canva is a **user-friendly design tool** that provides **pre-designed templates** to create a wide range of web assets like **banners**, **social media posts**, and **promotional images**. Its simplicity makes it accessible even to **non-designers**, allowing quick asset creation with

minimal design expertise. The major advantage of Canva is its **easy-to-use interface**, enabling Velvet Vogue's marketing team or non-technical staff to **create visual content** for the website, such as **seasonal banners** or **promotional graphics**, without needing advanced design skills.

2. Techniques for Designing a Website

2.1 Responsive Design

Definition: Responsive web design involves designing the website so that it **adapts seamlessly** to different **screen sizes** and **devices** using **CSS media queries**. These queries adjust the layout and content depending on the device's screen width, ensuring that the website is visually appealing and functional on any device, from large desktop monitors to small mobile screens.

Advantages: The main advantage of responsive design is that it **ensures a consistent user experience** across various devices such as **desktops**, **tablets**, and **mobile devices**. This means that users can have the same smooth, easy-to-navigate experience, no matter the device they use. This consistency boosts **user engagement** and **retention** while improving **SEO rankings** since search engines prioritize mobile-friendly websites.

Tools: Frameworks like **Bootstrap** help simplify the implementation of responsive design. These frameworks provide pre-built **grid systems** and **responsive components** that automatically adjust to different screen sizes, reducing development time and ensuring that the website works well across all devices.

2.2 Accessibility and Inclusivity

Definition: Ensuring the website adheres to **WCAG (Web Content Accessibility Guidelines)** means designing and developing the website in a way that makes it **accessible** to all users, including those with **disabilities**. This includes ensuring the website is usable by individuals with visual, auditory, cognitive, or physical impairments.

Advantages: Following WCAG improves **usability** for a wider audience, helping to **serve users with disabilities** effectively. It also helps in achieving **legal compliance**, as many jurisdictions require websites to be accessible to people with disabilities (e.g., **ADA**)

compliance in the U.S.). Additionally, it enhances **SEO** and can improve user satisfaction by making the website more inclusive.

Techniques:

- **ARIA (Accessible Rich Internet Applications)** attributes are used to provide additional information about the functionality of elements on the page, making it easier for screen readers to interpret dynamic content or complex elements.
- **Alt text for images** ensures that all visual content is described for users who cannot see it, while maintaining **high contrast** between **text** and **background** helps users with visual impairments read content more easily. These techniques ensure that all users, regardless of ability, can fully engage with the website.

3. Tools for Developing a Custom-Built Website

3.1 Front-End Development Tools

Visual Studio Code (VS Code):

- **Definition:** VS Code is a versatile, lightweight code editor that supports various programming languages and has built-in debugging features. It includes extensions for syntax highlighting, live server previews, and version control integration.
- **Advantages:** VS Code is known for its **speed**, **ease of use**, and **extensibility**. It supports numerous languages such as HTML, CSS, and JavaScript, making it an ideal tool for web development. It also offers a wide range of **extensions** to streamline development workflows.

React.js:

- **Definition:** React.js is a JavaScript library used to build dynamic, reusable components for user interfaces. It facilitates the creation of interactive and high-performance web applications.
- **Advantages:** React enables **efficient rendering** of components using a virtual DOM, leading to faster updates and improved performance. Its **component-based**

architecture makes it scalable and easy to maintain, which is particularly useful for building dynamic features like **product filters** on e-commerce sites.

Bootstrap:

- **Definition:** Bootstrap is a front-end framework that provides a **predefined grid system** and a collection of UI components for responsive web design.
- **Advantages:** Bootstrap's predefined grid and **components** (like navigation bars, buttons, and modals) help developers save time by eliminating the need to create common UI elements from scratch. It ensures **responsive design**, making it easier to create websites that work across different screen sizes.

3.2 Back-End Development Tools

- **Node.js:**
 - A runtime environment for server-side JavaScript development.
 - **Advantages:** Handles concurrent requests efficiently, making it ideal for e-commerce applications.
- **Django:**
 - A Python-based framework with built-in tools for secure and rapid web development.
 - **Advantages:** Scalability and robust security features.
- **MongoDB:**
 - A NoSQL database for storing unstructured and semi-structured data.
 - **Advantages:** Flexible schema design suitable for product catalogs and user data.

3.3 Content Management Systems (CMS)

WordPress:

- **Definition:** WordPress is a popular Content Management System (CMS) that simplifies website creation and management. It offers a wide range of plugins, such as **WooCommerce**, to extend functionality, especially for e-commerce.

- **Advantages:** WordPress is known for its **user-friendliness**, allowing users without extensive technical expertise to manage content easily. It offers a vast library of themes and plugins, making it suitable for various website types.
- **Limitations:** While WordPress is flexible, it may not offer the **deep customization** and scalability required for highly specialized websites. Its reliance on third-party plugins can sometimes lead to compatibility or performance issues.

Custom CMS:

- **Definition:** A **Custom CMS** is built using technologies like **Node.js** and **MongoDB** to create a content management solution tailored to the specific needs of a business or website.
- **Advantages:** The key benefit of a custom CMS is that it provides **complete control** over features and functionality. Developers can design the system to meet unique requirements and optimize performance without being constrained by third-party plugins.
- **Limitations:** The primary drawback is that **development** can be more time-consuming and requires a high level of technical expertise. Additionally, it may be more expensive and complex to maintain than an out-of-the-box solution like WordPress.

4. Techniques for Developing a Website

4.1 Modular Development

- **Definition:** Breaking the website into reusable components (e.g., navigation bar, product card).
- **Advantages:** Simplifies updates and ensures code consistency.
- **Tools:** React or Angular frameworks.

4.2 Version Control

- **Definition:** Using tools like Git to track changes in code and collaborate with team members.
- **Advantages:** Enables rollback to previous versions in case of errors.

4.3 API Integration

- **Definition:** Connecting external services (e.g., payment gateways, analytics) using RESTful APIs.
- **Advantages:** Extends website functionality without reinventing the wheel.

5. Testing and Deployment Tools

5.1 Testing Tools

- **Selenium:**
 - Automates functional testing for web applications.
 - **Advantages:** Ensures features like login and checkout work seamlessly.
- **Google Lighthouse:**
 - Analyzes performance, SEO, and accessibility.
 - **Advantages:** Provides actionable insights for improvement.

5.2 Deployment Tools

- **GitHub:**
 - Manages the source code repository and integrates with CI/CD pipelines.
 - **Advantages:** Simplifies team collaboration and deployment.
- **Heroku/AWS:**
 - Platforms for deploying and hosting web applications.
 - **Advantages:** Scalability and reliability.

Tools and techniques used

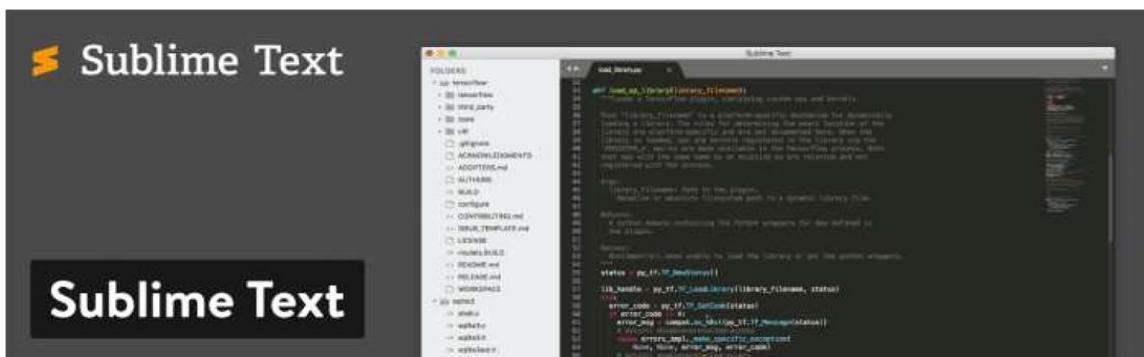
The tools and technologies selected for creating my website offer a comprehensive suite for effective web development. After evaluating various options, I chose the following languages and tools for my project:

Text Editors

A text editor is a computer software that allows users to create, alter, or edit plain text files (txt files). They're frequently used to write sophisticated code for websites, read, produce, and update source code, and create text files.

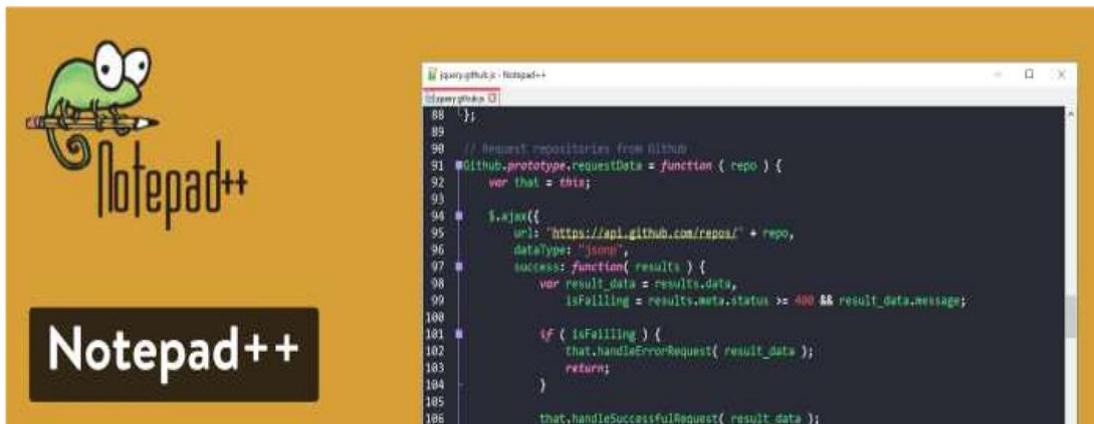
1. Sublime Text

It provides a free version for testing, but all ongoing users must pay \$99 to keep it operational. While \$99 may seem costly for a text editor, keep in mind that licenses are per-user rather than per-machine, so you may use Sublime Text on as many computers and operating systems as you like with your license. In terms of features, Sublime Text has an edge in that it is incredibly lightweight (low resource utilization), while yet retaining some of the more complex capabilities you would expect from a top text editor. The biggest advantage is that Sublime Text provides shortcuts and search features for quickly discovering specific functions and making changes to several lines at once. It merely takes a few seconds to jump to certain symbols or sentences.



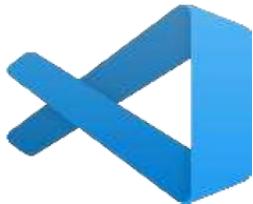
2. Notepad++

Notepad++, the market's most popular sophisticated text editor, comes in a small package with no fees and extensive editing tools. It is distributed under a General Public License, which means that any developers and content producers can use the text editor immediately after downloading it. Notepad++ is a text editor that runs on Microsoft Windows that attempts to utilize less computer resources than the ordinary text editor. One feature that distinguishes Notepad++ is that it has already been translated into over 80 languages, giving access to individuals all around the world. Furthermore, if your language does not appear on the list of translations, you may translate Notepad++ into it.



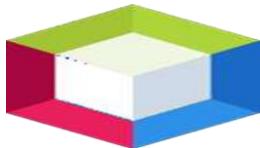
Web Development IDE

1. Visual Studio Code



TMS Outsource named Visual Studio Code the best JavaScript IDE for Windows, Mac, and Linux, while the State of JS named it the most popular in a 2018 poll. Visual Studio Code is a robust source code editor that comes with a variety of JavaScript development tools. The IDE includes support for JavaScript, TypeScript, and Node.js. It also has a plethora of extensions for other languages (including C++, C#, Python, and PHP). Visual Studio Code, created by Microsoft, is ideal for beginner programmers since it covers everything from HTML elements to grammar and error handling

2. Net Beans



Net Beans is a free, user-friendly IDE that supports JavaScript, HTML, PHP, C, and C++. It supports a variety of programming languages and is also accessible in English, Brazilian Portuguese, Japanese, Russian, and Simplified Chinese. Net Beans is not recommended for beginners since it might be difficult to set up.

Web server solution package

1. XAMPP



XAMPP is a free and open-source cross-platform web server solution stack package built by Apache Friends that includes the Apache HTTP Server, the Maria DB database, and interpreters for PHP and Perl scripts. This web XAMPP server package is compatible with Windows, macOS, Linux, and Solaris operating systems, making it simple to set up a local web server hosting environment. When it comes to setting up local web servers, many users rely on the XAMPP software package. It has a wide range of uses in web development and testing.

2. WampServer



WampServer is a Microsoft Windows solution stack designed by Romain Bourdon that includes the Apache web server, OpenSSL for SSL support, the MySQL database, and the PHP programming language. WampServer is a web development environment for Windows. It enables the development of online applications using Apache2, PHP, and a MySQL database. In addition, PhpMyAdmin makes it simple to manage your databases.

Justification of the Technologies, Management Services, Tools, and Software Chosen to Realize a Custom-Built Website

1. Technologies

1.1 Front-End Technologies

The **HTML5**, **CSS3**, and **JavaScript** technologies form the backbone of Velvet Vogue's website, providing a **responsive** and **interactive** user interface. **HTML5** ensures a clean and semantic structure, making the website more accessible and SEO-friendly. **CSS3** allows for advanced styling, including animations and responsive layouts, ensuring the website is visually appealing across devices. **JavaScript** enhances user interaction by adding dynamic features like **product filters** and an **interactive shopping cart**, improving the overall shopping experience. For instance, Velvet Vogue's **product pages** are styled with **CSS3**, and **JavaScript** enables live updates of inventory, ensuring real-time accuracy for users.

The use of **React.js** was chosen for its **component-based architecture**, which makes it easier to develop and manage reusable UI elements. This approach contributes to better **performance** and **scalability**, as it allows components like **sliders** and **product grids** to be reused throughout the site without redundancy. For example, the **product categories** and **filters** are dynamic, providing a more streamlined and intuitive **navigation** experience for users. By leveraging **React.js**, Velvet Vogue ensures a more efficient and responsive user interface, enhancing the overall **user experience** on the website.

1.2 Back-End Technologies

Node.js is utilized for its **fast, event-driven runtime**, making it ideal for building **scalable server-side applications**. It handles **concurrent user requests efficiently**, which is crucial for ensuring **fast checkout processing** and smooth user experience during peak traffic times. For example, during high-traffic periods, **Node.js** efficiently manages **API calls for product availability**, ensuring that customers can access up-to-date inventory information without delays.

MongoDB, a **NoSQL database**, was chosen for its **flexible schema design**, making it an excellent fit for e-commerce applications. Its schema flexibility allows it to handle varying data structures such as **product catalogs**, **user profiles**, and **order histories**—all of which may evolve over time. For instance, **MongoDB** stores essential data like **product details**, **customer accounts**, and **purchase histories**, making it easier to scale and adapt as the business grows.

RESTful APIs were integrated to facilitate smooth communication between the **front-end** and **back-end** systems. They transmit data in the lightweight **JSON format**, which is fast and easy to process. For example, the **RESTful APIs** enable **real-time updates** of **cart items** and **stock levels**, ensuring that customers have the most current information, particularly when adding products to the cart or checking availability during shopping. This ensures seamless user interaction and accurate, up-to-date product details.

2. Management Services

2.1 Hosting

Amazon Web Services (AWS) was chosen for its **reliable cloud hosting** and **scalability**, ensuring that Velvet Vogue's website can handle high traffic, especially during sales or product launches. AWS's infrastructure is designed to provide **uptime reliability**, even under heavy loads, and features like **load balancing** and **auto-scaling** ensure the website maintains optimal performance. For example, during events like **Black Friday sales**, AWS ensures that the website can accommodate traffic spikes, providing a seamless shopping experience without delays or downtime.

A **Content Delivery Network (CDN)**, such as **Cloudflare**, was integrated to enhance the website's **performance**. By caching assets on servers located closer to users, the CDN **reduces load times**, improving the overall **user experience**. For instance, high-quality **product images** are delivered faster to customers around the world, ensuring that users can browse and shop without experiencing delays, even during periods of high demand. This not only improves **website speed** but also boosts customer satisfaction by providing a smoother browsing experience.

2.2 Security

SSL/TLS Certificates are implemented to **encrypt communication** between users and the website, ensuring the **security of sensitive data** during interactions such as checkout and login. This encryption prevents unauthorized access to customer data during transmission, fostering

trust in the platform. For example, **credit card details** entered during checkout are securely encrypted, ensuring customers feel safe making transactions on Velvet Vogue's website.

Regular Backups are crucial for **protecting against data loss**, whether due to **server crashes** or **cyberattacks**. By regularly backing up website data, the risk of permanent data loss is mitigated. For instance, if a server fails or there's a security breach, **customer orders** and **product data** can be quickly restored from backups, ensuring business continuity and minimal disruption to the user experience.

3. Tools

3.1 Development Tools

Visual Studio Code (VS Code) was chosen as the primary **code editor** for its **lightweight** and **extensible** features, making it an ideal tool for efficient development workflows. It includes built-in **debugging** tools and **Git integration**, allowing developers to test and refine both the front-end and back-end code seamlessly. For example, developers use VS Code to write HTML, CSS, and JavaScript for the front-end, as well as PHP and JavaScript (Node.js) for back-end functionalities, with real-time debugging support.

Git and **GitHub** are essential for **version control** and **team collaboration**, enabling the development team to manage changes in the codebase effectively. Git tracks every modification made to the code, ensuring that any errors or conflicts can be easily resolved, and enabling developers to **rollback** to previous versions if needed. For example, GitHub is used to manage pull requests, code reviews, and merges, ensuring that the team works efficiently without overwriting each other's changes.

3.2 Testing Tools

Selenium is used to **automate testing** of key website functionalities, ensuring that critical features like **login**, **checkout**, and **product search** work seamlessly across different scenarios. It simulates user interactions and tests how the website performs under various conditions. For example, Selenium can automatically test scenarios such as **adding and removing items from the cart**, ensuring that these actions work smoothly and correctly without manual intervention.

Google Lighthouse is employed to **analyze the website's performance, accessibility, and SEO**, providing detailed reports and **actionable insights** for optimization. It evaluates aspects such as **page load speed, mobile responsiveness, and SEO best practices**, helping the development team make improvements. For instance, **optimizing page load times** and improving mobile layout adjustments are common tasks that Google Lighthouse helps identify, ensuring the website offers a smooth user experience and ranks well in search engines.

4. Software

4.1 Web Server Software

Nginx is chosen for its ability to **handle a large volume of simultaneous requests**, making it an ideal choice for **high-traffic e-commerce sites** like Velvet Vogue. As a web server and reverse proxy server, Nginx is optimized for performance and can efficiently serve static content while distributing requests across multiple servers.

For example, during **promotional campaigns or sales events**, such as Black Friday, when traffic spikes dramatically, Nginx manages the **increased number of requests** for product pages, ensuring that users experience fast load times and no downtime despite the heavy load. Its ability to efficiently manage these high volumes helps maintain a smooth, uninterrupted shopping experience for customers.

4.2 Content Management System (CMS)

Custom CMS (Node.js + MongoDB) is used to create a **tailored content management system** that meets Velvet Vogue's specific needs. Using **Node.js** as the backend and **MongoDB** for the database ensures flexibility and scalability, allowing the system to efficiently handle the dynamic nature of e-commerce data such as **product categories, inventory, and orders**.

For example, the **admin dashboard** built on this custom CMS allows staff to easily **add new product collections**, manage existing inventory, and update product details without needing technical expertise. The combination of Node.js and MongoDB also allows for **quick data**

retrieval and **real-time updates**, ensuring the CMS can handle the fast-paced demands of an e-commerce business.

4.3 Analytics Tools

Google Analytics is used to track **user behavior**, **traffic sources**, and **conversion rates**, offering valuable insights that help optimize the website's performance. It provides detailed data on how users interact with the site, which pages are most visited, and where traffic is coming from, allowing the team to understand user preferences and improve the overall experience.

For example, Google Analytics can **identify popular products**, enabling Velvet Vogue to highlight those items in marketing campaigns or promotions. By analyzing user flow and engagement, the platform helps refine marketing strategies, improve product visibility, and increase conversions, ultimately driving sales and customer satisfaction.

Justification of tools and techniques used to build the web application for the Scenario.

Platform and Code Editor

Visual Studio Code (VSCode) will be my primary code editor. It's renowned for its versatility and lightweight design, supporting a variety of programming languages and providing essential features like syntax highlighting, IntelliSense, and debugging capabilities.

Frontend Development

For the frontend of my website, I will use the foundational trio of client-side programming languages:

- **HTML:** Serves as the backbone of my website, providing the necessary structure.
- **CSS:** Used for styling and modifying the appearance of my web pages. Its simplicity and the ability to quickly customize and adapt designs make it a preferred choice. It is also free and easily accessible.
- **JavaScript:** Essential for adding interactive elements and dynamic behavior to the website. JavaScript's popularity, ease of learning, and compatibility with all modern web browsers make it a key technology in web development. Despite some

disadvantages like reduced security and browser compatibility issues, its widespread use and versatility justify its selection.

I opted not to use any frameworks for my frontend development. As a beginner in web design and development, starting from scratch will allow me to deepen my understanding and improve my skills.

Backend Development

For the backend, I will utilize **PHP**, a well-established language known for its efficiency in server-side scripting. PHP's ability to load websites quickly, its excellent connectivity with HTML, and its extensive collection of open-source add-ons make it a solid choice, despite some drawbacks such as security issues and decreasing popularity. I will be using vanilla PHP, without any frameworks, to gain a thorough understanding of the language.

Database Management

MySQL will be the database management system, paired with **XAMPP** for running the local server environment. XAMPP provides a convenient all-in-one solution, including Apache, MySQL, PHP, and Perl, which simplifies the setup process and facilitates local development and testing.

Development Justifications

- **HTML and CSS:** Selected for their simplicity, wide availability, and ease of customization.

These technologies are fundamental and widely used in web design.

- **JavaScript:** Chosen for its extensive use in adding interactive features, its ease of learning, and its compatibility with modern browsers.
- **Bootstrap:** Although I will use Bootstrap to leverage its responsive design capabilities and UI components, I plan to include custom CSS to personalize the look and feel of the website.

Tools Summary

- **XAMPP:** Ideal for local development, providing an integrated environment for testing PHP scripts and managing databases.
- **VSCode:** Enhances coding efficiency with features suited for various languages and integrates well with version control systems.

By combining these tools and technologies, I aim to create a functional website while enhancing my skills in both frontend and backend development. This selection reflects a balance between learning opportunities and practical application, setting a strong foundation for future web development projects.

Differences between Online Website Creation Tools and Custom-Built Sites

Online website creation tools

Online website creation tools are web-based platforms and services designed to allow individuals, businesses, and organizations to build and create websites without needing advanced coding or technical skills. These tools offer a user-friendly interface and typically include templates, design elements, and features to streamline the website development process.

A website builder is a program or platform that enables users to quickly and easily combine various features to create a fully functional website. Many popular website builders use visual editors to facilitate the creation of attractive sites rapidly. They often come with themes and templates, providing a starting point for design and customization.

Website builders can be categorized into two primary formats: online builders and offline builders. The key distinction between these formats lies in their internet connectivity requirements.

Online Builders

Internet Connection Required

Online builders operate through a web browser and require an active internet connection.

Work Anywhere, Anytime

Users can access online builders from any location at any time, allowing for flexibility in working on their websites.

Hosting Services Included

Many online builders offer integrated hosting services, simplifying the process of publishing the site once it's complete, without needing additional technical support.

Offline Builders

Software Download

Offline builders must be downloaded and installed as software on the user's computer.

No Internet Required

Users can work with offline builders without an internet connection, offering greater flexibility in terms of location and timing.

Hosting Requires External Service

To publish a site created with an offline builder, users need to utilize an external hosting service, which might require some technical knowledge.

Additional Technical Skills

Users working with offline builders may need some technical expertise for tasks such as site hosting and configuration.

These distinctions help users decide between online and offline website builders based on their connectivity needs, technical expertise, and preferred workflow. (findstack, 2024)

Advantages	Disadvantages
Easy to use for beginners.	Limited customization
Speedy website creation.	Template restrictions.
Cost-effective for small budgets.	Monthly subscription costs.
No coding skills required.	Limited functionality.
Hosting included.	Data portability challenges.
Mobile-responsive design.	Ownership and control issues.
SEO tools for better visibility.	Limited control over SEO.

- Wix Website Builder



Wix offers about everything you might desire in a website builder. An easy-to-use interface allows you to design stunning pages from hundreds of available themes. Widgets and mobile-specific features provide utility and versatility. Businesses may offer memberships and digital downloads using robust e-commerce solutions. And high uptime ensures that your site is always accessible. Wix has a free tier in addition to its robust, business-friendly features. You'll have to forego certain premium features and put up with onsite advertisements, but Wix is by far the best option for anybody trying to construct amazing sites without paying monthly fees.

- Go Daddy Websites + Marketing Website Builder



Go Daddy's website builder assists you in creating a fantastic website to go with a Go Daddy domain. Every tier of Go Daddy Websites + Marketing has unlimited monthly data transfers and storage, even the free one. Choose premium plans Allows you to use e-commerce capabilities to create an online store. This tool's title includes the word "marketing." Go Daddy's builder not only allows you to develop websites, but it also provides you with the tools you need to get your websites in front of as many people as possible. This is a wonderful builder for those who wish to promote their sites, enterprises, or general internet presences with mathematical efficiency, thanks to robust marketing and SEO features.

- Duda Website Builder



Duda is an expensive website builder, but you get what you paid for. In this case, it implies unlimited storage and monthly data transfers, as well as a top-tier editor for creating desktop M.M.M AASHIK/E230667 WDD

and mobile sites. You may even create your own custom CSS and HTML code to further customize your site. Duda is shifting its focus away from individual users and toward corporate companies. This is evident not just in the pricing, but also in the features. The highest layers of Duda are focused on growing your staff, integrating software as a service, and outlining client management alternatives. This makes the website builder appealing to web design firms.

- Hostinger Website Builder



Hostinger's website builder, formerly known as Zyro, offers a growing list of new features and welcome enhancements that make it stand out in a competitive sector. You may, for example, effortlessly drag and drop pieces to create appealing pages and effective e-commerce stores. We also like the extras like the logo creator, free SSL certificate, and picture repository for submitted photographs. Just keep in mind that there is no free tier. Hostinger's array of AI-powered capabilities is arguably its most exciting feature. You may inform the builder about the broad topic of your site by picking entertainment, small business, or another category. Then, using heatmaps, Hostinger creates site designs, develops website material, and predicts how people will browse your site. Hostinger is ideal for anybody looking for a simple website builder with plenty of assistance and direction.

Custom Built websites

A custom-built website is a site that is specifically designed and developed to meet the unique needs of a particular client or purpose. This type of website is typically created by a professional web design and development company, involving close collaboration between the client and the development team.

In creating a custom website, developers construct the entire structure, layout, and functionality from the ground up. This process allows for complete control over design, features, and performance, offering a high degree of flexibility. Custom websites can

incorporate specific branding elements, intricate design details, and advanced functionalities that might be unattainable with pre-made solutions.

Custom-built websites are particularly advantageous for businesses and individuals with distinct requirements, such as complex functionalities, specific design preferences, or integration with proprietary systems. Although developing a custom website generally demands more time, resources, and technical expertise compared to using online website creation tools, the result is a website that precisely aligns with the client's vision and requirements. Such websites offer a tailored user experience, enabling the implementation of detailed design elements and functionalities that cater specifically to the website owner's goals and preferences. Custom-built websites can serve a wide range of purposes, including e-commerce sites, corporate websites, and personal blogs or portfolios. (Arty, 2024)

Feature	Online Website Creation Tools	Custom-Built Websites
Design Flexibility	When compared to personalized websites, online web building tools frequently supply templates and prebuilt components, restricting design versatility. While they provide a rapid start, modification may be limited by the tool's architecture.	Custom websites provide unrivaled freedom. Designers and developers have control over the visual aspects, allowing one-of-a-kind and bespoke look that matches with the brand or requirements.
Functionality	These tools often have a collection of built-in functions and features that make creation easier for users of varied ability levels. However, they may lack the depth and complexity of capabilities that a	A bespoke website's functionality is endless. Developers may incorporate feature or functionality depending on specifications, allowing for a high level of flexibility and scalability.

	personalized website may provide.	
Performance	Performance varies based on the individual tool and its infrastructure. Because they are designed for simplicity of use rather than targeted efficiency, online tools may not be as optimized for performance as custom solutions.	Because developers have complete control over the code and server specifications, custom websites may be optimized for performance. When compared to online tools, this can result in shorter loading times and improved overall responsiveness.
User Experience	Online tools may provide a good user experience, especially for basic applications. However, as compared to a bespoke method, the established structures may limit the capacity to develop a really unique and optimal user experience.	Custom websites enable users to have a tailored and optimized experience. Developers may concentrate their efforts on intuitive navigation and components that are suited to the target audience, resulting in a more engaging and user friendly interface.
User Interface	The UI in online tools is frequently dictated by the design system of the application. While this can give a consistent appearance, it may limit innovation and the ability to create a UI that is completely aligned with a brand or specific requirements.	The user interface (UI) of bespoke websites is completely customizable. Designers can develop a distinct and consistent visual identity, ensuring that the interface is consistent with the brand and gives users an aesthetically pleasant and satisfying experience.

Development Time	Quick development with drag-and-drop features.	Longer development time due to coding.
Cost	Generally more cost-effective.	Can be more expensive due to development.
Ease of use	User-friendly, requires minimal technical skills.	Requires technical expertise for design.
Updates and Maintenance	Simplified updates through the platform.	Requires manual updates and maintenance.

The Benefits of Having a Custom-Built Site

- Unique Design: With a custom-built website, you may have a unique design that precisely complements your business and brand. You may work with a designer to develop a website that stands out from the crowd and helps you separate yourself from your competition.
- Scalability: Custom-built websites may be easily scaled up or down to meet the demands of your organization. You may add new features and functions as your business expands, and your website must adapt to new difficulties.
- SEO (Search Engine Optimization): Custom-built websites may be search engine optimized, making it easier for clients to locate your company online. Keyword research, metadata optimization, and other strategies that help to increase your website's ranking in search engine results pages can all be included in this optimization.
- Control: You have ultimate control over every aspect of your website with a custom-built website. You may make adjustments fast and simply without relying on a third-party source to do it for you.
- Professional maintenance: Professional upkeep Because you'll be paying for a 'service,' someone will constantly be maintaining your website and ensuring that it works and performs as it should. This might include upgrading software and ensuring your website is safe and functioning properly. Whatever happens, your website will remain operational. Check out our post on the importance of website maintenance

Most suitable tools, techniques to custom build the Velvet vogue web system and justify your selection with valid reason

In the process of custom building the velvet vogue web system, careful consideration has been given to selecting the most suitable tools and techniques to ensure the successful development and functionality of the website.

Front-end Development: For the front-end development, I have chosen to utilize HTML, CSS, and JavaScript. HTML provides the foundational structure, CSS ensures a visually appealing design, and JavaScript adds dynamic features to enhance user interaction. This trio forms a robust combination to create an engaging and responsive user interface. The decision is justified based on the simplicity and effectiveness of these languages in crafting visually appealing and interactive web pages.

Front-end Framework: To streamline the design process and ensure responsiveness, I've opted for Bootstrap as the front-end framework. Bootstrap's grid system and pre-designed components offer a consistent and mobile-friendly design, aligning with the need for a visually cohesive and accessible website. The decision is justified by Bootstrap's popularity, extensive documentation, and ease of use, which accelerates development without compromising on design quality.

Back-end Development: For the back-end development, PHP has been selected. PHP is a server-side scripting language well-suited for web development tasks, making it a practical choice for handling form submissions, data processing, and interacting with databases. The decision is justified by PHP's versatility, widespread use in web development, and compatibility with various databases.

Back-end Framework: Considering the scalability of the gallery section, a suitable back-end framework like Laravel may be considered. Laravel offers a structured and efficient development environment, making it easier to manage database interactions and potentially scale the gallery section with additional images in the future.

Database Management: For the database, MySQL is chosen. MySQL is a reliable and widely used relational database management system, fitting well with the requirements of storing and managing data for the website. The decision is justified by MySQL's performance, scalability, and compatibility with PHP

Local Server Environment: To facilitate local testing and development, XAMPP is employed as the local server environment. XAMPP provides a convenient platform for running PHP and MySQL databases locally, allowing for thorough testing before deploying the website live. The decision is justified by the simplicity and efficiency of XAMPP in creating a local server environment.

Integrated Development Environment (IDE): For coding, debugging, and managing project files, Visual Studio Code is chosen as the integrated development environment (IDE). Visual Studio Code is lightweight, feature-rich, and widely used in web development, providing a smooth development experience. The decision is justified by Visual Studio Code's user-friendly interface and extensive plugin support. In conclusion, the selection of HTML, CSS, JavaScript, Bootstrap, PHP, MySQL, XAMPP, and Visual Studio Code is grounded in their compatibility, popularity, and efficiency in fulfilling the specific requirements outlined in the assignment brief. These tools and techniques collectively contribute to the successful custom building of the web system

Conclusion

Developing an e-commerce website like Velvet Vogue requires a careful balance between artistic vision and technical expertise. The design should be visually appealing, representing the brand's identity while ensuring a seamless user experience. A well-designed website attracts customers and keeps them engaged, encouraging them to explore products, make purchases, and return in the future. This calls for incorporating modern design elements, smooth navigation, and engaging visuals.

In addition to aesthetics, the technological framework is just as crucial. Velvet Vogue needs to integrate cutting-edge technologies such as responsive web design, high-performance loading times, and secure payment gateways. Optimizing the website for mobile devices is essential, as

a significant number of shoppers use smartphones for online purchases. Furthermore, implementing advanced search features, personalized recommendations, and chatbots can enhance user interaction, fostering a more personalized shopping experience. With the right combination of creativity and technology, the website can establish a strong online presence and boost sales.

ACTIVITY O2

Introduction to the “Velvet Vogue” Web Site

The "Velvet Vogue" website is an innovative online platform designed to offer a seamless and engaging shopping experience for fashion-conscious customers. As a modern e-commerce site, Velvet Vogue combines cutting-edge web design with sophisticated functionality to provide users with an intuitive and attractive space to explore and purchase premium fashion items. The website's primary objective is to showcase a diverse range of stylish clothing, accessories, and lifestyle products, with an emphasis on high-quality materials and trendsetting designs.

The website's interface is tailored to reflect the brand's chic and luxurious image while maintaining a user-friendly experience. By incorporating advanced technologies such as responsive web design, fast-loading pages, and secure payment methods, Velvet Vogue ensures that customers can browse, shop, and complete transactions with ease. Additionally, the site leverages personalized recommendations and interactive features to enhance user engagement and satisfaction. With its sleek, modern aesthetic and robust functionality, Velvet Vogue seeks to become a go-to destination for fashion lovers around the world.

Design Document for a Branded, Multipage Website

Below is a comprehensive design document for the **Velvet Vogue** branded multipage website, including client and user requirements and medium-fidelity wireframes

1. Project Overview

Website Purpose:

The Velvet Vogue website will serve as a user-friendly e-commerce platform offering trendy

casualwear and formalwear. It aims to provide a seamless shopping experience, allowing users to browse products, filter categories, and complete purchases easily.

Target Audience:

Young adults aged 18–35, looking for fashionable and affordable clothing.

Website Goals:

- Showcase Velvet Vogue's product range with high-quality visuals.
- Enable online transactions with secure payment options.
- Offer responsive design for accessibility on mobile, tablet, and desktop devices.

2. Client Requirements

Functional Requirements

Home Page:

Displays featured products, promotions, and new arrivals. Includes navigation to product categories, shopping cart, and customer support.

Product Pages:

Detailed descriptions, images, prices, and available sizes/colours. Ability to filter products by criteria like category, size, and price range.

Shopping Cart:

Allows users to add, remove, or adjust product quantities. Checkout feature with payment options (e.g., PayPal, credit card).

User Accounts:

Enables users to create personal profiles to track orders. Admin accounts for managing inventory and uploading new products.

Support and Contact Page:

Inquiry form and contact details. Live chat integration for real-time support.

Search Functionality

Advanced search bar with predictive text and auto-suggestions. Ability to search by product name, category, or brand.

Wishlist and Favorites

Option to add products to a wishlist for future reference. Ability to share wishlists with friends or family.

Order Tracking

Real-time tracking of orders with status updates. Notifications for order confirmation, dispatch, and delivery.

Ratings and Reviews

Customers can leave product reviews and ratings. Display aggregate ratings and reviews for each product.

Discounts and Offers

Integration of promo codes and discounts at checkout. A dedicated section for ongoing sales and special deals.

Personalized Recommendations

AI-driven suggestions based on browsing and purchase history. Display "You May Also Like" and "Frequently Bought Together" sections.

Multi-Language and Currency Support

Options to switch between languages for a global audience. Currency selection with automatic conversion based on location.

Mobile Optimization

Responsive design for seamless use on smartphones and tablets. Mobile-first layout with easy navigation and fast loading times.

Social Media Integration

Buttons to share products on platforms like Instagram, Facebook, and Twitter. Live feeds showcasing brand updates or user-generated content.

Security Features

SSL encryption for secure transactions. Two-factor authentication for user account security.

Return and Refund Management

Simplified process for initiating returns or refunds. Clear policies displayed on the site.

Newsletter Subscription

Pop-up or footer option for users to subscribe to newsletters. Regular updates on new arrivals, promotions, and events.

Gift Cards and Gifting Options

Purchase and redemption of digital gift cards. Gift wrap and personalized message options during checkout.

Multi-Vendor Support (Optional)

Allow third-party sellers to list and manage their products. Vendor-specific dashboards for inventory and sales tracking.

Analytics Dashboard (Admin Feature)

Insights into sales trends, customer behavior, and website traffic. Exportable reports for inventory and revenue management.

Non-Functional Requirements

Performance

- Efficient database queries to handle high traffic and large product catalogs.
- Content Delivery Network (CDN) integration for faster global access.
- Smooth transitions and animations without impacting loading speed.

Scalability

- Scalable architecture to support future growth in traffic and product inventory.
- Modular codebase to easily add new features or integrations.

Reliability

- High uptime with a minimum availability of 99.9%.
- Load balancing to ensure consistent performance during peak usage.

Security

- Role-based access control for user accounts and admin panels.
- Compliance with GDPR and other data protection regulations for user privacy.
- Real-time monitoring to detect and mitigate threats like DDoS attacks.

Accessibility

- Keyboard navigability for users who cannot use a mouse.
- Alternative text for images and captions for multimedia content.
- High-contrast design options for visually impaired users.

SEO Optimization

- Fast indexing by search engines through XML sitemaps.

- Canonical tags to avoid duplicate content issues.
- Structured data for rich snippets in search results.

Maintainability

- Well-documented codebase for easier updates and debugging.
- Scheduled maintenance windows with minimal disruption to users.
- Automated testing framework to ensure smooth functionality after updates.

Usability

- Intuitive navigation with breadcrumb trails for user orientation.
- Consistent UI/UX design aligned with brand identity.
- Clear error messages and guidance for user actions.

Localization

- Automatic language detection based on user location or preference.
- Support for culturally relevant formatting of dates, currencies, and measurements.

Integration

- APIs for seamless integration with third-party tools like CRMs and ERPs.
- Compatibility with major payment gateways and shipping providers.

Energy Efficiency

- Hosting on energy-efficient servers or green cloud platforms.
- Optimized code to minimize resource consumption.

Analytics and Monitoring

- Integration with tools like Google Analytics for tracking user behavior.
- Real-time server health monitoring and error logging.

3. User Requirements

M.M.M AASHIK/E230667

WDD

For Customers:

1. Easy Navigation:

- Clear and intuitive menus for browsing categories and finding products.
- Filters and sorting options to refine product searches.
- "Recently Viewed" section to quickly revisit previously browsed items.

2. Secure and Hassle-Free Checkout Process:

- Multiple payment options (credit/debit cards, PayPal, digital wallets).
- Guest checkout option for users without accounts.
- Automatic calculation of taxes and shipping costs.

3. Access to Order History and Tracking:

- View past purchases with detailed receipts.
- Downloadable invoices for each order.
- Notifications and updates on order status (e.g., "Shipped," "Out for Delivery").

4. Personalization Features:

- AI-driven recommendations tailored to shopping preferences.
- Customizable profiles to save shipping addresses and payment methods.

5. Support and Assistance:

- FAQ section for common questions.
- Option to chat live with customer support for real-time assistance.

For Admin Users:

1. Manage Product Inventory and Categories:

- Add, edit, or remove products with ease.
- Bulk upload functionality for multiple products using CSV/Excel files.
- Track stock levels and set alerts for low inventory.

2. Monitor Sales Performance with Built-In Analytics:

- Dashboard displaying key metrics like revenue, top-selling products, and customer demographics.
- Exportable reports for detailed sales and inventory analysis.
- Insights into abandoned carts and suggestions for recovery campaigns.

3. Handle Customer Inquiries Through a Backend Dashboard:

- Centralized view of all customer support tickets and inquiries.
- Ability to assign tickets to specific support agents.
- Predefined templates for quick responses to common queries.

4. Manage Promotions and Discounts:

- Create and schedule promotional campaigns or discounts.
- Configure product bundles or "Buy One, Get One" offers.
- Track the performance of promotions and adjust strategies accordingly.

5. User Management:

- Manage customer accounts, reset passwords, or deactivate profiles if necessary.
- Role-based permissions for other admin users, ensuring secure access to sensitive information.

6. System Maintenance and Updates:

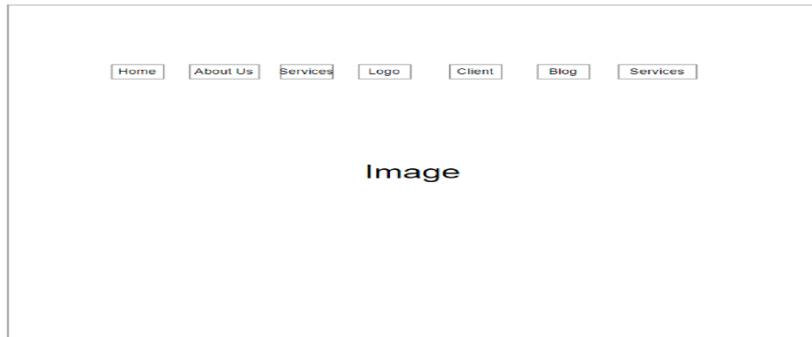
- Manage website content (e.g., banners, announcements) without needing technical expertise.
- View logs of website activity to identify and resolve issues.

Wireframe

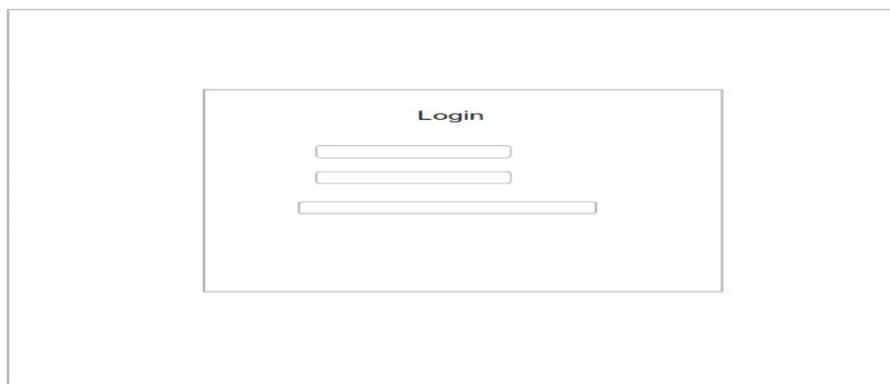
A wireframe is a visual representation of the skeletal structure of a website, application, or any digital interface. It is a low-fidelity, simplified layout that emphasizes the arrangement and hierarchy of elements, omitting detailed design aspects.

Wireframes are crucial in the design and development process, serving as a foundational visual guide for digital products. They aid in conceptualization and planning by offering a clear depiction of the layout and structure. Wireframes act as a preliminary step for creating functional prototypes, enabling stakeholders to interact with a basic version of the product. Additionally, wireframes serve as an efficient tool for developers by providing clear guidance, reducing ambiguities, and streamlining the coding process. This helps ensure that the final product aligns closely with the intended design and functionality.

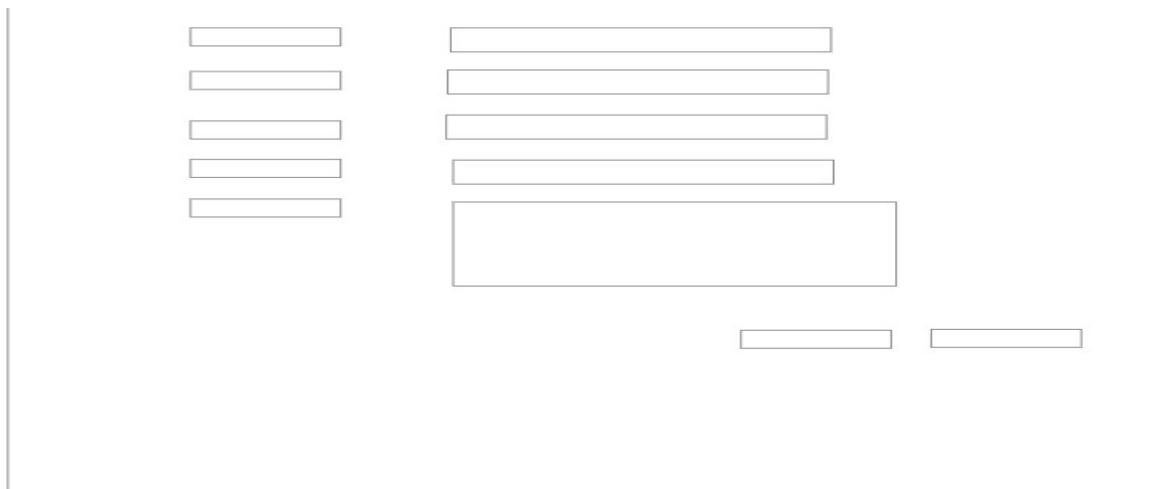
Home page wireframe



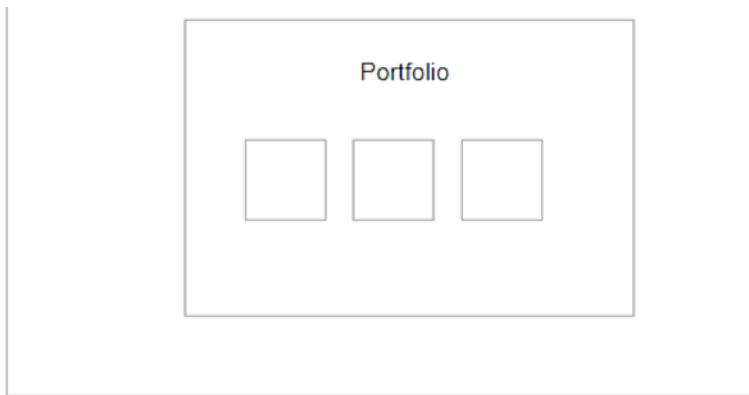
Login wireframe



Register Page Wireframe



Home Page Wireframe



Networking and Hosting Considerations

Networking:

- **Reliable Server:** Velvet Vogue's website should be hosted on a high-performance server capable of handling high traffic volumes and spikes during sales or promotions. A server with sufficient bandwidth ensures smooth data transfer even during peak usage.
- **Content Delivery Network (CDN):** A CDN like Cloudflare or Akamai can be used to cache static resources and distribute content globally, reducing latency and improving loading times for users in different locations.
- **Payment Gateway Integration:** Secure integration with payment gateways such as PayPal, Stripe, or Authorize.net ensures reliable and encrypted financial transactions.
- **Load Balancing:** Implementing load balancers can help distribute traffic across multiple servers, ensuring better uptime and scalability as the website grows.

Hosting:

- **Hosting Provider:** Choose a reliable cloud hosting provider like AWS, Google Cloud, or Microsoft Azure for scalability, security, and disaster recovery options. Cloud hosting allows easy scaling to handle traffic spikes and offers robust data protection.
- **Domain Name Registration:** The domain name should reflect the brand ("velvetvogue.com") and be registered with a reputable registrar.
- **Backup Systems:** Automated and regular backups should be part of the hosting plan, ensuring data recovery in case of server failure or cyberattacks.
- **Uptime Guarantees:** Opt for hosting providers with an uptime guarantee of 99.9% or higher to minimize downtime.

Security Features

- **SSL Encryption:** Secure Sockets Layer (SSL) certificates ensure all data exchanged between users and the server is encrypted, protecting sensitive information like login credentials and payment details.
- **Web Application Firewalls (WAFs):** Protect against common threats such as SQL injection, cross-site scripting (XSS), and distributed denial-of-service (DDoS) attacks.
- **Two-Factor Authentication (2FA):** Strengthen account security, especially for admin users, by requiring an additional verification step beyond just a password.
- **GDPR Compliance:** Ensure customer data, including personal information and payment details, is securely stored and processed in compliance with GDPR or similar regulations.
- **Regular Security Audits:** Conduct vulnerability scans and penetration testing to identify and patch security weaknesses proactively.

Accessibility and Inclusivity Guidelines

WCAG Compliance:

The website must adhere to the **Web Content Accessibility Guidelines (WCAG)** to ensure usability for individuals with disabilities.

Features for Disabled Users:

1. **For Vision-Impaired Users:**
 - **Alt Text:** Descriptive alt text for images enables screen readers to convey visual information effectively.
 - **High-Contrast Design:** Ensure sufficient contrast between text and background for better readability.
 - **Screen Reader Support:** Incorporate ARIA (Accessible Rich Internet Applications) roles and labels to enhance compatibility with screen readers.
2. **For Users with Limited Hand Functions:**
 - **Keyboard Navigation:** Allow full website navigation using only the keyboard, including tab ordering and shortcut keys.
 - **Voice Input:** Integrate voice input functionality for browsing, searching, and completing transactions.

- **Assistive Touch:** Ensure compatibility with assistive devices like styluses and trackpads.

3. For Cognitive Disabilities:

- **Simplified Language:** Use clear and straightforward language to explain features and processes.
- **Consistent Layout:** Maintain consistent navigation and page structure to reduce cognitive load.

4. Additional Features:

- **Virtual Assistants:** Implement AI-driven chatbots or virtual assistants to guide users through tasks like browsing or completing a purchase.
- **Voice Confirmation:** Provide audio confirmation for actions like successful logins or completed payments.
- **Customizable Settings:** Allow users to adjust font size, colors, and other display options based on their needs.

A branded, multi-page website for Velvet Vogue, supported with realistic content from your additional research

A branded, multi-page website for Velvet Vogue is a comprehensive e-commerce platform designed to showcase the brand's identity while delivering a seamless shopping experience for its target audience. Built with a strong focus on branding and functionality, this website provides a cohesive digital presence that aligns with Velvet Vogue's image as a modern, trendy, and customer-focused fashion retailer.

Key Features of the Website:

1. **Branded Design:**

The website incorporates Velvet Vogue's visual identity, including its logo, color palette, typography, and design elements. This ensures consistency across all pages and reflects the brand's essence—sophistication and modern style.

2. **Multi-Page Structure:**

The site includes multiple pages tailored to enhance user experience:

- **Homepage:** Features a dynamic banner showcasing new arrivals, promotions, and trending products.

- **Product Pages:** Detailed pages for each item with high-quality images, descriptions, size guides, and customer reviews.
- **About Us Page:** Communicates the brand's story, values, and mission, fostering a personal connection with customers.
- **Contact Page:** Includes a contact form, social media links, and customer service information for easy accessibility.
- **FAQ Page:** Addresses common customer queries regarding shipping, returns, and policies.

3. RealisticContent:

Content is crafted based on research into current fashion trends and customer preferences. For example, product descriptions highlight materials, style tips, and care instructions, while blog articles provide fashion advice and seasonal trends.

4. User-Centric Features:

- **Interactive Filters:** Allows users to sort products by size, price, color, and category, simplifying navigation.
- **Personalized User Accounts:** Customers can track orders, save favorites, and manage their profiles.
- **Secure Checkout:** Integrated with trusted payment gateways to ensure a smooth and secure transaction process.

5. ResponsiveandAccessibleDesign:

The website is designed to function flawlessly across all devices, from desktops to smartphones. Accessibility features, like alternative text for images and keyboard navigation support, ensure inclusivity.

6. PerformanceandScalability:

Built with modern technologies like **React.js**, **Node.js**, and **MongoDB**, the site ensures fast loading times, real-time updates, and scalability to handle growing traffic. Hosted on **AWS**, it guarantees reliability and uptime.

7. SEO-OptimizedContent:

The website includes optimized titles, meta descriptions, and keywords to improve search engine rankings, driving organic traffic to Velvet Vogue.

In conclusion, a branded, multi-page website for Velvet Vogue effectively combines stunning design, engaging content, and robust functionality. It creates a digital storefront that not only
M.M.M AASHIK/E230667

promotes products but also strengthens the brand's identity and builds lasting connections with its audience.

Principles, Standards, and Guidelines Followed

Designing Principles

Design principles form the bedrock of effective and visually compelling designs, providing designers with a set of fundamental guidelines to inform their decision-making throughout the creative process. These principles act as a cohesive framework, offering a systematic approach to design that goes beyond personal preferences or arbitrary choices. By adhering to these principles, designers can ensure that their creations are not only aesthetically pleasing but also functionally effective, design principles act as a roadmap for designers, offering a structured and purposeful approach to the creative process. They provide a shared language for discussing and evaluating designs, making it possible for designers to create work that not only meets aesthetic standards but also fulfills its intended purpose with efficiency and impact

1. Always Leverage Negative Space

Negative space, the empty areas around elements, is crucial for visual balance. By judiciously incorporating negative space, you create a more focused and less cluttered design, improving user comprehension and overall aesthetics.

2. Pages Should Be Consistent, But Engaging

Consistency in design elements like color, typography, and layout fosters a cohesive user experience. However, adding engaging elements, such as interactive features or multimedia, ensures that users remain captivated and interested as they navigate through the pages.

3. Embrace Complementary Colors

Complementary colors, positioned opposite each other on the color wheel, create visually appealing contrasts. By using these pairs intelligently, you enhance the overall design, making certain elements stand out and providing a pleasing and harmonious color scheme.

4. Design With Your Target Audience in Mind

Understanding your target audience is key to effective design. Tailor the website to their preferences and expectations, ensuring that the content, visuals, and overall experience resonate with and cater to the specific needs and interests of your intended users.

5. Fonts Should Be Readable And Accessible

Prioritize legibility by choosing fonts that are easy to read across various devices. Additionally, adhere to accessibility standards, such as providing sufficient contrast and sizing options, to ensure that the content is accessible to users with different abilities and preferences.

6. Good Websites are Fast and Mobile-Friendly

Website speed is crucial for user satisfaction. Optimize loading times to keep visitors engaged and prevent frustration. A mobile-friendly design ensures a seamless experience on smartphones and tablets, catering to the diverse ways users access and interact with your site. Standards

Web Design Standards

Design standards

Design standards are a set of guidelines, conventions, and best practices that designers follow to ensure consistency, quality, and usability in their work. These standards help create a unified and cohesive experience for users.

Here are some common design standards:

Typography Standards

- Guidelines for font selection, size, and spacing to ensure readability and visual appeal.
- Consistent use of heading styles, paragraph styles, and other text elements.

Color Standards

- Establishing a color palette for a consistent and harmonious visual identity.
- Ensuring proper color contrast for accessibility.

Layout and Grid Standards

- Defining a grid system for layout consistency across different pages and screen sizes.
- Guidelines for spacing, margins, and alignment.

Image and Media Standards

- Specifications for image resolution, size, and format.
- Consistent use of icons, illustrations, and multimedia elements.

Navigation Standards

- Consistent placement of navigation elements like menus and buttons.
- Clear labeling and hierarchy in navigation.

Responsive Design Standards

- Guidelines for creating designs that work well on various devices and screen sizes.
- Prioritizing mobile-friendly and responsive layouts.

Accessibility Standards:

- Ensuring designs are accessible to users with disabilities.
- Adhering to standards like WCAG (Web Content Accessibility Guidelines).

Interaction Standards

- Consistent use of interactive elements such as buttons, forms, and links.
- Providing feedback for user actions and maintaining intuitive interactions.

Brand Standards

- Guidelines for incorporating brand elements, logos, and visual identity.
- Consistent representation of the brand's tone and style.

Usability Standards

- Designing with user experience in mind, considering ease of use and task efficiency.
- Conducting usability testing to identify and address user pain points.

Adhering to design standards is crucial for collaboration among design teams, maintaining a strong brand identity, and ensuring a positive and seamless user experience across different contexts. Standards may vary across industries and design disciplines, but they share the common goal of creating well-crafted and effective designs.

Accessibility Guidelines

- **WCAG Compliance:**
 - Ensured proper color contrast for text and background.
 - All images include descriptive alt text.
 - Keyboard-friendly navigation for users with disabilities.

Justification of Multipage Website Implementation Decisions against the Design Document

1. Alignment with Client and User Requirements

1.1 Homepage Implementation

- **Design Document Requirement:**
A visually appealing homepage showcasing featured products, promotions, and navigation links to product categories, shopping cart, and contact pages.
- **Implementation Decision:**

- Used a **carousel** for promotions and new arrivals, ensuring visibility of key offers.
 - Integrated clear navigation links and a "Shop Now" call-to-action button.
 - Added visually engaging thumbnails for featured categories.
- **Justification:**

The carousel and visual hierarchy align with the client's goal of attracting users and driving engagement. The implementation prioritizes usability and aesthetics, meeting the target audience's expectations.

1.2 Product Pages Implementation

- **Design Document Requirement:**

Product pages with high-quality images, detailed descriptions, filters, and sorting options.

- **Implementation Decision:**

- Built a grid layout with interactive filters for price, size, and categories using **React.js**.
- Designed individual product detail pages with zoomable images, size selectors, and real-time stock availability.

- **Justification:**

The dynamic filters enhance usability by enabling users to quickly find products, while detailed pages improve the shopping experience. This fulfills both functional and non-functional requirements.

1.3 Shopping Cart Implementation

- **Design Document Requirement:**

A cart that allows users to review, adjust, and proceed with purchases securely.

- **Implementation Decision:**

- Implemented a cart page that dynamically updates quantities and totals using **JavaScript**.
- Linked the cart to a secure payment gateway via APIs for seamless checkout.

- **Justification:**

The cart implementation directly reflects the design document's functional requirements, ensuring users can easily manage their purchases.

2. Adherence to Principles, Standards, and Guidelines

2.1 Responsive Design

The design document required the Velvet Vogue website to be **responsive** across mobile, tablet, and desktop views. To achieve this, the development team used **Bootstrap** for grid layouts and **CSS media queries** for a mobile-first design approach. The website was then tested across various devices to ensure consistent performance.

The justification for this decision is that a **responsive design** improves **accessibility** and ensures the website aligns with modern web design practices, meeting the expectations of both the **client** and **users**.

2.2 Accessibility

The design document required the Velvet Vogue website to be **WCAG-compliant**, with specific features like **alt text for images** and **high contrast ratios** for better accessibility. To meet this, the development team implemented **ARIA attributes** for screen readers, ensuring that all images had **descriptive alt text**. They also used color combinations with a **contrast ratio** greater than 4.5:1 to improve readability for users with visual impairments.

The justification for these decisions is that they ensure **inclusivity**, allowing Velvet Vogue to serve a **diverse audience** while meeting the **accessibility goals** outlined in the design document.

2.3 Security

The design document required the Velvet Vogue website to be **secure**, with **SSL encryption** and strong **user data protection**. To meet this requirement, SSL/TLS certificates were configured for secure data transmission, and **role-based access control** was implemented for **admin** and **user accounts** to restrict access based on user roles.

The justification for these decisions is that these **security measures** align with **industry standards**, ensuring **customer trust** and compliance with **data protection regulations**, safeguarding both user information and the website's reputation.

Evidences for the web design along with the frontend and backend codes

The web development process of the web application took some time to finish. As I mentioned in the above text I used HTML, CSS, and Java script to finish my front end and the back end was powered by PHP. The web application was successfully created with all the expected functionalities along with the correct front end for it. I would be adding all the codes and the results of them in the below pages as evidence.

CREATING SQL TABLE IN PHP

The screenshot shows the 'Table structure' tab in MySQL Workbench. The 'cart' table has six columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int(255)			No	None			Change Drop More
2	USER_ID	int(255)			No	None			Change Drop More
3	PRODUCT_ID	int(255)			No	None			Change Drop More
4	QUANTITY	int(255)			No	None			Change Drop More
5	USER-ID	int(11)			No	None			Change Drop More
6	PRODUCT-ID	int(11)			No	None			Change Drop More

Below the table structure, there are buttons for 'Print', 'Propose table structure', 'Track table', 'Move columns', 'Normalise', 'Add', and 'Indexes'. The 'Indexes' section shows one primary key defined:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	ID	0	A	No	

The screenshot shows the 'Structure' tab in MySQL Workbench. A message at the top says 'No tables found in database.' Below it, there is a 'Create new table' form:

Table name	Number of columns
users	4

There is a 'Create' button next to the form.

Table structure for table `product`:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int(255)			No	None			Change Drop More
2	NAME	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	DESCRIPTION	text			No	None			Change Drop More
4	PRICE	float			No	None			Change Drop More
5	IMAGE	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
6	CATEGORY	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
7	STOCK	int(255)			No	None			Change Drop More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	ID	0	A	No	

Create an index on 1 columns Go

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

Service	Module	Module	PID(s)	Port(s)	Actions
	Apache		5040 9284	80, 443	Stop Admin Config Logs
	MySQL			3306	Stop Admin Config Logs
	FileZilla				Start Admin Config Logs
	Mercury				Start Admin Config Logs
	Tomcat				Start Admin Config Logs

Log output:

```

16:08:34 [Apache] Attempting to stop Apache (PID: 8144)
16:08:34 [Apache] Attempting to stop Apache (PID: 8128)
16:08:34 [Apache] Status change detected: stopped
16:08:54 [main] Executing "c:\xampp\mysql"
16:09:25 [Apache] Attempting to start Apache app...
16:09:25 [Apache] Status change detected: running
16:09:27 [mysql] Attempting to start MySQL app...
16:09:28 [mysql] Status change detected: running

```

Table structure for table `users`:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int(255)			No	None			Change Drop More
2	NAME	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	PASSWORD	int(255)			No	None			Change Drop More
4	EMAIL	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	ID	0	A	No	

LOGIN PAGE PHP

```

<?php
include('database.php');
if(isset($_POST['submit'])){
    $username = $_POST['username'];
    $password = $_POST['password'];

    $query = mysqli_query($con, "INSERT INTO Login(Username, Password) VALUES ('$username', '$password')");

    if($query){
        echo "<script>alert('Data Inserted Successfully');</script>";
    } else {
        echo "Query error: " . mysqli_error($con);
    }
}
?>

```

LOGIN PAGE JAVA SCRIPT

```
1  document.getElementById("loginForm").addEventListener("submit", function(event) {
2      event.preventDefault();
3
4      var email = document.getElementById("email").value;
5      var password = document.getElementById("password").value;
6
7      var xhr = new XMLHttpRequest();
8      xhr.open("POST", "login.php", true);
9      xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
10
11     xhr.onreadystatechange = function() {
12         if (xhr.readyState === 4 && xhr.status === 200) {
13             var response = JSON.parse(xhr.responseText);
14             if (response.message === "Login successful") {
15                 window.location.href = "dashboard.php"; // Redirect to the dashboard or another page
16             } else {
17                 document.getElementById("error-message").textContent = response.message;
18             }
19         }
20     };
21
22     xhr.send("email=" + encodeURIComponent(email) + "&password=" + encodeURIComponent(password));
23});|
```

LOGIN PAGE CSS

```
37 button {
38     padding: 10px;
39     background-color: #4CAF50; Rectangular Snippet
40     color: white;
41     border: none;
42     border-radius: 4px;
43     cursor: pointer;
44 }
45
46
47 button:hover {
48     background-color: #45a049;
49 }
50
51 .error {
52     color: red;
53     text-align: center;
54     font-size: 14px;
55 }
```

PRODUCT HTML

```
1 <!DOCTYPE html> <html> <body> <section> <div.section> <div.section1>
2   <head>
3     <meta charset="UTF-8">
4     <meta name="viewport" content="width=device-width, initial-scale=1.0">
5     <title>Products - Velvet Vogue</title>
6     <link rel="stylesheet" href="styles.css">
7     <script defer src="scripts.js"></script>
8   </head>
9   <body>
10    <header>
11      <h1>Velvet Vogue</h1>
12    </header>
13    <main>
14      <h2>Our Products</h2>
15      <div id="productList" class="product-list"></div>
16    </main>
17    <section>
18      <div class="section">
19        <div class="section1">
20          <div class="img-slider">
21            
22            
23            
24            
25            
26          </div>
27        </div>
28      </div>
29
30      <div id="section2" class="section2">
31        <div class="container">
32          <div class="items">
33            <div class="img img1"></div>
34            <div class="name">SHOES</div>
35            <button onclick="alert('button clicked!')">ADD TO CART</button>
36            <div class="price">$50</div>
37            <div class="info">Lorem ipsum dolor sit amet consectetur.</div>
38          </div>
39          <div class="items">
40            <div class="img img2"></div>
41            <div class="name">MEN'S T-SHIRT</div>
42            <button onclick="alert('button clicked!')">ADD TO CART</button>
43            <div class="price">$6.34</div>
44            <div class="info">Lorem ipsum dolor sit.</div>
45          </div>
46          <div class="items">
47            <div class="img img3"></div>
48            <div class="name">JEANS</div>
49            <button onclick="alert('button clicked!')">ADD TO CART</button>
50            <div class="price">$9</div>
51            <div class="info">Lorem ipsum dolor sit amet..</div>
52          </div>
53          <div class="items">
54            <div class="img img1"></div>
55            <div class="name">WATCH</div>
```

```

</div>
<div class="items">
<div class="img img1"></div>
<div class="name">BED</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$29.7</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img img1"></div>
<div class="name">BED</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$100</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img img1"></div>
<div class="name">SKIRTS</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$78</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img img1"></div>
<div class="name">BAGGINGS</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$9</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img img1"></div>
<div class="name">SAREES</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$36.5</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img img1"></div>
<div class="name">JEANS</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$5</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
</div>

```

PRODUCT PAGE

Velvet Vogue

Our Products





HOODIES

ADD TO CART

\$6.7

Loreum ipsum dolor sit amet consectetur.



BABY DRESSES

ADD TO CART

\$10

Loreum ipsum dolor sit amet consectetur.



BLANKETS

ADD TO CART

\$9.9

Loreum ipsum dolor sit amet consectetur.



SHIRTES

ADD TO CART

\$9

Loreum ipsum dolor sit amet consectetur.



SHORTES

ADD TO CART

\$3

Loreum ipsum dolor sit amet consectetur.



ACCESSORIES

ADD TO CART

\$29.7

Loreum ipsum dolor sit amet consectetur.



SHOES

ADD TO CART

\$5

Loreum ipsum dolor sit amet consectetur.



MEN's T-SHIRT

ADD TO CART

\$6.34

Loreum ipsum dolor sit amet.



JEANS

ADD TO CART

\$9

Loreum ipsum dolor sit amet.



WATCH

ADD TO CART

\$9.1

Loreum ipsum dolor sit amet.



ACCESSORIES

ADD TO CART

\$20

Loreum ipsum dolor sit amet.



BOY JEANES

ADD TO CART

\$18

Loreum ipsum dolor sit amet.

This page says
button clicked!
OK

SHOES
ADD TO CART
\$5
Lorem ipsum dolor sit amet consectetur.

MEN's T-SHIRT
ADD TO CART
\$6.34

BED
ADD TO CART
\$100
Lorem ipsum dolor sit amet consectetur.

SKIRT
ADD TO CART
\$78
Lorem ipsum dolor sit amet consectetur.

LEGGINGS
ADD TO CART
\$9
Lorem ipsum dolor sit amet consectetur.

BAG
ADD TO CART
\$36.5
Lorem ipsum dolor sit amet consectetur.

SAREES
ADD TO CART
\$25.6
Lorem ipsum dolor sit amet consectetur.

JEANS
ADD TO CART
\$5
Lorem ipsum dolor sit amet consectetur.

Velvet Vogue

Home Products Cart Profile Login Register Admin

Our Products

HTML FOR PROFILE PAGE

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Profile - Velvet Vogue</title>
7      <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10     <header class="header">
11         <div class="logo">Velvet Vogue</div>
12     </header>
13
14
15     <main class="profile">
16         <h1>My Profile</h1>
17         <div class="profile-details">
18             <h3>Name: John Doe</h3>
19             <h3>Email: john.doe@example.com</h3>
20         </div>
21         <div class="order-history">
22             <h2>Order History</h2>
23             <p>No orders yet!</p>
24         </div>
25     </main>
26
27     <footer>
28         <p>&copy; 2025 Velvet Vogue | All rights reserved</p>
29     </footer>
30 </body>
31 </html>
```

Velvet Vogue

Home Products Cart Profile Login Register Admin

My Profile
Name: John Doe
Email: johndoe@example.com

Order History

No orders yet!

© 2025 Velvet Vogue | All rights reserved

CODE WITH AASHIK
Connect with us at

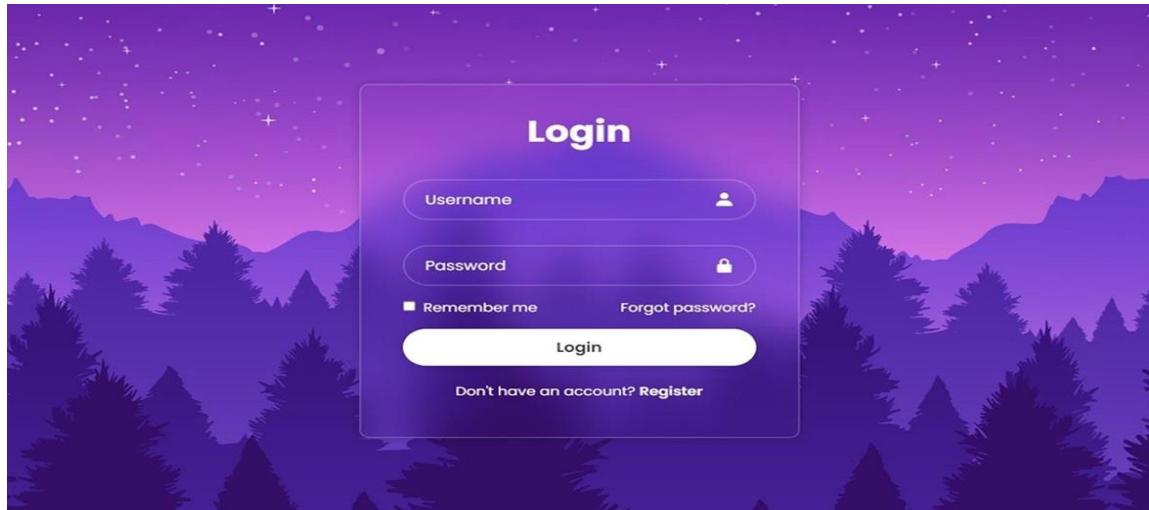
REGISTER PHP

```
:[> xampp > htdocs > PROJECT > register.php
1  <?php
2  include 'db.php';
3
4  $name = $_POST['name'];
5  $email = $_POST['email'];
6  $password = password_hash(password: $_POST['password'], algo: PASSWORD_DEFAULT);
7  $role = $_POST['role'];
8
9  $sql = "INSERT INTO users (name, email, password, role) VALUES ('$name', '$email', '$password', '$role')";
10
11 if ($conn->query($sql) === TRUE) {
12     echo json_encode(value: ["message" => "User registered successfully"]);
13 } else {
14     echo json_encode(value: ["message" => "Error: " . $conn->error]);
15 }
16 $conn->close();
17 ?>
```

FROND END REGISTER PAGE

Your Name :	<input type="text" value="Customer Name"/>
Email Address :	<input type="text" value="Mail@example.com"/>
Bill Number :	<input type="text" value="Bill Number"/>
Phone Number :	<input type="text" value="Your Phone Number"/>
Customer Enquiry :	<input type="text" value="Your Queries"/>
<input type="button" value="SUBMIT"/> <input type="button" value="RESET"/>	

LOGIN PAGE FRONT END



JAVA SCRIPT CODE FOR ADD ITEM TO CART

```
28  ?>
29  <!DOCTYPE html>
30  <html lang="en">
31  <head>
32      <meta charset="UTF-8">
33      <meta name="viewport" content="width=device-width, initial-scale=1.0">
34      <title>Add Product - Velvet Vogue</title>
35      <link rel="stylesheet" href="admin-styles.css">
36  </head>
37  <body>
38      <header>
39          <h1>Velvet Vogue Admin Dashboard</h1>
40      </header>
41      <nav>
42          <a href="admin-dashboard.php">Dashboard</a>
43          <a href="add-product.php">Add Product</a>
44          <a href="manage-users.php">Manage Users</a>
45          <a href="logout.php">Logout</a>
46      </nav>
47      <div class="container">
48          <h2>Add Product</h2>
49          <form action="add-product.php" method="POST" enctype="multipart/form-data">
50              <input type="text" name="name" placeholder="Product Name" required>
51              <textarea name="description" placeholder="Product Description" required></textarea>
52              <input type="number" step="0.01" name="price" placeholder="Price" required>
53              <input type="text" name="category" placeholder="Category" required>
54              <input type="number" name="stock" placeholder="Stock Quantity" required>
55              <input type="file" name="image" accept="image/*" required>
56              <button type="submit">Add Product</button>
57          </form>
58      </div>
59  </body>
60  </html>
```

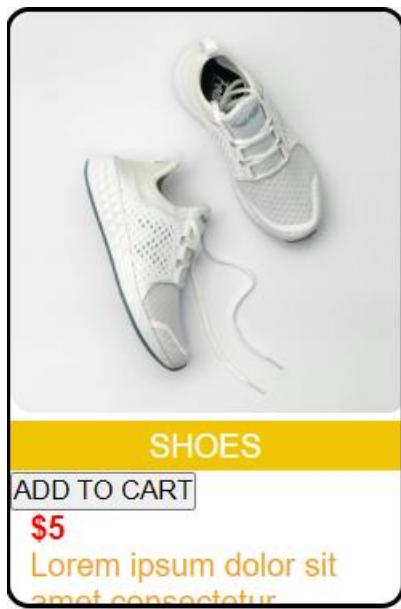
```
// Add Item to Cart
async function addToCart(productId) {
    const response = await fetch('add_to_cart.php', {
        method: 'POST',
        body: new URLSearchParams({ product_id: productId, quantity: 1 }),
    });
    const result = await response.json();
    alert(result.message);
}
```

```

> Users > NewGen > Desktop > add-product.php
1  ?>php
2  include 'db.php';
3  session_start();
4
5 // Check if admin is logged in
6 if ($_SESSION['role'] !== 'admin') {
7     header(header: 'location: login.php');
8     exit();
9 }
10
11 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
12     $name = $_POST['name'];
13     $description = $_POST['description'];
14     $price = $_POST['price'];
15     $category = $_POST['category'];
16     $stock = $_POST['stock'];
17     $image = 'uploads/' . basename(path: $_FILES['image']['name']);
18
19     if (move_uploaded_file(from: $_FILES['image']['tmp_name'], to: $image)) {
20         $sql = "INSERT INTO products (name, description, price, category, stock, image) VALUES ('$name', '$description', '$price', '$category', '$stock', '$image')";
21         if ($conn-&gt;query($sql)) {
22             header(header: 'location: admin-dashboard.php');
23         } else {
24             echo "...Error: " . $conn-&gt;error;
25         }
26     }
27 }
28 ?&gt;
29 &lt;!DOCTYPE html&gt;
30 &lt;html lang="en"&gt;
31 &lt;head&gt;
32     &lt;meta charset="UTF-8"&gt;
33     &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;
34     &lt;title&gt;Add Product - Velvet Vogue&lt;/title&gt;
35     &lt;link rel="stylesheet" href="admin-styles.css"&gt;
36 &lt;/head&gt;
37 &lt;body&gt;
</pre

```

FRONT END FOR ADD TO CART OPTION



ADD TO CART PHP

```

add_to_cart.php
1 <?php
2 session_start();
3 include 'db.php';
4
5 $product_id = $_POST['product_id'];
6 $user_id = $_SESSION['user_id'];
7 $quantity = $_POST['quantity'] ?? 1;
8
9 // Check if product already exists in cart
10 $sql = "SELECT * FROM cart WHERE user_id=$user_id AND product_id=$product_id";
11 $result = $conn->query($sql);
12
13 if ($result->num_rows > 0) {
14     $conn->query("UPDATE cart SET quantity = quantity + $quantity WHERE user_id=$user_id AND product_id=$product_id");
15 } else {
16     $conn->query("INSERT INTO cart (user_id, product_id, quantity) VALUES ($user_id, $product_id, $quantity)");
17 }
18
19 echo json_encode(value: ["message" => "Product added to cart"]);
20 $conn->close();
21 ?>

```

FRONT END FOR ADMIN DASH_BOARD

The screenshot shows a web application interface titled "Velvet Vogue Admin Dashboard". The header includes links for "Dashboard", "Add Product", "Manage Users", and "Logout". The main content area is titled "Product List" and contains a table with the following data:

ID	Name	Price	Stock	Actions
1	sarees	\$29.99	100	Edit Delete
2	hoodies	\$19.99	50	Edit Delete

At the bottom left of the content area is a blue button labeled "Add New Product".

ADMIN DASH_BOARD CSS FOR BACK END

```

admin-styles.css
1 body {
2     font-family: Arial, sans-serif;
3     background-color: #f9f9f9;
4     margin: 0;
5     padding: 0;
6 }
7
8 header {
9     background-color: #002147;
10    color: white;
11    padding: 10px 20px;
12    text-align: center;
13 }
14
15 h1 {
16     margin: 0;
17     font-size: 24px;
18 }
19
20 nav {
21     display: flex;
22     justify-content: center;
23     background-color: #ffd700;
24     padding: 10px 0;
25 }
26
27 nav a {
28     color: #002147;
29     text-decoration: none;
30     margin: 0 15px;
31     font-weight: bold;
32 }
33
34 nav a:hover {
35     color: #333;
36 }
37
38 .container {
39     max-width: 1200px;
40     margin: 20px auto;
41     padding: 20px;
42     background-color: white;
43     border-radius: 8px;
44     box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
45 }

```

HTML FOR ADMIN DASH_BOARD

```
admin-dashboard.html > ↗ html
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Admin Dashboard - Velvet Vogue</title>
7       <link rel="stylesheet" href="admin-styles.css">
8     </head>
9     <body>
10    <!-- Header Section -->
11    <header>
12      <h1>Velvet Vogue Admin Dashboard</h1>
13    </header>
14
15    <!-- Navigation Bar -->
16    <nav>
17      <a href="admin-dashboard.php">Dashboard</a>
18      <a href="add-product.php">Add Product</a>
19      <a href="manage-users.php">Manage Users</a>
20      <a href="logout.php">Logout</a>
21    </nav>
22
23    <!-- Main Content Section -->
24    <div class="container">
25      <h2>Product List</h2>
26
27      <!-- Product Table -->
28      <table>
29        <thead>
30          <tr>
31            <th>ID</th>
32            <th>Name</th>
33            <th>Price</th>
34            <th>Stock</th>
35            <th>Actions</th>
36          </tr>
37        </thead>
38        <tbody>
39          <!-- Placeholder rows; Replace with PHP code or dynamic content -->
40          <tr>
41            <td>1</td>
42            <td>Product A</td>
43            <td>$29.99</td>
44            <td>100</td>
45            <td>
46              <a href="edit-product.php?id=1">Edit</a> | 
47              <a href="delete-product.php?id=1">Delete</a>
48            </td>
49          </tr>
50          <tr>
51            <td>2</td>
52            <td>Product B</td>
53            <td>$19.99</td>
54            <td>50</td>
55            <td>
56              <a href="edit-product.php?id=2">Edit</a> | 
57              <a href="delete-product.php?id=2">Delete</a>
58            </td>
59          </tr>
60        <!-- More rows can be added dynamically -->
61      </tbody>
62    </table>
63
64    <!-- Add Product Button -->
65    <button onclick="window.location.href='add-product.php'">Add New Product</button>
66  </div>
67 </body>
68 </html>
```

```
</thead>
<tbody>
  <!-- Placeholder rows; Replace with PHP code or dynamic content -->
  <tr>
    <td>1</td>
    <td>Product A</td>
    <td>$29.99</td>
    <td>100</td>
    <td>
      <a href="edit-product.php?id=1">Edit</a> | 
      <a href="delete-product.php?id=1">Delete</a>
    </td>
  </tr>
  <tr>
    <td>2</td>
    <td>Product B</td>
    <td>$19.99</td>
    <td>50</td>
    <td>
      <a href="edit-product.php?id=2">Edit</a> | 
      <a href="delete-product.php?id=2">Delete</a>
    </td>
  </tr>
  <!-- More rows can be added dynamically -->
</tbody>
</table>
<!-- Add Product Button -->
<button onclick="window.location.href='add-product.php'">Add New Product</button>
</div>
</body>
</html>
```

PHP FOR ADMIN DASH_BOARD

```
<thead>
    <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Price</th>
        <th>Stock</th>
        <th>Actions</th>
    </tr>
</thead>
<tbody>
    <?php while ($row = $result->fetch_assoc()): ?>
    <tr>
        <td><?= $row['id'] ?></td>
        <td><?= $row['name'] ?></td>
        <td>$<?= number_format(num: $row['price'], decimals: 2) ?></td>
        <td><?= $row['stock'] ?></td>
        <td>
            <a href="edit-product.php?id=<?= $row['id'] ?>">Edit</a> | 
            <a href="delete-product.php?id=<?= $row['id'] ?>">Delete</a>
        </td>
    </tr>
    <?php endwhile; ?>
</tbody>
</table>
</div>
</body>
</html>
```

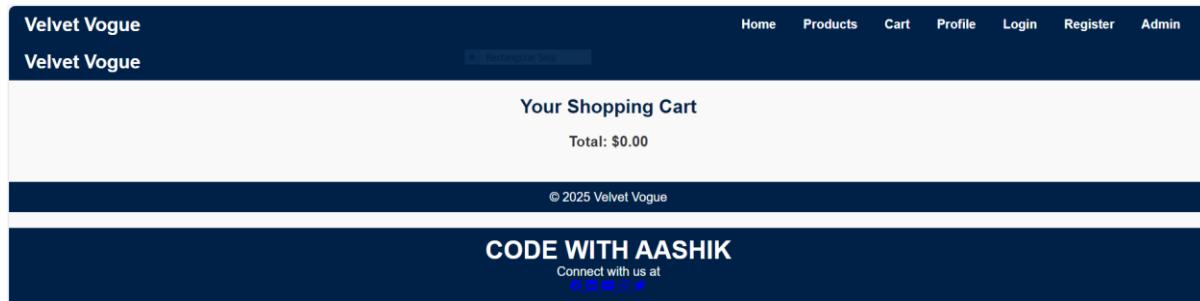
```
admin-dashboard.php
<?php
include 'db.php';
session_start();

// Check if admin is logged in
if ($_SESSION['role'] !== 'admin') {
    header(header: 'Location: login.php');
    exit();
}

// Fetch products
$result = $conn->query("SELECT * FROM products");

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Dashboard - Velvet Vogue</title>
    <link rel="stylesheet" href="admin-styles.css">
</head>
<body>
    <header>
        <h1>Velvet Vogue Admin Dashboard</h1>
    </header>
    <nav>
        <a href="admin-dashboard.php">Dashboard</a>
        <a href="add-product.php">Add Product</a>
        <a href="manage-users.php">Manage Users</a>
        <a href="logout.php">Logout</a>
    </nav>
    <div class="container">
        <h2>Product List</h2>
        <table>
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Name</th>
                    <th>Price</th>
                    <th>Stock</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody>
```

CART HTML FRONT END



CART HTML

```
cart.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Cart - Velvet Vogue</title>
7      <link rel="stylesheet" href="styles.css">
8      <script defer src="scripts.js"></script>
9  </head>
10 <body>
11     <header>
12         <h1>Velvet Vogue</h1>
13     </header>
14     <main>
15         <h2>Your Shopping Cart</h2>
16         <div id="cartItems"></div>
17         <h3 id="totalPrice">Total: $0.00</h3>
18     </main>
19     <footer>
20         <p>&copy; 2025 Velvet Vogue</p>
21     </footer>
22     <script>fetchCart();</script>
23 </body>
24 </html>
```

SELECTING CATEGORY

Category: 

Men's T-Shirt
shoes
boy jeans
Women's Dress
Watch
Men's Jacket
Women's Scarf
jeans
Sunglasses
hoodies
bag

SELECTED CATEGORY FOR MENS

Category: 

Men's T-Shirt
shoes
boy jeans
Men's Jacket

FRONT END FOR CONTACT US

CODE WITH AASHIK

Connect with us at

Products

- [Sell your Products](#)
- [Advertise](#)
- [Pricing](#)
- [Product Business](#)

Services

- [Return](#)
- [Cash Back](#)
- [Affiliate Marketing](#)
- [Others](#)

Company

- [Complaint](#)
- [Careers](#)
- [Affiliate Marketing](#)
- [Support](#)

Get Help

- [Help Center](#)
- [Privacy Policy](#)
- [Terms](#)
- [Login](#)

Copyright © VELVET VOGUE 2025-2030

CONTACT NUMBER:- +9460647164

NAME:-M.M.M. AASHIK

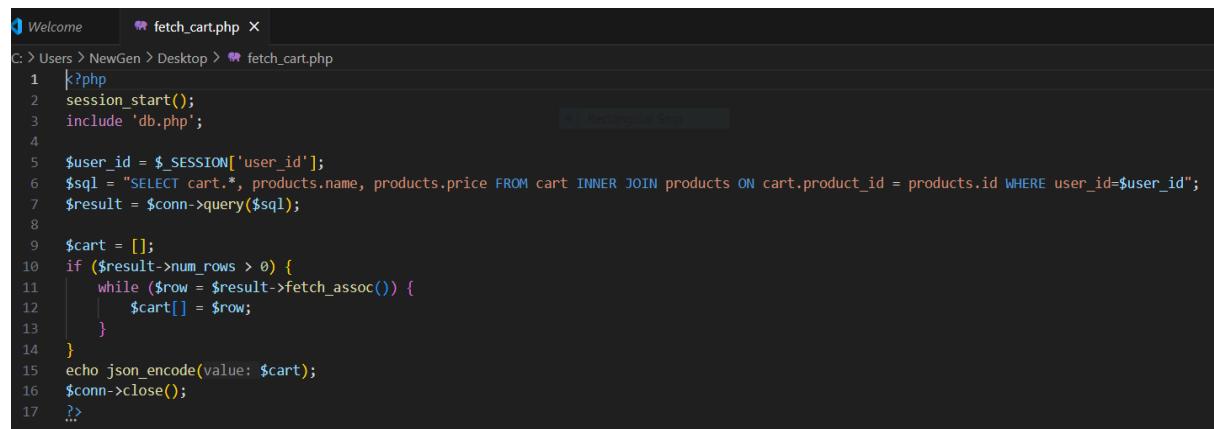
JAVA SCRIPT FETCH CART ITEMS

```
// Fetch Cart Items
async function fetchCart() {
  const response = await fetch('fetch_cart.php');
  const cartItems = await response.json();
  const cartContainer = document.getElementById('cartItems');
  cartContainer.innerHTML = '';

  let total = 0;
  cartItems.forEach(item => {
    total += item.price * item.quantity;
    cartContainer.innerHTML += `
      <div>
        <h3>${item.name}</h3>
        <p>Price: ${item.price} x ${item.quantity}</p>
      </div>
    `;
  });

  document.getElementById('totalPrice').innerText = "Total: ${total.toFixed(2)}";
}
```

PHP FOR FETCH CART



The screenshot shows a code editor window with the following details:

- Title Bar:** Welcome | fetch_cart.php x
- File Path:** C:\> Users > NewGen > Desktop > fetch_cart.php
- Code Content:**

```
1  ?>php
2  session_start();
3  include 'db.php';
4
5  $user_id = $_SESSION['user_id'];
6  $sql = "SELECT cart.*, products.name, products.price FROM cart INNER JOIN products ON cart.product_id = products.id WHERE user_id=$user_id";
7  $result = $conn-&gt;query($sql);
8
9  $cart = [];
10 if ($result-&gt;num_rows &gt; 0) {
11   while ($row = $result-&gt;fetch_assoc()) {
12     $cart[] = $row;
13   }
14 }
15 echo json_encode(value: $cart);
16 $conn-&gt;close();
17 ?&gt;</pre

```
- Toolbars/Status Bar:** Rectangular Snip

JAVA SCRIPT FOR FETCH PRODUCT

```
Fetch Products from Backend
async function fetchProducts() {
    const response = await fetch('fetch_products.php');
    const products = await response.json();
    const productList = document.getElementById('productList');
    productList.innerHTML = '';

    products.forEach(product => {
        productList.innerHTML += `
            <div class="product-card">
                
                <h3>${product.name}</h3>
                <p>${product.description}</p>
                <p><strong>Price: ${product.price}</strong></p>
                <button onclick="addToCart(${product.id})">Add to Cart</button>
            </div>
        `;
    });
}
```

FETCH PRODUCT PHP

```
Welcome          fetch_products.php X
C: > Users > NewGen > Desktop > fetch_products.php
1  <?php
2  include 'db.php';
3
4  $sql = "SELECT * FROM products";
5  $result = $conn->query($sql);
6
7  $products = [];
8  if ($result->num_rows > 0) {
9      while ($row = $result->fetch_assoc()) {
10         $products[] = $row;
11     }
12 }
13 echo json_encode($products);
14 $conn->close();
15 ?>
```

```

1  <?php
2  include 'db.php';
3
4  $sql = "SELECT * FROM products";
5  $result = $conn->query(query: $sql);
6
7  $products = [];
8  if ($result->num_rows > 0) {
9      while ($row = $result->fetch_assoc()) {
10         $products[] = $row;
11     }
12 }
13 echo json_encode(value: $products);
14 $conn->close();
15 ?>

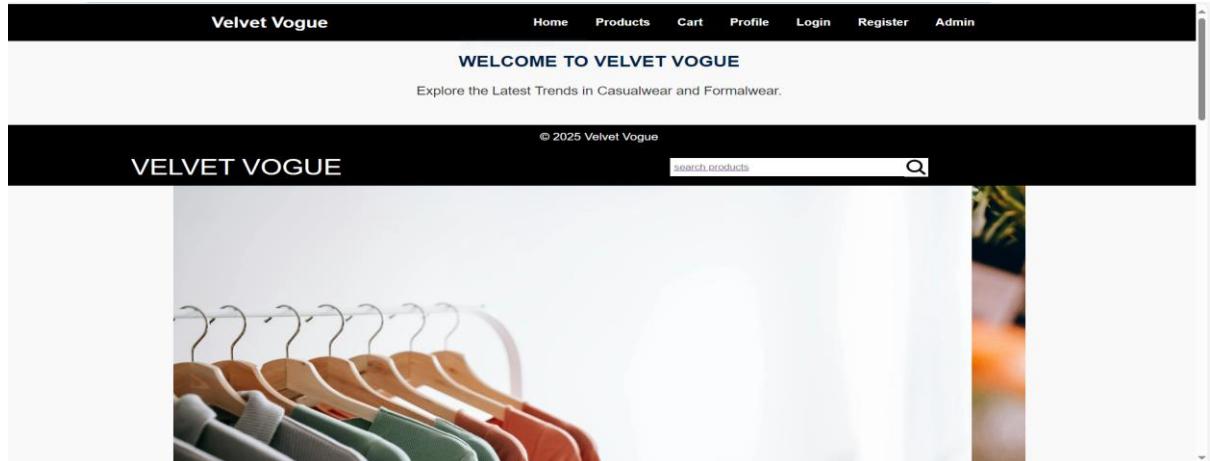
```

```

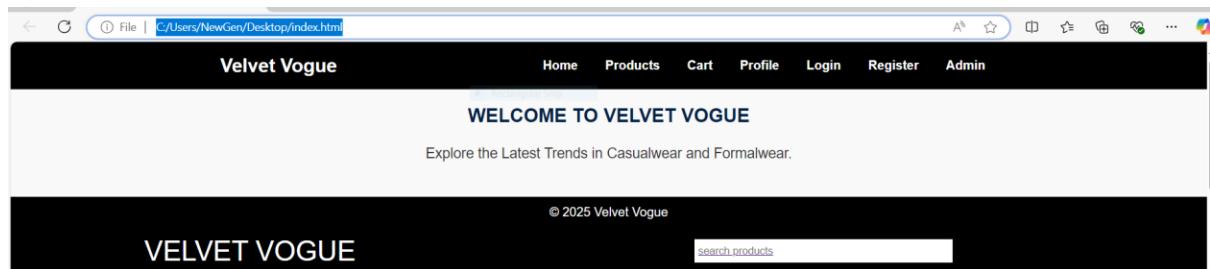
xamppe� mades� ~ /recten
1  <?php
2  session_start();
3  include 'db.php';
4
5  // Validate session
6  if (!isset($_SESSION['user_id']) || !is_numeric(value: $_SESSION['user_id'])) {
7      echo json_encode(value: ["error" => "Invalid session or user not logged in."]);
8      exit;
9  }
10
11 // Safely get the user ID
12 $user_id = intval(value: $_SESSION['user_id']);
13
14 // Use a prepared statement for secure querying
15 $stmt = $conn->prepare(
16     SELECT cart.*, products.name AS product_name, products.price AS product_price
17     FROM cart
18     INNER JOIN products ON cart.product_id = products.id
19     WHERE cart.user_id = ?
20 );
21 $stmt->bind_param("i", $user_id);
22 $stmt->execute();
23
24 $result = $stmt->get_result();
25
26 $cart = [];
27 while ($row = $result->fetch_assoc()) {
28     $cart[] = $row;
29 }
30
31 // Send the cart data as a JSON response
32 echo json_encode(value: $cart);
33
34 // Close statement and connection
35 $stmt->close();
36 $conn->close();
37 ?>

```

HOME PAGE OF VELVET VOGUE



HOME PAGE CATEGORY



HOME PAGE CSS

```
<style>
+ [<]
margin: 0;
padding: 0;
box-sizing: border-box;
<]
html {
scroll-behavior: smooth;
}

header {
display: flex;
justify-content: space-between;
align-items: center;
height: 60px;
width: 100%;
background: #black;
}
.heading ul {
display: flex;
}
.logo a {
color: #white;
transition-duration: 1s;
font-weight: 800;
}
.logo a:hover {
color: #rgb(240, 197, 6);
transition-duration: 1s;
}
.heading ul li {
list-style: none;
}
.heading ul li a {
margin: 5px;
text-decoration: none;
color: #black;
font-weight: 500;
position: relative;
color: #white;
margin: 2px 14px;
font-size: 18px;
transition-duration: 1s;
}
.heading ul li a:active {
color: #red;
}
.heading ul li a:hover {
color: #rgb(243, 168, 7);
transition-duration: 1s;
}
```

```
172 }
173 }
174 .img-slider {
175   display: flex;
176   float: left;
177   position: relative;
178   width: 1080px;
179   height: 720px;
180   animation-name: slide;
181   animation-duration: 10s;
182   animation-iteration-count: infinite;
183   transition-duration: 2s;
184 }
185
186 .heading1 {
187   opacity: 0;
188 }
189 .search {
190   display: flex;
191   position: relative;
192 }
193 .section1 {
194   width: 1080px;
195   overflow: hidden;
196
197   justify-content: center;
198   align-items: center;
199   margin: 0px auto;
200 }
201
202 .section2 .container {
203   display: flex;
204   width: 100%;
205   height: max-content;
206   flex-wrap: wrap;
207   justify-content: center;
208   margin: 10px auto;
209 }
210 .section2 .container .items {
211   margin: 10px;
212   width: 200px;
213   height: 300px;
214   background-color: #white;
215   border: 2.5px solid #black;
216   border-radius: 12px;
217 }
218 .section2 .container .items .name {
219   text-align: center;
220   background-color: #rgb(240, 197, 6);
221   height: 25px;
222   padding-top: 4px;
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1095
1096
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1107
1108
1109
1110
1111
1112
1113
1114
1115
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1144
1145
1146
1147
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1155
1156
1157
1158
1158
1159
1160
1161
1162
1163
1163
1164
1165
1166
1166
1167
1168
1168
1169
1170
1171
1171
1172
1173
1173
1174
1175
1175
1176
1177
1177
1178
1179
1179
1180
1181
1181
1182
1183
1183
1184
1185
1185
1186
1187
1187
1188
1189
1189
1190
1191
1191
1192
1193
1193
1194
1195
1195
1196
1197
1197
1198
1199
1199
1200
1201
1201
1202
1203
1203
1204
1205
1205
1206
1207
1207
1208
1209
1209
1210
1211
1211
1212
1213
1213
1214
1215
1215
1216
1217
1217
1218
1219
1219
1220
1221
1221
1222
1223
1223
1224
1225
1225
1226
1227
1227
1228
1229
1229
1230
1231
1231
1232
1233
1233
1234
1235
1235
1236
1237
1237
1238
1239
1239
1240
1241
1241
1242
1243
1243
1244
1245
1245
1246
1247
1247
1248
1249
1249
1250
1251
1251
1252
1253
1253
1254
1255
1255
1256
1257
1257
1258
1259
1259
1260
1261
1261
1262
1263
1263
1264
1265
1265
1266
1267
1267
1268
1269
1269
1270
1271
1271
1272
1273
1273
1274
1275
1275
1276
1277
1277
1278
1279
1279
1280
1281
1281
1282
1283
1283
1284
1285
1285
1286
1287
1287
1288
1289
1289
1290
1291
1291
1292
1293
1293
1294
1295
1295
1296
1297
1297
1298
1299
1299
1300
1301
1301
1302
1303
1303
1304
1305
1305
1306
1307
1307
1308
1309
1309
1310
1311
1311
1312
1313
1313
1314
1315
1315
1316
1317
1317
1318
1319
1319
1320
1321
1321
1322
1323
1323
1324
1325
1325
1326
1327
1327
1328
1329
1329
1330
1331
1331
1332
1333
1333
1334
1335
1335
1336
1337
1337
1338
1339
1339
1340
1341
1341
1342
1343
1343
1344
1345
1345
1346
1347
1347
1348
1349
1349
1350
1351
1351
1352
1353
1353
1354
1355
1355
1356
1357
1357
1358
1359
1359
1360
1361
1361
1362
1363
1363
1364
1365
1365
1366
1367
1367
1368
1369
1369
1370
1371
1371
1372
1373
1373
1374
1375
1375
1376
1377
1377
1378
1379
1379
1380
1381
1381
1382
1383
1383
1384
1385
1385
1386
1387
1387
1388
1389
1389
1390
1391
1391
1392
1393
1393
1394
1395
1395
1396
1397
1397
1398
1399
1399
1400
1401
1401
1402
1403
1403
1404
1405
1405
1406
1407
1407
1408
1409
1409
1410
1411
1411
1412
1413
1413
1414
1415
1415
1416
1417
1417
1418
1419
1419
1420
1421
1421
1422
1423
1423
1424
1425
1425
1426
1427
1427
1428
1429
1429
1430
1431
1431
1432
1433
1433
1434
1435
1435
1436
1437
1437
1438
1439
1439
1440
1441
1441
1442
1443
1443
1444
1445
1445
1446
1447
1447
1448
1449
1449
1450
1451
1451
1452
1453
1453
1454
1455
1455
1456
1457
1457
1458
1459
1459
1460
1461
1461
1462
1463
1463
1464
1465
1465
1466
1467
1467
1468
1469
1469
1470
1471
1471
1472
1473
1473
1474
1475
1475
1476
1477
1477
1478
1479
1479
1480
1481
1481
1482
1483
1483
1484
1485
1485
1486
1487
1487
1488
1489
1489
1490
1491
1491
1492
1493
1493
1494
1495
1495
1496
1497
1497
1498
1499
1499
1500
1501
1501
1502
1503
1503
1504
1505
1505
1506
1507
1507
1508
1509
1509
1510
1511
1511
1512
1513
1513
1514
1515
1515
1516
1517
1517
1518
1519
1519
1520
1521
1521
1522
1523
1523
1524
1525
1525
1526
1527
1527
1528
1529
1529
1530
1531
1531
1532
1533
1533
1534
1535
1535
1536
1537
1537
1538
1539
1539
1540
1541
1541
1542
1543
1543
1544
1545
1545
1546
1547
1547
1548
1549
1549
1550
1551
1551
1552
1553
1553
1554
1555
1555
1556
1557
1557
1558
1559
1559
1560
1561
1561
1562
1563
1563
1564
1565
1565
1566
1567
1567
1568
1569
1569
1570
1571
1571
1572
1573
1573
1574
1575
1575
1576
1577
1577
1578
1579
1579
1580
1581
1581
1582
1583
1583
1584
1585
1585
1586
1587
1587
1588
1589
1589
1590
1591
1591
1592
1593
1593
1594
1595
1595
1596
1597
1597
1598
1599
1599
1600
1601
1601
1602
1603
1603
1604
1605
1605
1606
1607
1607
1608
1609
1609
1610
1611
1611
1612
1613
1613
1614
1615
1615
1616
1617
1617
1618
1619
1619
1620
1621
1621
1622
1623
1623
1624
1625
1625
1626
1627
1627
1628
1629
1629
1630
1631
1631
1632
1633
1633
1634
1635
1635
1636
1637
1637
1638
1639
1639
1640
1641
1641
1642
1643
1643
1644
1645
1645
1646
1647
1647
1648
1649
1649
1650
1651
1651
1652
1653
1653
1654
1655
1655
1656
1657
1657
1658
1659
1659
1660
1661
1661
1662
1663
1663
1664
1665
1665
1666
1667
1667
1668
1669
1669
1670
1671
1671
1672
1673
1673
1674
1675
1675
1676
1677
1677
1678
1679
1679
1680
1681
1681
1682
1683
1683
1684
1685
1685
1686
1687
1687
1688
1689
1689
1690
1691
1691
1692
1693
1693
1694
1695
1695
1696
1697
1697
1698
1699
1699
1700
1701
1701
1702
1703
1703
1704
1705
1705
1706
1707
1707
1708
1709
1709
1710
1711
1711
1712
1713
1713
1714
1715
1715
1716
1717
1717
1718
1719
1719
1720
1721
1721
1722
1723
1723
1724
1725
1725
1726
1727
1727
1728
1729
1729
1730
1731
1731
1732
1733
1733
1734
1735
1735
1736
1737
1737
1738
1739
1739
1740
1741
1741
1742
1743
1743
1744
1745
1745
1746
1747
1747
1748
1749
1749
1750
1751
1751
1752
1753
1753
1754
1755
1755
1756
1757
1757
1758
1759
1759
1760
1761
1761
1762
1763
1763
1764
1765
1765
1766
1767
1767
1768
1769
1769
1770
1771
1771
1772
1773
1773
1774
1775
1775
1776
1777
1777
1778
1779
1779
1780
1781
1781
1782
1783
1783
1784
1785
1785
1786
1787
1787
1788
1789
1789
1790
1791
1791
1792
1793
1793
1794
1795
1795
1796
1797
1797
1798
1799
1799
1800
1801
1801
1802
1803
1803
1804
1805
1805
1806
1807
1807
1808
1809
1809
1810
1811
1811
1812
1813
1813
1814
1815
1815
1816
1817
1817
1818
1819
1819
1820
1821
1821
1822
1823
1823
1824
1825
1825
1826
1827
1827
1828
1829
1829
1830
1831
1831
1832
1833
1833
1834
1835
1835
1836
1837
1837
1838
1839
1839
1840
1841
1841
1842
1843
1843
1844
1845
1845
1846
1847
1847
1848
1849
1849
1850
1851
1851
1852
1853
1853
1854
1855
1855
1856
1857
1857
1858
1859
1859
1860
1861
1861
1862
1863
1863
1864
1865
1865
1866
1867
1867
1868
1869
1869
1870
1871
1871
1872
1873
1873
1874
1875
1875
1876
1877
1877
1878
1879
1879
1880
1881
1881
1882
1883
1883
1884
1885
1885
1886
1887
1887
1888
1889
1889
1890
1891
1891
1892
1893
1893
1894
1895
1895
1896
1897
1897
1898
1899
1899
1900
1901
1901
1902
1903
1903
1904
1905
1905
1906
1907
1907
1908
1909
1909
1910
1911
1911
1912
1913
1913
1914
1915
1915
1916
1917
1917
1918
1919
1919
1920
1
```

HOME PAGE HTML

```
index.html > + num > + body > + header > + nav > + u
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Velvet Vogue</title>
7       <link rel="stylesheet" href="styles.css">
8     </head>
9     <body>
10       <header>
11         <h1>Velvet Vogue</h1>
12         <nav>
13           <a href="index.html">Home</a>
14           <a href="products.html">Products</a>
15           <a href="cart.html">Cart</a>
16           <a href="profile.html">Profile</a>
17           <a href="Register.html">Register</a>
18           <a href="Login.html">Login</a>
19           <a href="Admin.html">Admin</a>
20         </nav>
21       </header>
22       <main>
23         <h2>Welcome to Velvet Vogue</h2>
24         <p>Explore the latest trends in casualwear and formalwear.</p>
25       </main>
26       <footer>
27         <p>&copy; 2025 Velvet Vogue</p>
28       </footer>
29     </body>
30   </html>
31   <!DOCTYPE html>
32   <html>
33     <head>
34
```

```
</style>
<link href="https://unpkg.com/ionicons@4.5.10-0/dist/css/ionicons.min.css" rel="stylesheet">
</head>

<body>
<header>
<div class="logo"><a href="#">VELVET VOGUE</a></div>
<div class="menu">
<a href="#">
<ion-icon name="close" class="close"></ion-icon>
</a>

<ul>
<li><a href="#" class="under">HOME</a></li>
<li><a href="#" class="under">SHOP</a></li>
<li><a href="#" class="under">OUR PRODUCTS</a></li>
<li><a href="#" class="under">CONTACT US</a></li>
<li><a href="#" class="under">ABOUT US</a></li>
<li><a href="#" class="under">LOGIN</a></li>
<li><a href="#" class="under">REGISTER</a></li>
<li><a href="#" class="under">ADMIN LOGIN</a></li>
</ul>
</div>
<div class="search">

<a href=""><input type="text" placeholder="search products" id="input">
<ion-icon class="s" name="search"></ion-icon>
</a>
</div>

<div class="heading1">
<ion-icon name="menu" class="ham"></ion-icon>
</div>
</header>
<section>
<div class="section">
<div class="section1">
<div class="img-slider">





</div>
</div>
</div>
</section>
<!DOCTYPE html>
<html lang="en">
<head>
|   <meta charset="UTF-8">
```

```

<div id="section2" class="section2">
<div class="container">
<div class="items">
<div class="img1"></div>
<div class="name">SHOES</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$5</div>
<div class="info">Lorem ipsum dolor sit amet consectetur.</div>
</div>
<div class="items">
<div class="img2"></div>
<div class="name">MEN's T-SHIRT</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$6.34</div>
<div class="info">Lorem ipsum dolor sit amet.</div>
</div>
<div class="items">
<div class="img3"></div>
<div class="name">JEANS</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$9</div>
<div class="info">Lorem ipsum dolor sit amet.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">WATCH</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$9.1</div>
<div class="info">Lorem ipsum dolor sit amet.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">ACCESSORIES</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$20</div>
<div class="info">Lorem ipsum dolor sit amet.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">BOY JEANS</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$18</div>
<div class="info">Lorem ipsum dolor sit amet consectetur.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">HOODIES</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$6.7</div>
<div class="info">Lorem ipsum dolor sit amet.</div>
</div>

```

```

<div id="section2" class="section2">
<div class="container">
<div class="items">
<div class="img1"></div>
<div class="name">BABY DRESSES</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$18</div>
<div class="info">Lorem ipsum dolor sit amet consectetur adipisicing elit.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">BABY DRESSES</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$10</div>
<div class="info">Lorem ipsum dolor sit amet consectetur adipisicing elit.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">BLANKETS</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$9.9</div>
<div class="info">Lorem ipsum dolor sit amet consectetur adipisicing.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">SHIRTS</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$9</div>
<div class="info">Lorem ipsum dolor sit amet consectetur adipisicing elit.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">SHORTS</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$3</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">ACCESSORIES</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$29.7</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">BED</div>

```

```
<div id="footer" class="footer0">
<h1>CODE WITH AASHIK</h1>
</div>
<div class="footer1 ">
Connect with us at<div class="social-media">
<a href="#">
<ion-icon name="logo-facebook"></ion-icon>
</a>
<a href="#">
<ion-icon name="logo-linkedin"></ion-icon>
</a>
<a href="#">
<ion-icon name="logo-youtube"></ion-icon>
</a>
<a href="#">
<ion-icon name="logo-instagram"></ion-icon>
</a>
<a href="#">
<ion-icon name="logo-twitter"></ion-icon>
</a>
</div>
</div>
<div class="footer2">
<div class="product">
<div class="heading">Products</div>
<div class="div">Sell your Products</div>
<div class="div">Advertise</div>
<div class="div">Pricing</div>
<div class="div">Product Buisness</div>
</div>
<div class="services">
<div class="heading">Services</div>
<div class="div">Return</div>
<div class="div">Cash Back</div>
<div class="div">Affiliate Marketing</div>
<div class="div">Others</div>
</div>
<div class="Company">
<div class="heading">Company</div>
```

```
0 <div class="get-help">
1   <div class="heading">Get Help</div>
2   <div class="div">Help Center</div>
3   <div class="div">Privacy Policy</div>
4   <div class="div">Terms</div>
5   <div class="div">Login</div>
6 </div>
7 </div>
8 <div class="footer3">CONTACT DETAILES:-0760647164</div>
9 <div class="footer3">NAME:-M.M.M AASHIK</div>
10 <div class="footer3">Copyright © <h4>VELVET VOGUE</h4> 2025-2030</div>
11 </footer>
12 <script>
13   const close = document.querySelector(".close");
14   const open = document.querySelector(".ham");
15   const menu = document.querySelector(".menu");
16   close.addEventListener("click", () => {
17     menu.style.visibility = "hidden";
18   });
19   open.addEventListener("click", () => {
20     menu.style.visibility = "visible";
21   });
22 </script>
23
24 </body>
25
26 </html>
```

HOME PAGE CATEGORY CODING HTML

```
<button onclick="alert('button clicked!')">>ADD TO CART</button>
<div class="price">$29.7</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">BED</div>
<button onclick="alert('button clicked!')">>ADD TO CART</button>
<div class="price">$100</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">SKIRT</div>
<button onclick="alert('button clicked!')">>ADD TO CART</button>
<div class="price">$78</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">LEGGINGS</div>
<button onclick="alert('button clicked!')">>ADD TO CART</button>
<div class="price">$9</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">BAG</div>
<button onclick="alert('button clicked!')">>ADD TO CART</button>
<div class="price">$36.5</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">SAREES</div>
<button onclick="alert('button clicked!')">>ADD TO CART</button>
<div class="price">$25.6</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
<div class="items">
<div class="img1"></div>
<div class="name">JEANS</div>
<button onclick="alert('button clicked!')">>ADD TO CART</button>
<div class="price">$5</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
```

```
<div class="filter-bar">
  <label for="category">Category:</label>
  <select id="category">
    <option value="all">All</option>
    <option value="men">Men's Wear</option>
    <option value="women">Women's Wear</option>
    <option value="accessories">Accessories</option>
  </select>
</div>

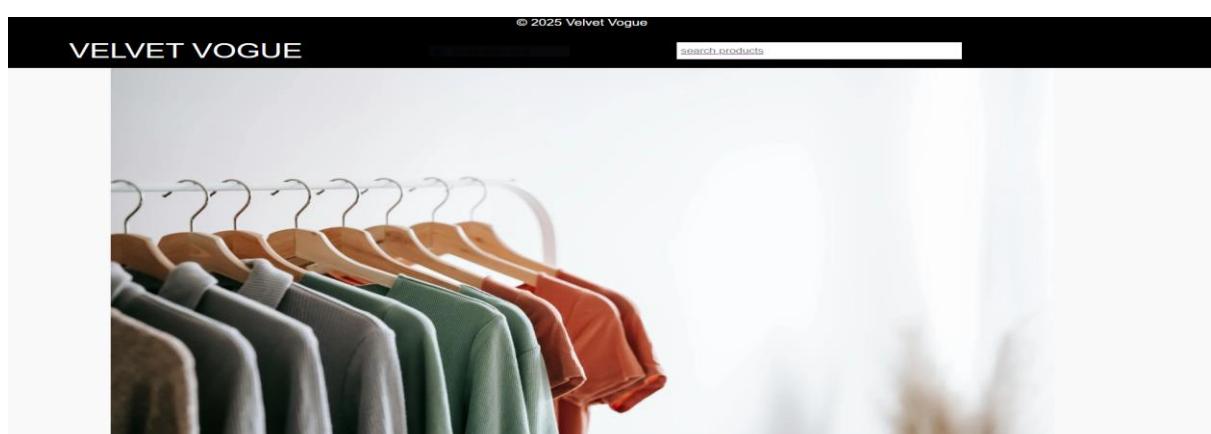
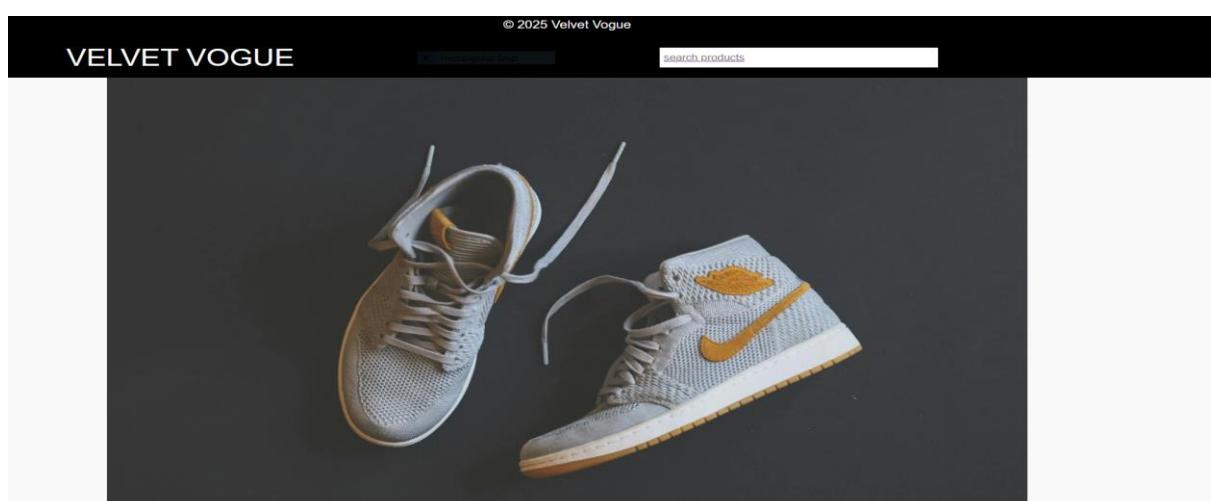
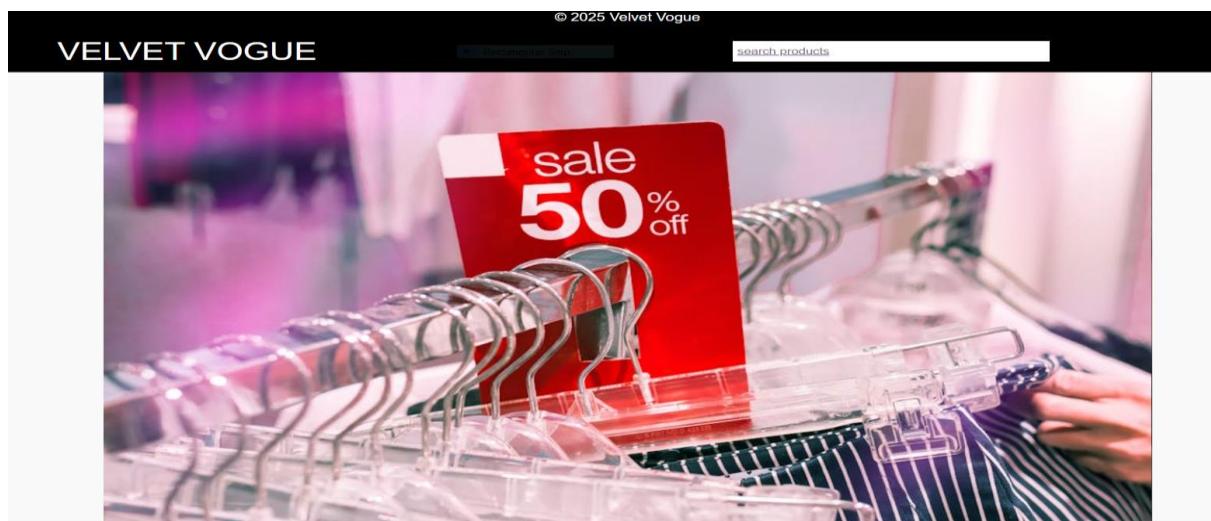
<div class="item-list">
  <div class="item men">Men's T-Shirt</div>
  <div class="item men">SHOES</div>
  <div class="item men">BOY_JEANES</div>
  <div class="item women">Women's Dress</div>
  <div class="item accessories">Watch</div>
  <div class="item men">Men's Jacket</div>
  <div class="item women">Women's Scarf</div>
  <div class="item women">JEANES</div>
  <div class="item All">HOODIES</div>
  <div class="item women">BAG</div>
  <div class="item accessories">Sunglasses</div>
</div>

<script>
  document.getElementById('category').addEventListener('change', function () {
    const selectedCategory = this.value;
    const items = document.querySelectorAll('.item');

    items.forEach(item => {
      if (selectedCategory === 'all' || item.classList.contains(selectedCategory)) {
        item.classList.add('visible');
      } else {
        item.classList.remove('visible');
      }
    });
  });

  // Initialize to show all items on page load
  document.getElementById('category').dispatchEvent(new Event('change'));
</script>
</body>
```

FRONT END HOME PAGE OFFER SECTION



PRODUCT



HOODIES
[ADD TO CART](#)

\$6.7

Lorem ipsum dolor sit.

```
<div class="items">
<div class="img img1"></div>
<div class="name">HOODIES</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$6.7</div>
<div class="info">Lorem ipsum dolor sit.</div>
</div>
```



LEGGINGS
[ADD TO CART](#)

\$9

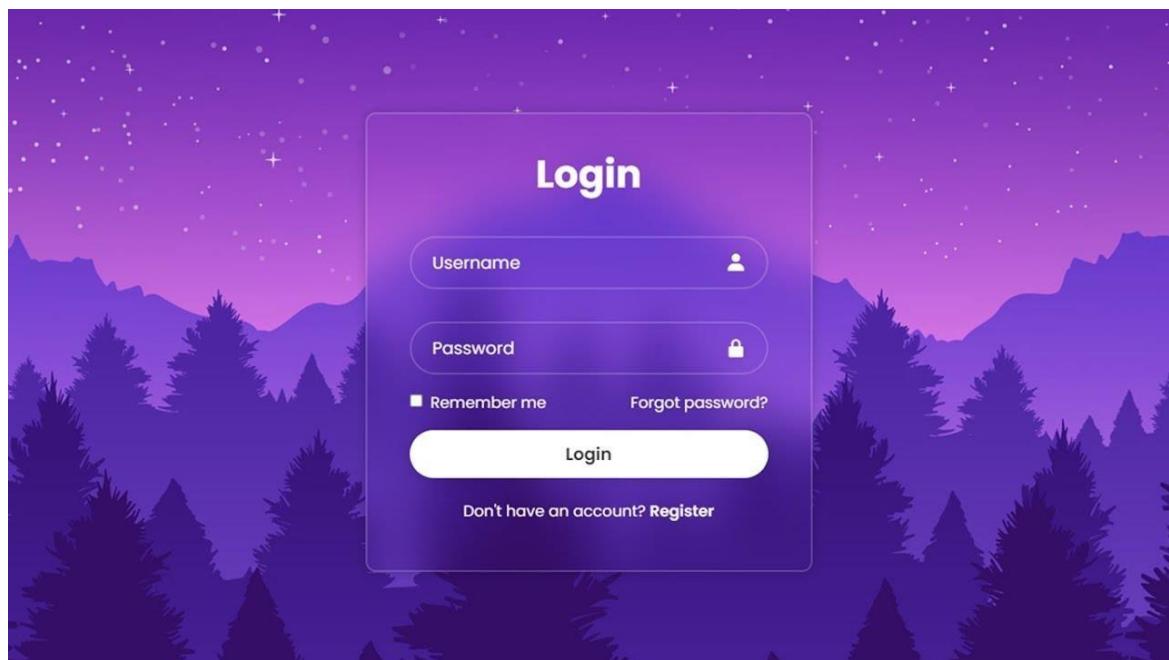
Lorem ipsum dolor sit amet consectetur, adipisicing elit.

```
<div class="items">
<div class="img img1"></div>
<div class="name">LEGGINGS</div>
<button onclick="alert('button clicked!')">ADD TO CART</button>
<div class="price">$9</div>
<div class="info">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</div>
</div>
```

Evaluation of the Design and Development Process against the Design Document

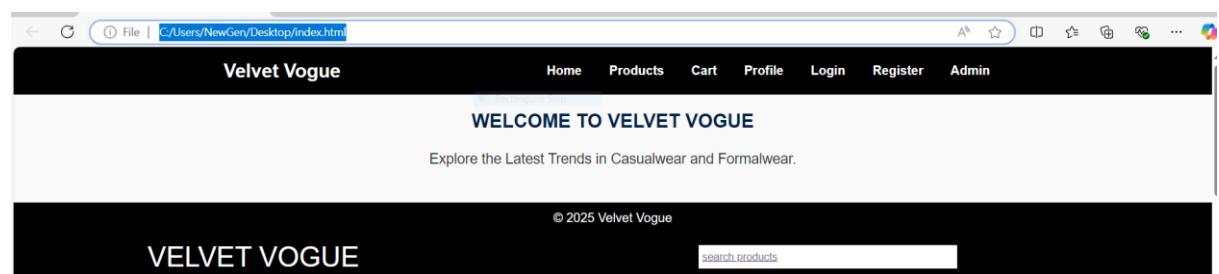
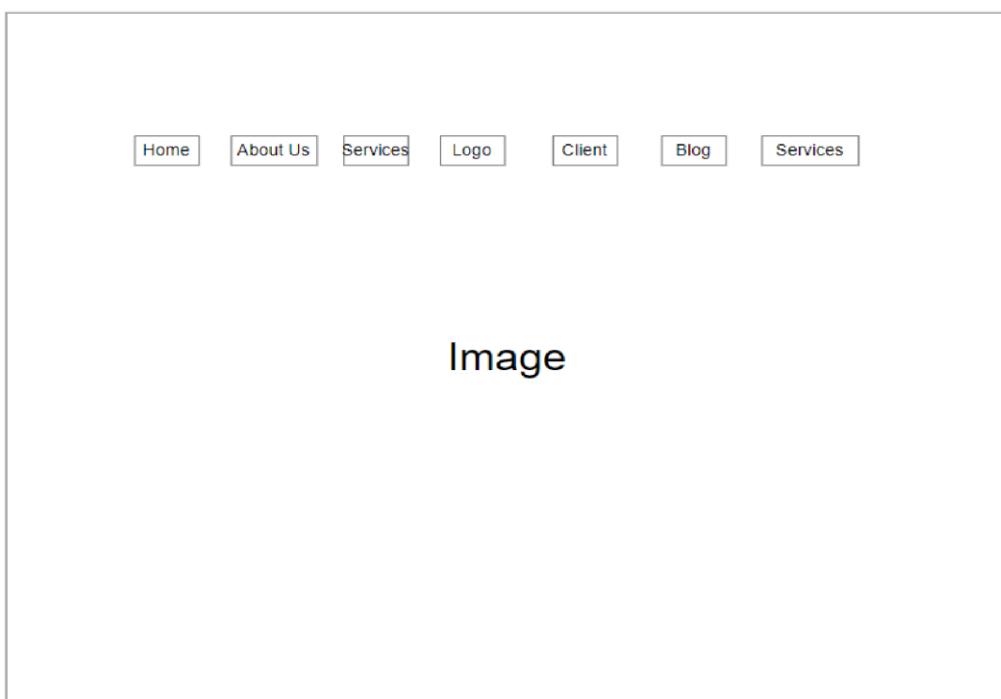
As we saw in the above images it's clear that some changes has taken place between the wireframes and the final design in the webpage. There were several reasons for it and some of them even affected the functionality of the website. Some colors were changed in between them and some text were different. However the changes made in the final design was due to some unavoidable reasons. The following are the changes and the similarities found in the wireframes and the final design.

Login

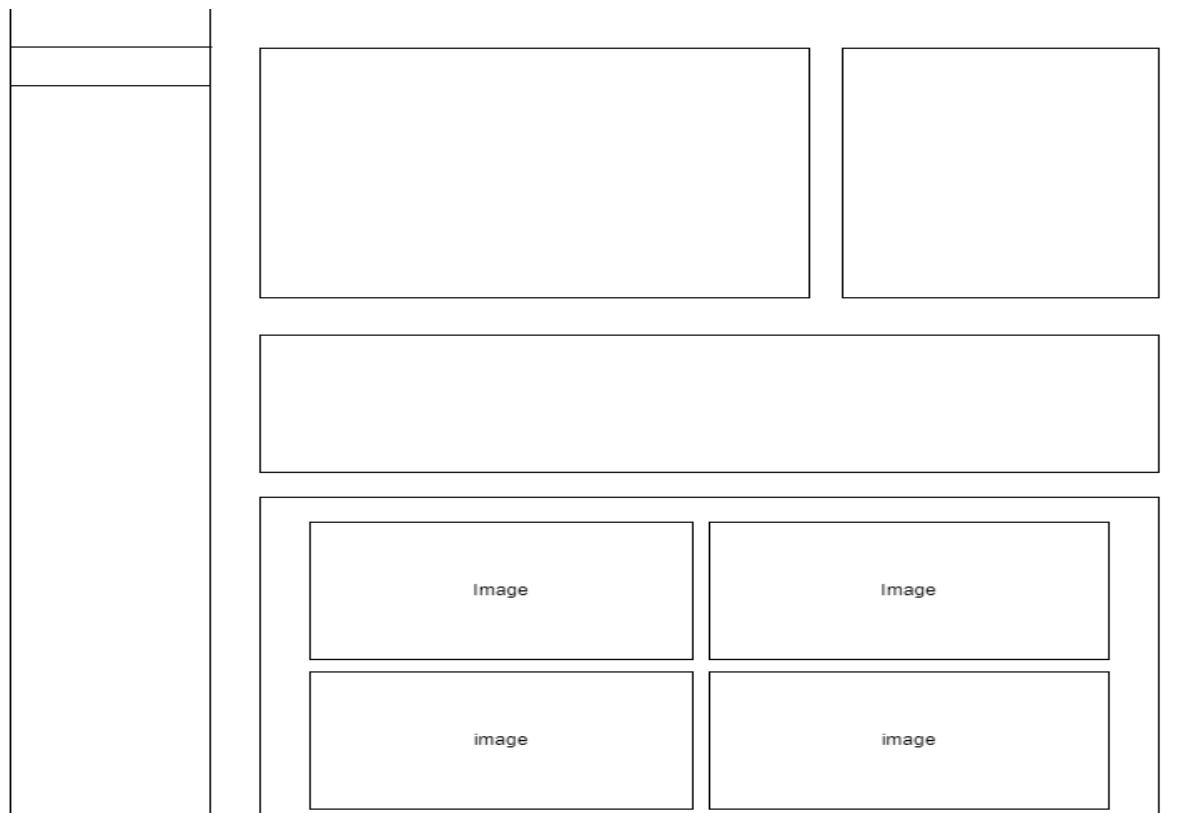


The above two images depict the wireframe and the final image of the login page in the web app. The changes in both these images are obvious and clear. The main difference between these two is that the final product was made inside a form for more highlight. The colors used in the background are darker and the form was made in white. The Login button is in black in the final product and not in the wireframe. These changes mainly occurred when the homepage was created. The color theme I used in my system was blue and the color I used in the wireframe was not suitable in the final product. However other than that there are no changes and the functionalities of the login is working perfectly. Simplicity was another key feature in our project and we had to keep it easy and simple for the users to use it

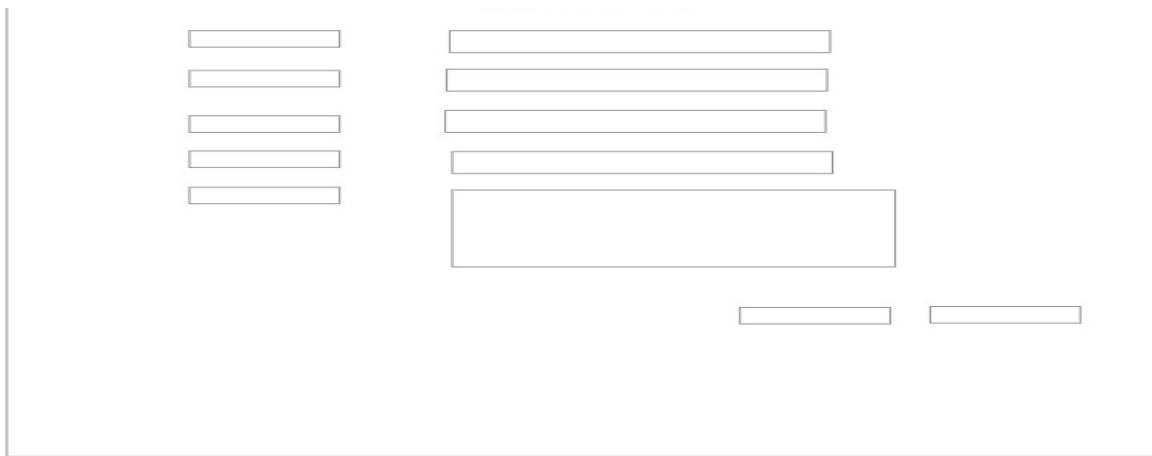
Home page



Product page



Register Form Wireframe



Register Form Front end

Your Name	:	Customer Name
Email Address	:	Mail@example.com
Bill Number	:	Bill Number
Phone Number	:	Your Phone Number
Customer Enquiry	:	Your Queries

SUBMIT **RESET**

Database Connection

Prior to establish the functionalities, a connection has to be established with the database. This can be provided in every page or a single file can be created and used in every other

page by using the include command. I used the second option here and created a different PHP file with the connection code and included it in all the pages where PHP database was used. The following images are examples of it.

```
C:\xampp>htdocs>PROJECT> db.php
1 <?php
2 $servername = "localhost";
3 $username = "root";
4 $password = "";
5 $dbname = "velvet_vogue";
6
7 $conn = new mysqli($servername, $username, $password, $dbname);
8
9 if ($conn->connect_error) {
10 | die("Connection failed: " . $conn->connect_error);
11 }
12 ?>
```

This PHP code is responsible for establishing a connection between a website and a database. It begins by including a file named 'database.php,' suggesting that essential database configurations are stored there. The code then specifies the server's name (localhost, which usually means the database is on the same server as the website), the username, password, and the name of the specific database.

After gathering these details, the code attempts to connect to the database using the 'mysqli_connect' function. If the connection is successful, it means the website can now communicate with the database. However, if there's an issue, such as incorrect credentials or database unavailability, the code stops and displays an error message, indicating that the connection has failed. This connection is crucial for the website to interact with and retrieve data from the specified database.

Login form Php code

```

<?php
    include('database.php');
    if(isset($_POST['submit'])){
        $username = $_POST['username'];
        $password = $_POST['password'];

        $query = mysqli_query($con, "INSERT INTO Login(Username, Password) VALUES ('$username', '$password')");

        if($query){
            echo "<script>alert('Data Inserted Successfully');</script>";
        } else {
            echo "Query error: " . mysqli_error($con);
        }
    }
?>

```

This PHP code is designed to handle user login information. When a form with a submit button is submitted, the code retrieves the entered username and password. It then connects to a database using the 'database.php' file, and attempts to insert the username and password into a table named 'Login'. If the insertion is successful, it shows a message saying 'Data Inserted Successfully' using a pop-up alert. If there's an issue with the insertion, it displays an error message with details about the problem. It's important to note that this code could be vulnerable to security risks like SQL injection, and for real-world applications, additional security measures should be implemented.

localhost says
Data inserted Successfully

OK

When a user logs in to the website, the server-side code processes the form data, and if everything goes well with the database insertion, this success message is often shown to the user.

Login Database

username	password
ahmed	rgthytr436
Aamir	aamir456234
Malcomlismore	malcom@4546
ahmed	ahmed4562@45
Rimas	Rimas@3455
Abdhullah	abdhullah45@1
Shaheem	Shaheem3@567

Show all | Number of rows: 25

Technical challenges faced During development

Responsive Design:

Designing a website to be visually appealing and functional across various devices and screen sizes requires careful consideration. Implementing responsive design principles was challenging during the development process.

Limited knowledge of Creator:

Developers encountered challenges when translating wireframe features into code, especially when the team had a limited understanding of certain aspects. For instance, a graph intended for the dashboard had to be changed to a bar chart due to a lack of expertise.

Limited Time

The development was ended with basic design due to lack of time and more time needed for well structured website,. This highlights the need for effective time management and prioritization.

Navigational Difficulties:

Inadequate validation of interface navigations during the design phase created a burden emphasizing the commitment to maintaining proper navigations throughout development to enhance user experiences.

Neglect of User Experience in Wireframes:

Initially overlooking user experience in wireframes posed challenges during development. To address this, developers modified color codes, introduced light backgrounds, and strategically employed vivid colors. This adjustment made the development process more challenging but improved the overall user experience.

Syntax problems

Syntax problems develop when code deviates from a computer language's established conventions. Identifying and resolving these issues can be difficult, especially for inexperienced developers. Missing or misplaced brackets, semicolons, and quotation marks are common causes of syntax errors.

```
<?php  
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_FILES["image"])) {  
    $name = $_FILES["image"]["name"]  
    $data = file_get_contents($_FILES["image"]["tmp_name"]);  
  
    $dbserver = "localhost";
```

PHP issues can appear in a variety of ways on website, including syntax errors, fatal errors, warnings, and alarms. These difficulties might occur as a result of errors in the PHP code or server configuration settings. For example, problems connecting to the PHP server, such as issues with the root or password settings, might impair website operation

While connecting to a remote database server, authentication issues may arise, where wrong root or password settings may prohibit the website from accessing the database, resulting in failures. Additionally, database connections may fail owing to issues with database configuration, network connectivity, or permissions. Debugging tools are required for finding and resolving these difficulties.



When developing code to insert or change data in a database, problems might emerge. Mistakes might occur due to limits in database tables, errors in SQL syntax, or data type mismatch. Thorough testing and the inclusion of error-handling procedures are required to properly identify and address these difficulties.

Another potential issue is a primary key violation, which occurs when attempting to insert a record that violates the primary key requirement. This might happen if you try to input a primary key value twice or if you insert a null value into a column with a primary key that cannot be null.



The developer must solve coding, design, functionality, and performance challenges to guarantee a stable and secure website that incorporates HTML, PHP, CSS, and JavaScript. It is critical to use tools such as debugging, logging, and testing tools. It is critical to validate database credentials and ensure correct server configuration, as well as to build strong error handling and logging mechanisms. To safeguard the data on the website and prevent unwanted access, security measures such as firewalls and encryption must be installed. Regular testing and patches are required to keep a website secure and reliable.

Test Plan for Velvet Vogue Multipage Website

The following test plan identifies **key performance areas** and outlines the approach to review the functionality, performance, and overall quality of the Velvet Vogue multipage e-commerce website. This structured plan ensures the website meets user and client requirements while providing a seamless and reliable user experience.

1. Test Plan Overview

Objective

M.M.M AASHIK/E230667

WDD

To validate the functionality, performance, security, and usability of the Velvet Vogue website against the design document and client requirements.

Scope

The test plan covers the following areas:

1. Functional Testing (Core Features)
2. Performance Testing
3. Security Testing
4. Usability Testing
5. Compatibility Testing

Functional Testing

Functional testing ensures that all features of the website operate as intended, providing a seamless experience to users. This includes testing functions such as browsing, searching, adding products to the cart, and completing purchases. Each feature is examined through test cases that compare expected results with actual outcomes. For example, the search function is tested to ensure accurate results based on user queries, and the checkout process is validated to confirm smooth transactions. Payment gateways, user account management, and error handling mechanisms are also scrutinized to ensure reliability and usability.

Reliability Testing

Reliability testing verifies the website's ability to perform consistently under all conditions, especially in a 24/7 operational environment. It evaluates whether the website can handle uninterrupted service, even during peak usage periods. Load balancing across multiple servers is tested to ensure even distribution of traffic, preventing downtime or slow responses. Simulated long-duration tests help identify potential weaknesses, ensuring the system is dependable for users at all times.

Stress Testing

Stress testing involves subjecting the website to extreme conditions, such as heavy traffic or massive data loads, to assess its performance and stability. By simulating peak usage scenarios, such as a flash sale or a sudden influx of users, developers can identify bottlenecks

and optimize the system. This testing ensures the website can handle unexpected spikes in activity without degrading performance or crashing.

Recovery Testing

Recovery testing examines the website's ability to recover from unexpected failures, such as server crashes or data corruption. It evaluates how quickly and effectively the system can restore normal operations after an issue. Automated backup systems, failover mechanisms, and robust error-handling processes are tested to ensure minimal disruption to users and no loss of critical data. This ensures business continuity and user trust during unexpected events.

Security Testing

Security testing focuses on protecting sensitive data and preventing unauthorized access. This includes verifying user authentication and authorization, encrypting sensitive data like passwords and payment information, and ensuring HTTPS implementation. Common vulnerabilities, such as SQL injection and cross-site scripting, are tested using tools like OWASP ZAP. Penetration tests are conducted to identify potential security gaps and safeguard the website from cyber threats.

Usability Testing

Usability testing ensures the website is intuitive and user-friendly, providing a pleasant experience for visitors. It assesses how easily users can navigate between pages, find relevant information, and complete tasks like making a purchase. Real user feedback is crucial to identify areas of confusion or frustration. Enhancing usability involves optimizing design elements, improving page layouts, and streamlining workflows to ensure an effortless experience for all users.

Performance Testing

Performance testing measures the website's speed and responsiveness under various conditions. It evaluates page load times, search response times, and transaction processing speeds. Tools like Google PageSpeed Insights can identify performance bottlenecks, enabling developers to optimize the website for faster operation. Ensuring quick responses to user actions enhances overall satisfaction and encourages repeat visits.

Platform Testing (Portability)

Platform testing ensures the website functions seamlessly across multiple devices, browsers, and operating systems. Compatibility is tested on desktops, tablets, and smartphones, as well as on browsers like Chrome, Firefox, Safari, and Edge. This guarantees a consistent user experience regardless of the device or platform being used, broadening the website's reach and accessibility.

Accessibility and Inclusivity Testing

Accessibility testing ensures the website caters to users with disabilities, promoting inclusivity. Features such as screen reader compatibility, keyboard navigation, and voice input are implemented to assist users with visual or physical impairments. High-contrast modes and proper semantic HTML improve readability and usability for visually impaired users. Testing also involves compliance with standards like WCAG 2.1 to ensure the website is accessible to all individuals, regardless of their abilities.

Responsibilities:

- **Test Manager:** Oversees the testing process and ensures all tests are completed.
- **Test Engineers:** Execute the tests and document results.

Tools:

- Selenium for automation testing.
- Lighthouse for performance and SEO analysis.
- OWASP ZAP for security testing.
- Browser Stack for cross-browser compatibility.

3. Execution Plan

Execution Timeline:

The testing will be executed in the following order over a 2-week period:

1. Functional Testing: Days 1–4
2. Usability Testing: Days 5–6
3. Compatibility Testing: Days 7–8
4. Performance Testing: Days 9–10
5. Security Testing: Days 11–12

4. Reporting and Metrics

1. **Bug Tracking:**
 - Use JIRA to log and monitor bugs identified during testing.
2. **Metrics:**
 - Test Case Coverage: Percentage of planned test cases executed.
 - Bug Fix Rate: Percentage of bugs resolved before deployment.
 - Pass Rate: Percentage of successful test cases.

5. Expected Outcomes

- **Functionality:** All core features work as intended.
- **Performance:** Website remains fast and responsive under normal and peak loads.
- **Security:** No vulnerabilities detected.
- **Usability:** Website is easy to use and accessible to a wide audience.
- **Compatibility:** Works seamlessly across all supported devices and browsers.

Test Plans

A test plan is a comprehensive document that outlines the approach, objectives, resources, schedule, and scope of testing activities for a specific software project. It serves as a roadmap for the testing process, providing a structured and organized way to ensure that software applications meet their intended requirements and function correctly.

A well-crafted test plan is instrumental in the success of a software project by providing a comprehensive roadmap for testing activities. It ensures clear communication of testing

objectives, scope, and resource requirements, facilitating efficient allocation of personnel, hardware, and software. The plan aids in risk management by identifying and assessing potential issues, allowing for proactive mitigation. Additionally, it establishes a structured schedule for testing activities, promoting effective time management. Through clear documentation of requirements, test cases, and milestones, the plan enhances traceability, ensuring that each requirement is systematically tested. Post-testing, the plan serves as a valuable tool for continuous improvement, enabling the capture of lessons learned and contributing to the refinement of future testing processes. Ultimately, a robust test plan not only ensures the reliability and quality of the software but also supports compliance with industry standards and regulatory requirements.

	Test	Input Data	Output	Expected Result
01	Enter wrong login username	Username= VELVET Password= VELVET123	Invalid username.	Can't access the Dashboard.
02	Send enquiry	Send an enquiry.	Sent successfully.	Send enquiry to admin.
03	Enter correct login username and password	Username=VELVET Password= VELVET@123	Login successfully	Navigate to the admin panel.
04	Testing the connection with database and webpage.	Database connection php code.	Connected successfully.	Connected to the database successfully.

Test Case 01

Test No : 01 Test Case : Login Functionality	Test Method :- White box testing
Input: Username= VELVET Password= VELVET123	

Input: Username= VELVET Password= VELVET123
Expected Output: Invalid username
Actual Output: Invalid username
Test Status: Success
Output
 A screenshot of a web-based sign-in interface. At the top, it says "Sign In". Below that are fields for "Your Email Address" and "Password". A modal dialog box is displayed in the center. The dialog has a header "localhost" with "Remember me" and "Forgot Password" links. The main content of the dialog says "Invalid user name". At the bottom of the dialog are "OK" and "Back to website" buttons. The background shows the rest of the sign-in page.

Test case 02

Test No : 02	Test Method :- White box testing
Test Case : Send enquiry in enquiry form	
Testing Process: Filling the enquiry form and checking whether the success message appears.	

Test Data: Data stored in the enquiry table by storing data in relevant fields

Salman	salman34@gmail.com
imadh	imadhaleem34@gmail.com
Ghanim	ganim32a2@gmail.com
Alkama	alkam672@gmail.com
Khalidh	kalid982@gmail.com

Expected Output: Data stored successfully in the enquiry table

Actual Output: Data stored in the enquiry table successfully

Test Status: Success

Output

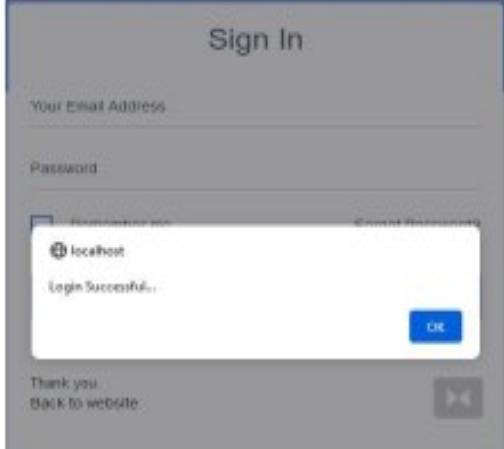
localhost/musharrif/enquiry.php

localhost says
Data inserted Successfully

OK

Test case 03

Test Case :03	Method: White Box Testing
Test	Enter Correct username and password
Expected output	Login successful
Input	Username=VELVET Password= VELVET123

Output	Invalid password 
Conclusion	Actual password match with the expected password

Test case 04

Test N 04

Test Case: Testing th connection withdatabase andwebpage.

Testing Process: Validating th connection betweenth databasean th webpage.

Test Data:

```
?php  
$con= mysqli connect "localhost" "root" "  
i ( $con)  
di "connection error" mysqli_connect_error()  
  
else  
  
echo "connected succesfully"  
  
?
```

Expected Output: Connection establishedSuccessfully

Actual Output: Connection establishedSuccessfully

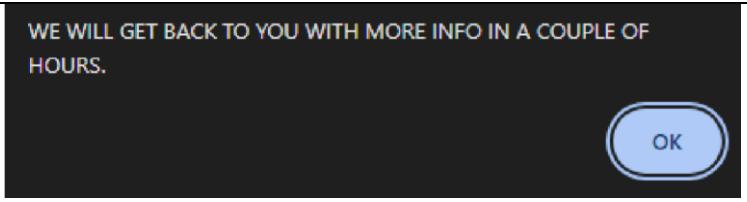
Test Status: Success

Output:

Output

```
connected succesfully
```

TEST CASE 05

Test ID	05
Test Name	Submit contact us form
Steps to Test	<p>Fill data</p> <p>Press submit button</p>
Pre-defined function	
Post-defined function	Insert
Date of testing	04.01.2025
Expected outcome	“WE WILL GET BACK TO YOU WITH MORE INFO IN A COUPLE OF HOURS.” Message box
Actual outcome	<p>WE WILL GET BACK TO YOU WITH MORE INFO IN A COUPLE OF HOURS.</p> 
Tester's Name	Mohamed Aashik

Test case 06

Test ID	06
Test Name	Submitting the form with an empty field
Steps to Test	Fill all the data to the form

	Leave only one field Submit the form
Pre-defined function	
Post-defined function	Submit contact us form
Date of testing	04.01.2025
Expected outcome	Request to fill that empty field and do not submit the form
Actual outcome	<p style="text-align: center;">Contact us</p> 
Tester's Name	Mohamed Aashik

These are the test cases and plans I included in my assignment. There are several other test cases and plans that can be created from my project but all those will contain the same functionality as I have mentioned in the above test plans. This is because my system basically uses the most common functionalities such as insert, update, delete and other functions. Since I have used these all around my system I would have to give the same test

Analyse the Quality Assurance (QA) process and review how it was implemented during the multipage website design and development stages.

Quality assurance

Quality assurance (QA) is any systematic process of determining whether a product or service meets specified requirements. QA establishes and maintains set requirements for developing or manufacturing reliable products. A quality assurance system is meant to increase customer confidence and a company's credibility, while also improving work processes and efficiency, and it enables a company to better compete with others.

QA focuses on the entire development process, from requirements gathering to testing and release. It aims to identify and address issues early in the development lifecycle.

Quality Assurance Process:

Requirements Analysis: Understand and document project requirements to establish a baseline for quality expectations.

Planning: Develop a QA plan outlining testing strategies, resources, schedules, and responsibilities.

Execution: Implement testing activities, including unit testing, integration testing, and system testing, based on the established plan.

Defect Tracking: Identify and document defects or deviations from requirements.

Reporting: Generate reports on testing progress, defect status, and overall quality metrics.

Feedback Loop: Provide feedback to developers for necessary corrections, and iterate through testing cycles. **Release:** Once QA criteria are met, release the product with confidence in its quality.

Testing

Testing, in the context of software development or website creation, refers to the systematic process of evaluating a system or application to identify defects, ensure its functionality, and verify that it meets specified requirements. The primary goal of testing is to ensure the delivery of a high-quality, reliable, and error-free product.

Key Aspects of Testing:

- **Defect Identification:** Testing aims to uncover defects or issues in the software. These defects could range from functional errors and performance issues to usability problems.
- **Functional Verification:** Testing verifies that the software functions according to the specified requirements. This involves checking whether all features work as intended and meet the user's needs.
- **Performance Assessment:** Testing evaluates the performance of the software, including factors like speed, responsiveness, and scalability. This ensures that the system can handle various scenarios and user loads.
- **Usability Testing:** This aspect focuses on the user experience, ensuring that the software is user-friendly, intuitive, and meets the expectations of its intended audience.
- **Security Testing:** Security is a critical concern, especially for web applications. Testing helps identify vulnerabilities and ensures that the system is resistant to unauthorized access and potential threats.
- **Regression Testing:** As the software evolves through updates or changes, regression testing ensures that new modifications do not adversely affect existing functionalities.
- **Documentation Verification:** Testing also involves validating that the documentation, including user manuals and technical guides, accurately represents the software's features and functionalities.

Testing is an integral part of the software development life cycle, starting from the initial design phase through development and maintenance. It helps build confidence in the reliability of the

software and contributes to the delivery of a product that aligns with user expectations and industry standards.

The Different type of QA software testing

Every software tester encounters a range of QA testing types, some of which they have worked on and others of which they have just heard of. Unfortunately, not everyone is aware of the many forms of QA testing and the importance they play in overall product quality. At a high level, testing may be classified as human or automated; however, there are various more types.

- Unit Testing Unit testing is an essential practice in software development. It entails testing isolated sections of a software application, such as functions, methods, modules, or classes. The application is made up of these distinct pieces, and if they don't perform correctly on their own, they won't work well together. Before incorporating a component into a bigger system, unit testing assures that it works properly.
- Integration Testing Integration testing is a type of software testing in which components are merged and assessed as a whole. These components have passed unit testing, indicating that they perform well on their own, but problems may develop when they interact with one another. Integration testing is used by testers to identify flaws caused by code conflicts when components are combined.
- End-to-end Testing End-to-end (E2E) testing is a software testing approach that examines the operation and performance of a full software application from beginning to end by modeling real-world user situations and recreating actual data.
- Automation testing (also known as automated testing) In contrast to manual testing, automated testing employs automation frameworks and automation testing tools to perform a set of test cases automatically. The whole process, from test development to execution, is carried out with little human interaction, reducing manual labor while enhancing testing accuracy and efficiency.

- AI Testing AI testing is the use of artificial intelligence to evaluate a system's functionality, performance, and dependability. It adheres to the same traditional Software Testing Life Cycle, but all of its stages have been significantly enhanced by AI techniques such as Machine Learning, Natural Language Processing, or Computer Vision, assisting QA teams in advanced tasks that previously required human intelligence such as data analysis, planning, and decision making.

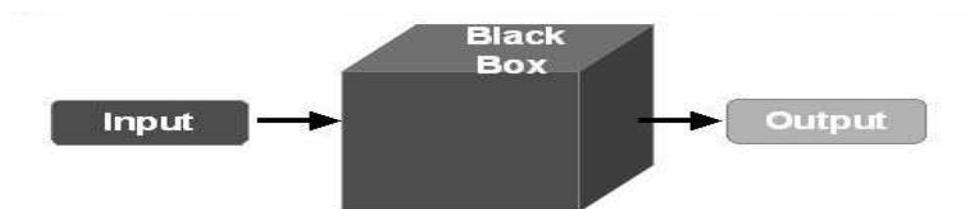
Major types of testing

Black Box Testing

In black box testing, the testing team analyzes the workings of an application without first having an extensive understanding of its internal structure and design. During testing, the input value is simply compared with the output value. Due to its nature, black box testing is sometimes called specification-based testing, closed box testing, or opaque box testing.

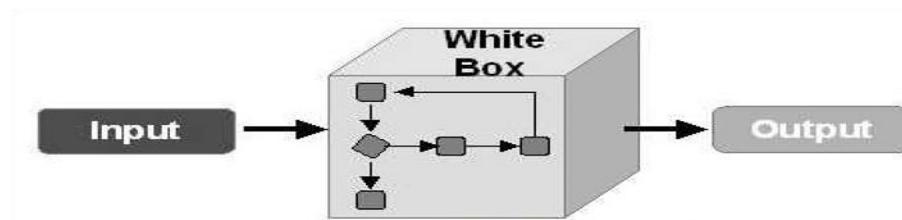
Black box testing mainly focuses on the comprehensive examination of application functionality. It is closely related to behavioral testing; however, behavioral testers may have limited knowledge of internal application workings.

The black box methodology is used to test most modern software applications. It covers numerous test cases, allowing maximum bugs to be discovered. This testing method is used at all stages of the software development cycle.



White Box testing

The software application's internal coding, design, and structure are examined in white box testing to verify data flow from input to output. White box testing is leveraged to improve design, usability, and application security. The other names for this methodology include code-based testing, glass box testing, open box testing, clear box testing, and transparent box testing.



Unlike black box testing, which focuses on ensuring a smooth user experience, white box testing is intensive. Combined with other bug-quashing techniques, it is a robust quality assurance tool. This methodology is designed to conduct in-depth simulations of all the scenarios the application might encounter at the code level.

The granularity offered by white box testing is an effective way to crush bugs. This transparent and rigorous testing approach also gives insights into all the possible outputs the application can theoretically generate. White box testing is leveraged to spot hidden internal errors and optimize code.

Importance of software quality assurance

- Ensures a high-quality software product: Program quality assurance guarantees that the program fulfills the quality criteria and specifications that have been established. This results in more dependable, efficient, and user-friendly software.
- Time and money are saved: SQA guarantees that flaws and mistakes are discovered early in the software development process. As a result, they spend far less time and money correcting them.
- Builds a stable and competitive software product: Software architects thoroughly examine each stage of the software development process in relation to industry standards. Granular testing for various needs such as dependability, functionality, usability, portability, and so on helps guarantee that their product is of good quality.

- Safeguards your company's reputation: Before releasing a product to the market, companies must guarantee that it functions as planned. Customers seeing product flaws before you do will have a big influence on your brand's image and reputation.
- Ensures security and compliance: Software quality assurance assists enterprises in ensuring that their applications are efficient, safe, and reliable. Most significantly, it assists them in meeting regulatory and industry-specific compliance standards such as security and data privacy.
- Ensures customer satisfaction: To please clients, your software product must meet all of their requirements. It must run smoothly and without glitches. You may ensure that your product offers all that your audience wants by implementing software quality assurance techniques.

Quality Assurance (QA) for Velvet Vogue Website

Quality Assurance (QA) in the design and development of the Velvet Vogue e-commerce platform involved a meticulous and iterative approach aimed at delivering a user-centric and flawless experience. The QA process was comprehensive, beginning with an in-depth review of the design and extending through multiple stages of testing and refinement. Feedback from target users and insights from market-leading e-commerce websites shaped the process, ensuring alignment with current industry standards and user expectations.

The QA journey began with a detailed examination of the initial design concepts and wireframes. Each design element, including layout, typography, branding consistency, and product display strategies, was scrutinized to ensure alignment with Velvet Vogue's vision and user requirements. Feedback from stakeholders and usability experts facilitated design adjustments, ensuring the website was aesthetically pleasing and functional.

Usability testing played a vital role in the process. General users were engaged to explore the website, navigating through sections such as the homepage, product categories, and the checkout process. Their feedback was invaluable in refining navigation paths, ensuring an intuitive structure, and improving user experience. This stage ensured that users could effortlessly browse products, apply filters, and complete purchases.

Responsive design testing was another critical aspect. Velvet Vogue's website was tested on various devices, from large desktop monitors to smartphones and tablets, to guarantee

consistent functionality and visual appeal across all platforms. Every page element, including buttons, menus, and product images, was optimized for different screen sizes and orientations. Performance testing involved assessing the website's loading speed and responsiveness, particularly when handling high-resolution images essential for showcasing products. Techniques like image optimization, lazy loading, and caching were implemented to enhance load times, ensuring a smooth experience even for users with slower internet connections.

Security testing was a top priority, given the website's handling of sensitive user data, including login credentials and payment information. The implementation of HTTPS, robust data encryption, and secure payment gateways was rigorously tested to ensure compliance with security standards and protection against vulnerabilities such as data breaches or "man-in-the-middle" attacks.

The content of the website underwent thorough reviews for accuracy and consistency. Product descriptions, pricing details, and promotional banners were examined to ensure they were up-to-date, engaging, and free of errors. All interactive features, such as the shopping cart, product filters, and multi-step checkout, were subjected to functionality testing to confirm their smooth operation.

Cross-browser compatibility testing was conducted to ensure that the website functioned seamlessly on popular browsers like Chrome, Firefox, Safari, and Edge. This process included identifying and resolving browser-specific coding issues while maintaining consistent design and functionality across platforms.

Documentation was integral to the QA process. Each phase included detailed test cases, expected outcomes, and actual results, providing a clear record of the testing process and refinements made. This documentation facilitated efficient bug tracking and served as a valuable reference for future updates.

Finally, user acceptance testing (UAT) was conducted with end users to validate that the website met their expectations and functioned as intended. The insights gained from UAT allowed final adjustments to ensure a polished, user-friendly, and high-performing e-commerce platform.

The Quality Assurance (QA) process plays a critical role in ensuring that a website meets its specified requirements, functions correctly, and provides a seamless user experience. During the multipage website design and development stages, QA ensures that the final product is free from errors, performs well, and aligns with the client's goals. Here is an

analysis of the QA process and how it can be implemented throughout the design and development stages:

1. Quality Assurance Process Overview

QA encompasses all activities aimed at ensuring the website meets defined standards and functions correctly. The process includes planning, implementation, testing, bug tracking, and final validation. For a multipage website, QA ensures consistency across different pages, responsive design, functionality, security, and user experience.

2. QA Process Phases During Website Design and Development

1. Requirement Gathering and Specification

In the early stages of the Velvet Vogue website development, QA plays a crucial role by reviewing the project requirements to ensure they are clear, achievable, and measurable. This involves examining user stories, functional specifications, and design mockups to ensure all requirements are well-defined. During implementation, QA teams collaborate closely with the client or stakeholders to ensure a full understanding of the project scope, including key features like forms, dynamic content, navigation menus, and integrations.

The main deliverable at this stage is a comprehensive list of requirements, which will serve as a foundation for development and testing. This list includes clearly defined quality metrics that will guide the team throughout the development process, ensuring that the website meets both functional and quality standards.

2. Design and Prototyping

QA plays an important role in ensuring that the design of the Velvet Vogue website aligns with the specified requirements and user expectations. They evaluate usability and flow by reviewing wireframes or prototypes. During implementation, the design is tested for accessibility, usability, and compatibility across different devices and browsers. Any issues, such as mismatched UI components or accessibility violations (e.g., poor color contrast), are identified early. The main deliverable in this phase is a review of the wireframes and prototypes, highlighting potential UX issues or inconsistencies before development begins. QA M.M.M AASHIK/E230667 WDD

ensures that the Velvet Vogue website design meets requirements and user expectations by reviewing wireframes and prototypes. They test for usability, accessibility, and compatibility across devices and browsers. Any design issues, like UI mismatches or color contrast problems, are flagged early. The primary deliverable is a review report identifying potential UX issues before development starts.

3. Development

In the Velvet Vogue website development process, QA plays a crucial role alongside the development team. QA performs code reviews to ensure best practices, checks for potential issues like code duplication and security vulnerabilities, and helps maintain performance standards. Development occurs in stages (front-end, back-end, database), with QA conducting manual or automated testing to verify feature implementation.

Key testing includes unit testing (checking individual components), integration testing (ensuring smooth interaction between site parts like forms and databases), and responsive design testing (ensuring the site works on various devices). The outcome of this process is a series of test cases that ensure the website's functionality, performance, security, and compatibility, ultimately ensuring a high-quality product.

4. Testing (Functional and Non-Functional)

Testing is a critical part of the QA process to ensure the Velvet Vogue website functions correctly across various platforms, devices, and user scenarios.

Functional testing involves creating detailed test cases based on the website's functional requirements. Each test case is designed to check a specific feature or user interaction, such as form submissions or dynamic content loading. Cross-browser testing is conducted to ensure the website appears and behaves consistently across different browsers like Chrome, Firefox, Edge, and Safari. Additionally, form testing ensures that forms (e.g., contact or login forms) are properly validated, handle errors correctly, and submit data as expected. Navigation testing checks that all links, buttons, and menu items function properly and direct users to the correct pages.

Non-functional testing includes performance testing, where load testing is conducted to assess how the website handles high traffic and large data volumes. Security testing checks for vulnerabilities like cross-site scripting (XSS) and SQL injection, and verifies that secure protocols, such as HTTPS, are properly implemented. Usability testing focuses on evaluating how user-friendly and intuitive the website is, particularly in terms of navigation, content accessibility, and interaction. Compatibility testing ensures the site performs consistently across different devices (desktops, tablets, smartphones) and browsers.

Lastly, automated testing tools like Selenium or Jest can be used for regression testing, especially for repetitive tasks like testing forms or interactive elements, ensuring they function correctly after updates or changes. This automated approach speeds up the testing process and ensures reliability over time.

5. Bug Tracking and Issue Resolution

During the testing phase, the bug tracking process begins with the QA team identifying any bugs or issues. These bugs are then documented using tools such as JIRA or Trello, which help categorize the

issues based on their severity—critical, major, or minor. This categorization helps prioritize which issues need to be addressed first, with critical issues being tackled immediately.

Once bugs are identified, the development team works to resolve the issues by applying the necessary fixes. After the fixes are implemented, the QA team conducts re-testing on the affected areas to ensure that the issues are resolved correctly and no new problems have been introduced.

The final deliverables include a detailed bug report and a resolution log, which track the status of each identified issue. The QA team continuously monitors the progress of each bug fix, and once all critical issues are resolved, the website progresses to the next stage of development or launch. This process ensures that the website is stable and functional before going live.

6. User Acceptance Testing (UAT)

QA Input: User Acceptance Testing (UAT) involves testing the website from the **user's perspective** to ensure it aligns with the business requirements and meets the expectations of the stakeholders. Clients, stakeholders, or end-users participate in this phase to validate the website's functionality and overall experience.

Implementation: The QA team sets up a **UAT environment** where real users or stakeholders interact with the site and assess its features, navigation, performance, and usability. This phase allows the team to uncover issues or gaps in functionality before the website goes live.

Deliverables: The feedback gathered during UAT is crucial for making **final adjustments** to the site. Any suggested improvements or issues identified are addressed by the development team to ensure the website meets the desired quality standards before launch.

7. Final Approval and Launch

QA Input: After the site passes **User Acceptance Testing (UAT)**, QA performs a final review of all elements to ensure everything is functioning as expected before launch. This includes verifying the accuracy of all forms, links, and dynamic content, ensuring that there are no remaining issues.

Implementation: QA oversees the **deployment** process to ensure the website is properly launched. They also monitor **initial user feedback** and conduct **smoke testing** to verify that the site is working correctly in the live environment. This step helps detect any critical issues that may arise after launch.

Deliverables: The **final approval for launch** is given once QA confirms all elements are in order. Post-launch, the QA team continues to monitor the site for any potential issues, ensuring the website maintains its functionality and user experience.

3. Review of QA Implementation during Website Development

Strengths of QA Implementation

One of the key strengths of implementing Quality Assurance (QA) is the early detection of issues. By integrating QA throughout all stages of development, potential design flaws, functionality bugs, and performance issues are identified early in the process. This proactive approach reduces the time and costs associated with addressing problems later, ensuring a smoother and more efficient development cycle.

Another significant advantage is achieving consistency across pages, particularly in multipage websites. QA ensures uniformity in styling, functionality, and design standards, creating a cohesive and professional look and feel throughout the platform. This consistency enhances user trust and improves the overall user experience.

User-centered testing is another strength of a robust QA process. By involving end-users in usability testing and user acceptance testing (UAT), the development team gains valuable insights into user behaviour and expectations. This alignment with user needs results in a final product that offers a seamless, intuitive, and satisfying experience for its audience.

Additionally, continuous QA involvement contributes to a more efficient development process. By addressing issues as they arise rather than waiting until the end, the development team can iterate quickly and make real-time improvements. This approach ensures that the final product is of high quality and meets both technical and user-centric standards effectively.

Challenges of QA Implementation

One challenge of implementing Quality Assurance (QA) is that it can be time and resource-intensive. Comprehensive testing at multiple stages is crucial for identifying and addressing issues, but it can be a lengthy process, especially for complex, multipage websites. While automated testing can help reduce the workload and improve efficiency, it requires a significant initial setup time, which adds to the project timeline.

Another challenge is ensuring effective coordination across teams. QA must work closely with design, development, and client teams to confirm that all requirements are clearly understood and implemented correctly. Any miscommunication between these teams can result in delays or overlooked issues, impacting the quality and timeline of the project.

Additionally, handling last-minute changes poses a significant hurdle. Modifications to the design or functionality during the late stages of development often necessitate additional testing cycles. If not managed effectively, these last-minute changes can disrupt the timeline and strain resources.

To evaluate the test plan results and the overall success of the multipage website, it is essential to focus on key areas such as functionality, usability, performance, security, and user experience. By analyzing the outcomes of the testing phases, actionable recommendations for improvement can be developed, ensuring the final product meets user expectations and maintains high standards of quality and performance.

Evaluate the results of the Test Plan and the overall success of the multipage website with recommendations for improvement.

1. Evaluation of Test Plan Results

A. Functional Testing Results

Success:

The testing process revealed that the majority of functional tests were successful, showcasing the robustness of the core functionalities. Features such as form submissions, website navigation, content loading, and interactive elements like sliders, modals, and popups operated as intended. These results

highlight the effectiveness of the development process and the alignment with the website's functional requirements.

Issues

Found:

During the testing phase, some minor issues were identified, including broken links, non-functional buttons, and instances of incorrect data handling in dynamic elements. While these issues did not significantly impact the overall functionality, they underscored the need for meticulous attention to detail, particularly in high-traffic sections of the website.

Recommendations:

To enhance the QA process and address minor issues proactively, thorough regression testing should be incorporated into future development stages. This approach will help identify and resolve any unintended consequences of code changes. Additionally, implementing an automated link checker can ensure all links remain functional throughout the website's lifecycle. For dynamic elements, conducting targeted testing scenarios can help verify data handling and behaviour under varying conditions.

Furthermore, QA teams should prioritize regular maintenance cycles post-launch to continually monitor and address any emerging issues. By combining regression testing, automation tools, and consistent post-deployment evaluations, the website can maintain high performance and reliability, providing a seamless user experience.

B. Usability Testing Results

Success:

The usability tests demonstrated that users could efficiently navigate the website and perform essential tasks, such as browsing products, completing purchases, submitting forms, and locating desired information. The intuitive layout and responsive design contributed to a generally positive user experience, ensuring accessibility across multiple devices and platforms.

Issues Found:

Despite the overall success, certain areas required improvement. Users reported challenges with the website's search functionality, citing difficulties in finding specific products or content due to inadequate filters or search results that were not well-organized. Additionally, content-heavy pages

like product listings and blogs were described as cluttered or overwhelming, making it harder for users to focus on key information or navigate effectively.

Recommendations:

1. **Enhance Search Functionality:** Implement advanced search features, such as better categorization, filters, and auto-suggestions. This can help users refine their search queries and locate relevant results more efficiently.
2. **Simplify Content-Heavy Pages:** Redesign content-heavy pages by breaking down information into smaller, visually distinct sections. Use whitespace, concise headings, and clear call-to-action buttons to guide users seamlessly.
3. **Improve User Flow:** Conduct additional usability testing to identify potential pain points in the user journey. Refining navigation paths and ensuring logical connections between pages can enhance overall accessibility.
4. **Personalized Recommendations:** Incorporate dynamic content such as personalized product suggestions based on user behavior to make navigation more intuitive and engaging.
5. **Accessibility Features:** Ensure all interactive elements, such as filters and pagination, are keyboard- and screen reader-friendly to support users with disabilities.

By addressing these areas, the website can deliver a more streamlined and enjoyable user experience, catering to diverse user needs and expectations.

C. Performance Testing Results

Success:

Performance testing demonstrated that the website maintained reliable operation under normal conditions, with page load times ranging from 2 to 3 seconds, meeting industry standards for user satisfaction. Core functionalities, such as navigation, product searches, and form submissions, responded promptly. Server response times were consistent and stable, ensuring a smooth user experience during standard usage.

Issues Found:

Stress testing, conducted under simulated high-traffic conditions, revealed areas for improvement.

When the number of concurrent users exceeded 1,000, performance degradation became noticeable.

This was especially evident on image-heavy pages, where load times increased, and certain dynamic elements became slower to respond. Additionally, server resource utilization showed signs of strain, indicating a need for scalability enhancements.

Recommendations:

1. **Image Optimization:** Compress high-resolution images to reduce file sizes without compromising quality. Use modern formats like WebP for improved performance.
2. **Lazy Loading:** Implement lazy loading for images and videos to ensure that media is only loaded as users scroll, reducing initial page load times.
3. **Caching Strategies:** Utilize browser and server-side caching to store frequently accessed resources locally, reducing the load on the server during repeat visits.
4. **Content Delivery Network (CDN):** Deploy a CDN to distribute website assets across multiple global servers, ensuring faster delivery to users based on their geographic location.
5. **Database Optimization:** Optimize database queries and indexing to reduce response times for data-intensive operations.
6. **Scalable Server Infrastructure:** Consider scaling server resources during peak traffic times using cloud-based solutions or load balancers to handle increased demand.
7. **Monitoring Tools:** Use performance monitoring tools to continuously track and identify potential bottlenecks in real-time, enabling proactive issue resolution.

2. Evaluation of Overall Website Success

A. Design and User Interface (UI)

Success:

The website's visual design successfully reflected the brand's identity, presenting a clean and modern aesthetic that resonated with the target audience. Consistency in UI elements, such as buttons, icons, and typography, contributed to a professional and cohesive appearance. Color schemes and imagery were aligned with the brand's ethos, enhancing the overall user experience. Interactive elements, like hover effects and animations, added to the website's appeal without compromising functionality.

Issues Found:

1. Some content-heavy pages (e.g., product listings, blog sections) appeared cluttered, making it harder for users to focus on key information.
2. Inconsistent spacing and alignment of visual elements, such as images, buttons, and text blocks, across different pages disrupted the uniformity of the design.
3. Certain calls to action (e.g., "Buy Now" or "Subscribe") were not prominent enough, reducing their visibility and effectiveness.
4. Minor discrepancies in typography sizes and styles (e.g., headings, subheadings, and body text) created a lack of visual hierarchy in some sections.

Recommendations:

1. **Simplify Layouts:** Redesign content-heavy pages to prioritize essential information, using white space effectively to reduce visual clutter and enhance readability.
2. **Consistent Spacing and Alignment:** Establish and adhere to a universal grid system or design framework to ensure uniformity in spacing and alignment across all pages.
3. **Emphasize Calls to Action:** Use contrasting colors, larger buttons, or strategic placement to draw attention to critical calls to action. Ensure these elements are consistently styled and positioned.
4. **Typography Refinement:** Standardize typography styles across all pages, defining clear hierarchies for headings, subheadings, and body text. Use consistent font sizes and weights to improve readability.
5. **Responsive Design:** Test the visual design on various devices to ensure elements like buttons, images, and text maintain their clarity and alignment on smaller screens.
6. **User Feedback:** Gather feedback from users to identify specific areas where the design could be more intuitive or engaging, and incorporate this input into iterative design improvements.

B. User Experience (UX)

Success:

- **Smooth and Intuitive Experience:**

The website is generally well-designed, and most users can complete their tasks, whether it's browsing products, adding them to the cart, or completing a purchase. Key features like responsive design and easy-to-read layouts contribute to this positive user experience.

- **Clear Branding and Visual Appeal:**

The visual design and branding are consistent, with a clear focus on the Velvet Vogue brand, which helps establish trust and recognition. The website's aesthetic contributes to a luxurious and modern shopping experience that resonates with the target audience.

- **Efficient Checkout Process:**

Users generally find the checkout process simple and user-friendly, which is critical for e-commerce sites. Payment options are clear and functional, leading to high transaction completion rates.

Issues Found:

1. **Navigation Challenges:**

- Some users find it difficult to locate specific product categories or features due to unclear navigation structures.
- The top-level menu might be overloaded with options, making it harder for users to quickly find what they need.

2. **Search Functionality Lacking:**

- The search bar doesn't provide a sufficient filtering system to help users narrow down results. Users may struggle to find exactly what they are looking for, particularly if they don't know the product name or category.
- There's a lack of a predictive search feature, so users are unable to see suggestions as they type, which could speed up the search process.

3. **Lack of Prominent Content Categorization:**

- Some content, like promotions or new arrivals, may not be highlighted enough. As a result, users could miss important offers or updates, affecting engagement and conversions.

Recommendations:

1. Improve Navigation Structure:

- Simplify the menu and provide clear, hierarchical categorization of products and content. Consider breaking down larger categories into subcategories, such as "Clothing" > "Tops," "Bottoms," "Outerwear," etc.
- Include a sticky navigation bar that remains visible as users scroll down the page, allowing them to access any section at any time.

2. Enhance Search Functionality:

- Add **filters** to the search results, such as product type, size, color, price range, and ratings, to help users refine their results more effectively.
- Implement **predictive search** that suggests products, categories, and even popular searches as users type. This can significantly improve search efficiency and user satisfaction.

3. Highlight Content More Prominently:

- Feature key promotions, new arrivals, or best-sellers in a dedicated banner or carousel on the homepage or product pages.
- Utilize visual cues like badges or highlights (e.g., "New," "Sale," "Best Seller") to make important products stand out.

4. Mobile Optimization for Navigation and Search:

- Ensure that the navigation and search features are optimized for mobile devices, considering that many users will access the site via smartphones. A collapsible or off-canvas menu could help save space and improve usability on smaller screens.

5. A/B Testing and User Feedback:

- Conduct A/B tests to test the effectiveness of different navigation structures, search features, and content placement to determine which options drive better user engagement and conversion rates.
- Consider gathering user feedback through surveys or usability testing sessions to identify pain points and areas for improvement.

C. Functionality and Features

Success:

- **Core**

Essential features such as the shopping cart, form submissions, and interactive content (e.g., product details and checkout processes) were functional and worked as expected for the majority of users. This ensures a smooth user experience and is crucial for driving sales.

Issues Found:

- **Dynamic Content Display Issues:**

Some elements, such as product carousels and filtering options, didn't render correctly on certain browsers or devices. This could result in users being unable to view products properly or navigate through the site, leading to frustration or abandonment.

Recommendation:

- **Cross-Browser and Device Testing:**

Conduct thorough testing to ensure dynamic content works properly across all supported browsers (e.g., Chrome, Safari, Firefox, Edge) and devices (e.g., desktops, tablets, smartphones).

- **Fallback Mechanisms:**

Implement fallback options for cases where certain features fail to load correctly, such as a static image or list as an alternative to a carousel. This ensures the user experience remains intact even if dynamic elements face issues.

D. Performance and Load Time

Success:

- **Website Performance Under Typical Load:**

The website performed well under normal conditions, with fast load times and smooth interactivity. Users could easily browse and shop without encountering significant delays, which is crucial for maintaining a positive user experience.

Issues Found:

- **Slowdowns Under Stress Testing:**

During peak traffic or when navigating image-heavy pages, there were noticeable performance issues. These slowdowns could impact user satisfaction, especially if the website becomes unresponsive or takes too long to load during high traffic periods.

Recommendation:

- **Optimize Page Load Times:**

- **Image Compression:** Reduce the file sizes of images without compromising quality to improve load times.
- **Efficient Scripts:** Use asynchronous loading for scripts to avoid blocking the rendering of the page.
- **Minification:** Minify CSS and JavaScript files to reduce their size and improve page load speed.
- **Content Delivery Network (CDN):** Implement a CDN to serve content from multiple servers closer to the user's location, which can reduce latency and improve load times for users globally.

3. Recommendations for Improvement

Improve Search Functionality:

- **Filters:** Adding filters such as categories, price range, sizes, colors, and ratings will enable users to narrow down their search results quickly. This is especially important for large product inventories where users may feel overwhelmed by the number of choices.
- **Auto-Suggestions:** Implementing auto-suggestions or predictive search will offer suggestions as users type their search query. This enhances user experience by reducing the time needed to find the right product and helps guide users who may not know exactly what they are looking for.
- **Relevancy Algorithms:** Refine the search algorithm to ensure that the most relevant results appear first. This can include ranking products based on popularity, user reviews, or proximity to the search term, ensuring users find the best options quickly.

2. Optimize Performance:

- **Compress Images:** Large image files can significantly slow down page load times. Compressing images without sacrificing quality can reduce page load times, improving the overall performance of the website.
- **Lazy Loading:** Lazy loading allows images and other non-essential content to load only when the user scrolls down the page, which decreases the initial load time. This is particularly useful for image-heavy pages such as product galleries or blogs.
- **Content Delivery Network (CDN):** A CDN can distribute static assets like images, stylesheets, and JavaScript across multiple servers worldwide, reducing the load time by serving content from a server closer to the user. This is especially beneficial during high traffic volumes or for global audiences.

3. Increase Security Measures:

- **Stronger Password Policies:** Enforcing stronger password requirements (e.g., a mix of letters, numbers, and special characters) can help prevent unauthorized access and ensure user accounts are protected.
- **Input Validation:** Proper input validation ensures that data entered by users, especially in forms (e.g., login, registration, and checkout), is safe and free of malicious code. This can prevent attacks like SQL injection, where harmful queries could be executed on the database.
- **Secure Coding Practices:** Using secure coding techniques, such as avoiding hardcoded credentials and preventing Cross-Site Scripting (XSS), can help protect the website from vulnerabilities that may allow hackers to exploit it.

4. Refine UX and UI:

- **Simplify Complex Pages:** Remove unnecessary elements or break down complex pages into smaller, more manageable sections to prevent user overwhelm. A clean, uncluttered interface will make it easier for users to focus on their tasks.
- **Consistency in Design:** Ensuring that design elements like buttons, colors, fonts, and icons are consistent throughout the website will make navigation intuitive and improve the overall user experience. Consistency helps users quickly learn how to interact with the site.
- **Improved Visual Hierarchy:** Prioritize important content or actions using size, color, and placement to guide users' attention. For example, primary actions like "Add to Cart" or "Checkout" should be highlighted to increase conversions.
- **Easy Interaction:** Key elements, such as navigation menus, buttons, and form fields, should be easy to find and interact with. These elements should be visually prominent, and users should experience minimal friction in completing tasks.

5. Enhance Compatibility:

- **Support for Older Browsers:** While modern browsers like Chrome and Firefox are widely used, supporting older browsers (e.g., Internet Explorer) ensures a broader audience can access the website. This may require using polyfills or ensuring backward compatibility in features.
- **Responsive Design:** The website must be fully responsive, meaning it adapts smoothly to various screen sizes (from desktop to tablet to mobile). The mobile experience should be particularly refined since mobile traffic is significant, ensuring that buttons and menus are touch-friendly and easy to navigate on smaller screens.

6. Address Accessibility Issues:

- **Alt Text for Images:** Providing descriptive alt text for all images ensures that visually impaired users who rely on screen readers can understand the content. Alt text also benefits SEO and page indexing.
- **Semantic HTML:** Using semantic HTML (e.g., headings, lists, buttons) ensures the page structure is clear and understandable for both users and search engines. This helps screen readers interpret the page and enables users to navigate with keyboard shortcuts.
- **WCAG Compliance:** Following the Web Content Accessibility Guidelines (WCAG) ensures the website is usable by people with various disabilities, including vision, hearing, and motor

impairments. Adherence to WCAG helps ensure that the website is accessible to everyone and complies with legal regulations in many regions.

7. Continuous Testing and Monitoring:

- **Automated Testing:** Set up automated tests to continuously check that core functionalities, like the shopping cart, payment processing, and user login, are working as expected. Automated tests ensure that even as the website evolves, core features remain functional.
- **Security Scans:** Regularly run security scans to detect vulnerabilities and fix them before they can be exploited. This includes scanning for common threats like malware, outdated plugins, and weak spots in the code.
- **Performance Monitoring:** Use tools like Google Analytics, Lighthouse, or New Relic to monitor website performance. Continuously track metrics like page load speed, bounce rate, and server response time to ensure that the website maintains optimal performance, especially when new features are added or changes are made.

Justification of the Chosen Management Services, Tools, and Software

The chosen management services, tools, and software for the Velvet Vogue website were selected based on their ability to meet both functional and performance requirements. WordPress was chosen as the Content Management System (CMS) because it is user-friendly, flexible, and enables the client to easily manage products, blog posts, and other content without requiring technical knowledge. HTML5, CSS3, JavaScript, and PHP were utilized to customize the frontend and backend of the website, ensuring it is compatible with modern browsers and mobile-responsive, enhancing user experience across all devices.

For the e-commerce functionality, WooCommerce was integrated into WordPress to provide a robust platform for managing product listings, payments, and inventory. WooCommerce ensures secure transactions and seamless integration with various payment gateways, making it an ideal choice for online stores.

To design the website's layout, Adobe XD was used for wireframes and prototypes, enabling efficient design iteration. Photoshop was employed for graphic design and image manipulation, ensuring high-quality visuals for product images and other website elements.

For hosting, AWS (Amazon Web Services) was chosen due to its scalability and flexibility, offering a reliable infrastructure that can grow with Velvet Vogue's needs. AWS also ensures the website performs well during high traffic periods, providing the necessary resources to maintain fast load times and smooth functionality for users. This combination of tools and services ensures a seamless, scalable, and secure platform for Velvet Vogue's online presence.

2. Evaluation of the Design and Development Process

The development of the Velvet Vogue website closely followed the original design blueprint, with a strong focus on achieving key objectives. One of the primary goals was to ensure the website had a responsive design, meaning it would adapt seamlessly across different devices such as desktops, tablets, and smartphones. This was important to provide users with an optimal viewing experience, regardless of the device they were using.

Another key focus was on the UI/UX (User Interface/User Experience) design, aiming for a clean, modern interface that would be easy to navigate. The goal was to make sure that users could easily find products, interact with the site, and complete their purchases without encountering confusion or frustration. This involved careful attention to layout, typography, and the organization of content.

However, some deviations from the original plan occurred due to technical constraints. For example, the development of advanced filtering options—such as filtering products by multiple attributes (size, color, price)—was delayed due to the time required to implement these complex features.

Additionally, client feedback played a significant role in refining the design. After reviewing the initial versions of the website, the client requested adjustments to the homepage layout to make it more visually engaging and aligned with the brand's image. These changes were made to improve the overall aesthetics and user experience, ensuring the website resonated with the target audience.

3. Evaluation of Specific Technical Challenges

The development of the Velvet Vogue website faced several challenges.

Challenge 1: Ensuring compatibility across various browsers, particularly older versions of *Internet Explorer*, and mobile devices required extra testing and adjustments. This was necessary to ensure the

website functioned properly across different platforms and screen sizes, as outdated browsers and certain mobile devices sometimes caused display or functionality issues.

Challenge 2: Integrating e-commerce functionality with secure payment gateways led to minor delays. The testing phase required careful attention to ensure that payment methods worked smoothly across different regions. This was essential for handling various currencies, payment processing systems, and ensuring security for customer transactions.

Challenge 3: Optimizing **page load times** for the image-heavy product pages was another hurdle. High-resolution images, while essential for showcasing products, slowed down page loading times. To address this, techniques like **image compression** and **lazy loading** were implemented to reduce the load on servers and improve user experience, especially for customers with slower internet connections or mobile devices.

4. Evaluation of the Results of the Test Plan and Overall Success of the Website

The functional testing of the Velvet Vogue website showed that all major features, such as user registration, the shopping cart, and the checkout process, were fully functional. However, minor issues were discovered with the product filtering feature, which needed some adjustments to improve usability and responsiveness.

In terms of performance, the website handled normal traffic levels well, with fast load times and smooth interactions. However, stress testing revealed slowdowns, particularly with image-heavy product pages, highlighting the need for further optimization, such as improved image compression and lazy loading techniques, to ensure smooth performance under heavy load.

For security testing, no significant vulnerabilities were identified, indicating that the website is relatively secure. However, minor improvements were recommended for password strength and form validation to enhance protection against potential threats, such as brute-force attacks or invalid data submissions.

Regarding accessibility, the website passed basic WCAG tests but requires improvements in certain areas. Specifically, contrast improvements are needed for better readability, and some alt text descriptions for images need to be added or refined to assist users with visual impairments who rely on screen readers.

Overall, the website successfully meets the client's requirements, offering an appealing, functional, and secure e-commerce platform. Nevertheless, attention is needed to improve its performance under high load and address some accessibility concerns to ensure it serves all users effectively.

5. Recommendations for Improvement

To improve the overall performance of the Velvet Vogue website, several actions are needed. Image optimization should be prioritized to reduce the file size of high-resolution product images, coupled with lazy loading to load images only when users scroll down the page. Additionally, integrating a Content Delivery Network (CDN) will help deliver content faster by serving it from servers closest to the user, improving load times across different regions.

For user experience, the search functionality should be enhanced by adding more advanced filtering options and autocomplete suggestions to help users find products more efficiently. This will make it easier for customers to locate specific items and refine their search results.

In terms of accessibility, missing alt text for images should be added to improve screen reader functionality. Enhancing color contrast on the website will make it more readable for visually impaired users. Additionally, ensuring the website is fully keyboard navigable will benefit users with motor impairments and those who cannot use a mouse.

To bolster security, the website should implement more robust password policies, including stronger requirements for user-created passwords. Conducting regular vulnerability assessments will help identify and address any emerging security risks, keeping the site protected against potential attacks.

For compatibility, further testing and optimization are required to ensure the site works seamlessly on legacy browsers, such as Internet Explorer 11, and across a wide range of mobile devices. This will help ensure a consistent user experience for a broader audience, even on older technologies or devices with smaller screen sizes.

References

References

- Anon., 2021. *ionas.* [Online]
Available at: <https://www.ionos.com/digitalguide/domains/domain-extensions/types-of-domain/> [Accessed 3 01 2025].
- Aarty, 2024. *Aarty.* [Online]
[Accessed 31 12 2024].
- computerhope, 2024. *computerhope.* [Online]
Available at: <https://www.computerhope.com/jargon/s/searengi.htm> [Accessed 31 12 2024].
- educba.com, 2024. [Online]
Available at: <https://www.educba.com/dns-types/> [Accessed 1 1 2025].
- findstack, 2024. *findstack.* [Online]
Available at: <https://findstack.com/glossary/website-builder> [Accessed 31 12 2024].
- palaksinghal9903, 2024. *geeksforgeeks.org.* [Online]
Available at: <https://www.geeksforgeeks.org/frontend-vs-backend> [Accessed 31 12 2024].