UNIVERSITY of GREENWICH

Alliance with FPT Education

ĐẠI HỌC GREENWICH (VIỆT NAM)

Liên kết với Tổ chức Giáo dục FPT

ASSIGNMENT

WEBSITE DESIGN AND DEVELOPMENT

STUDENT PERFORMANCE: TRAN QUANG HUY

ID: GCD18457

CLASS: GCD0824

TECHER: SRIKANTHRAJU

ASSIGNMENT 1 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	10: Website Design & Development		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Tran Quang Huy	**Student ID**	GCD18457
Class	GCD0824	**Assessor name**	

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student's signature	HUY

Grading grid

P1	P2	P3	P4	M1	M2	M3	D1

❒ **Summative Feedback:**	❒ **Resubmission Feedback:**

Grade:	**Assessor Signature:**	**Date:**

Signature & Date:

Contents

INTRODUCTION 1

I. SERVER TECHNOLOGIES AND WEBSITE MANAGEMENT SERVICES ASSOCIATED WITH HOSTING AND MANAGING WEBSITES 2

1 Domain name system (DNS): 2

2 The purpose and relationships between communication protocols, server hardware, operating systems and websites server software. 7

3 The impact of common websites development technologies and frameworks. 11

4 The impact of search engines on websites performance. 14

II. WEBSITE TECHNOLOGIES, TOOLS AND SOFTWARE USED IN WEBSITE DEVELOPMENT 19

1 The capabilities and relationships between front-end and back-end. 19

2 The differences between online websites creation tools and custom built sites. 22

3 Evaluate a range of tools and techniques available to design and develop a custom build website. 28

III. CUSTOM BUILT WEBSITE DESIGN WITH TOOLS AND TECHNIQUES 39

1. Design tools: 39

2. Front-end: 40

3. Back-end: 41

4. Server: 42

CONCULISION 43

References 44

TABLES OF PICTURES

Picture 1: How to find IP address from a website 2
Picture 2: Connect website by IP address 2
Picture 3: Example some DNS in the world 3
Picture 4: Purpose and types of DNS 4
Picture 5: Example Website's URL 10
Picture 6: Front-end and Back-End 19
Picture 7: Template websites and Custom website 22
Picture 8: WordPress interface 23
Picture 9: Wix interface 24
Picture 10: Custom built websites 25
Picture 11: SEO in custom built websites 25
Picture 12: Pros and cons of template and custom built website 26
Picture 13: Pencil interface 28
Picture 14: Draw.io interface 30
Picture 15: Sublime text interface 34
Picture 16: Chrome Developer Tools interface 35
Picture 17: Pencil tools interface 39
Picture 18: Visual studio code interface 40

TABLE OF FIGURES

Figure 1: The TCP/IP model or Internet layering scheme and its relation to some common protocols. 7
Figure 2: TCP/IP Model and OSI Model 8
Figure 3: How server and client works 10
Figure 4: Traffic referrer data in SEO 15
Figure 5: Proportions of web traffic 16
Figure 6: Front-end development 20
Figure 7: Back-end Development 21

TABLE OF TABLES

Table 1: The most common DNS record types 6
Table 2: PHP Framework 11
Table 3: Ruby Framework 12
Table 4: Python Framework 12
Table 5: JavaScript Framework 13
Table 6: Design/CSS Frameworks 13
Table 7: Compare Online website and custom built 27

INTRODUCTION

The web design process starts with a visual concept, which you could sketch by hand or with software like Photoshop. Then, you use HTML and CSS to build the website. HTML and CSS are the codes for writing web pages. HTML handles the basic structure and 'bones' of your page, while CSS handles the style and appearance.

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardized code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all. The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing markup. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating markup then they are also expected to be up to date with web accessibility guidelines.

This report presents website concepts such as static web, dynamic web, protocols, web programming language as well as tool’s support. This can tell how a website can run and what elements it takes to create a website.

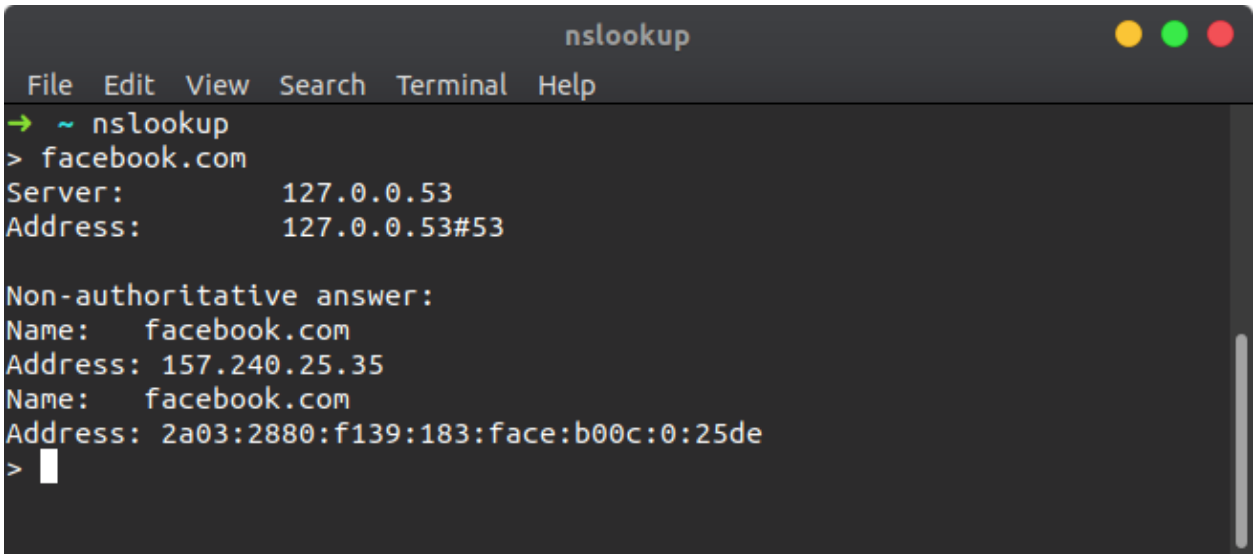
Some examples are screened on Linux operating system (Ubuntu) so it will be a bit different from windows operating system.

I. SERVER TECHNOLOGIES AND WEBSITE MANAGEMENT SERVICES ASSOCIATED WITH HOSTING AND MANAGING WEBSITES

1 Domain name system (DNS):

1.1 Domain names:

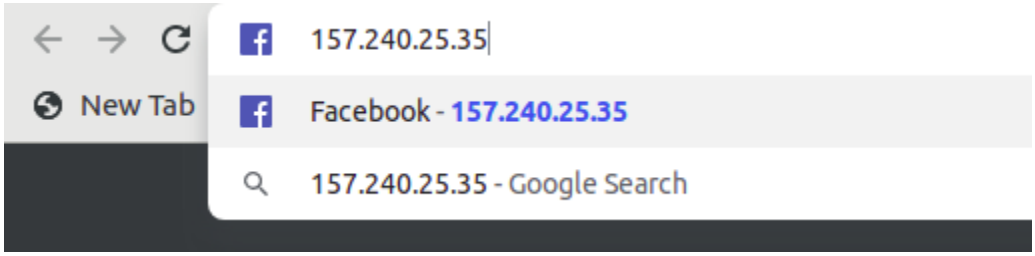
Domain names were created to convert the numeric **IP addresses** into a simple, recognizable name. In the Internet, these **domain names**, such as facebook.com, are much easier for people to remember than 157.240.25.35, which, at the time of this writing, is the numeric **IP address** for this server.



Picture 1: How to find IP address from a website

The Domain Name System (DNS) is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network. The domain name system maps the name people use to locate a website to the IP address that a computer uses to locate a website. (wikipedia, n.d.)

For example: If someone types **facebook.com** into a web browser, a server behind the scenes will map that name to the corresponding IP address, something similar in structure to **157.240.25.35**.



Picture 2: Connect website by IP address

The Internet DNS name space is partitioned organizationally and according to geography. The names are written with the highest-level domain on the right.

Picture 3: Example some DNS in the world

The top-level domains use across the Internet were:

.com - Commercial organizations
.edu - Universities and other educational institutions
.gov - Governmental agencies
.net - Major network support centers
.org - Non-profit organizations
.int - International organizations

In addition, each country has its own domains:

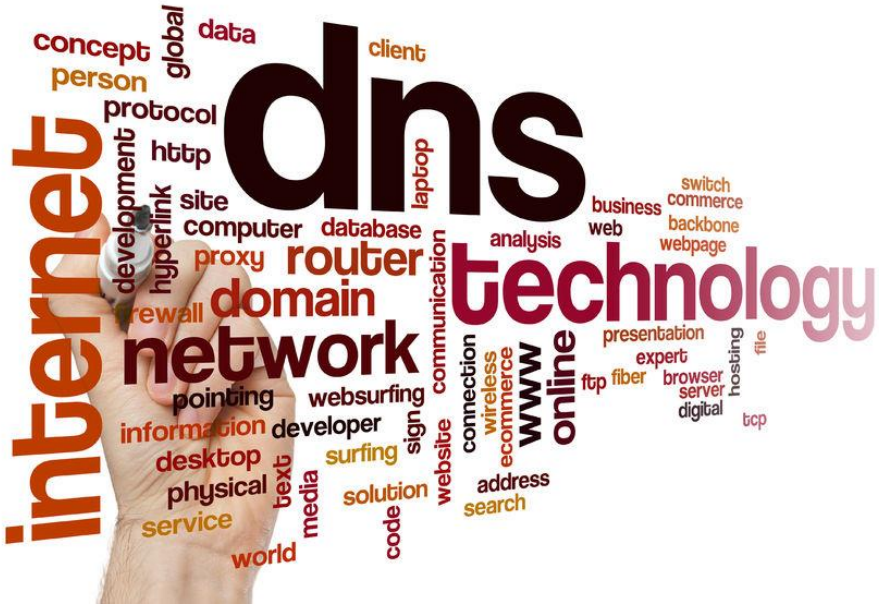
.vn - Viet Nam
.jp – Japan
.kr - Korea
.us - United States
.uk - United Kingdom (ducanhplus, n.d.)

The administration of domains may be devolved to **sub-domains** to assign a unique name to a particular department, function, or service related to the organization, etc.

For example: FPT education domain is **fpt.edu.vn** and there are some sub-domains like: **ap.fpt.edu.vn**, **tuyendung.fpt.edu.vn**, **etc**.

1.2 Purpose and types of DNS:

DNS is an essential part of the Internet. It manages to translate all the inquiries into IP addresses, and like this, it can identify different devices that are connected to the network. (cloudns, n.d.)



Picture 4: Purpose and types of DNS

There are three **types of queries** in the DNS system:

- **Recursive Query:** In recursive query, DNS client provides a hostname, and the DNS Resolver "must" provide an answer—it responds with either relevant resource record, or an error message if it can't be found. The resolver starts recursive query process, starting from the DNS Root Server, until it finds the Authoritative Name Server (for more on Authoritative Name Servers see DNS Server Types below) that holds the IP address and other information for the requested hostname.

- **Iterative Query:** DNS client provides a hostname, and the DNS Resolver returns the best answer it can. If the DNS resolver has the relevant DNS records in its cache, it returns them. If not, it refers the DNS client to the Root Server, another Authoritative Name Server which is nearest to the required DNS zone. The DNS client must then repeat the query directly against the DNS server it was referred.

- **Non-Recursive Query:** A non-recursive query is query in which the DNS Resolver already knows the answer. It either immediately returns a DNS record because it already stores it in local cache, or queries a DNS Name Server which is authoritative for the record, meaning it definitely holds the correct IP for that hostname. There is no need for additional rounds of queries (like in recursive or iterative queries). Rather, response is immediately returned to the client.

The following are the most common **DNS server types** that are used to resolve hostnames into IP addresses:

- **DNS Resolver:** DNS resolver (recursive resolver), is designed to receive DNS queries, which include human-readable hostname such as "www.google.com", and is responsible for tracking the IP address for that hostname.

- **DNS Root Server:** The root server is the first step in the journey from hostname to IP address. The DNS Root Server extracts the Top Level Domain (TLD) from the user's query.

  For example: www.google.com — and provides details for the .com TLD Name Server. In turn, that server will provide details for domains with the .com DNS zone, including "google.com".

  There are 13 root servers worldwide, indicated by the letters A through M, operated by organizations like the Internet Systems Consortium, Verisign, ICANN, the University of Maryland, and the U.S. Army Research Lab.

- **Authoritative DNS Server:** Higher level servers in the DNS hierarchy define which DNS server is the "authoritative" name server for a specific hostname, meaning that it holds the up-to-date information for that hostname.

  The Authoritative Name Server is the last stop in the name server query—it takes the hostname and returns the correct IP address to the DNS Resolver (or if it cannot find the domain, returns the message NXDOMAIN). (ns1, n.d.)

Each DNS Server contain a database of domain names and the corresponding IP addresses, as well as its cache, which holds a record of all other previous requests.

DNS servers create a DNS record to provide important information about a domain or hostname, particularly its current IP address. The most common DNS record types are:

Record Type	Purpose
A	Known as a DNS host record, stores a host name and it corresponding IPv4 address.
AAA	stores a host-name and its corresponding IPv6 address.
CNAME Record	can be used to alias a hostname to another hostname. When a DNS client requests a record that contains a CNAME, which points to another hostname, the DNS resolution process is repeated with the new hostname.
MX Record	specifies an SMTP email server for the domain, used to route outgoing emails to an email server.
NS Record	specifies that a DNS Zone, such as "example.com" Website development technologies consists of Front-end technology and Back-end technology. The establishment of frameworks that are built to serve those technologies are hitting impact on every website's design, functionality and management: is Website development technologies consists of Front-end technology and Back-end technology. The establishment of frameworks that are built to serve those technologies are hitting impact on every website's design, functionality and management: development technologies consists of Front-end technology and Back-end technology. The establishment of frameworks that are built to serve those technologies are hitting impact on every website's design, functionality and management: delegated to a specific Authoritative Name Server, and provides the address of the name server.
PTR Record	allows a DNS resolver to provide an IP address and receive a hostname (reverse DNS lookup).
CERT Record	stores encryption certificates—PKIX, SPKI, PGP, and so on.
SRV Record	a service location record, like MX but for other communication protocols.
TXT Record	typically carries machine-readable data such as opportunistic encryption, sender policy framework, DKIM, DMARC, etc.
SOA Record	this record appears at the beginning of a DNS zone file, and indicates the Authoritative Name Server for the current DNS zone, contact details for the domain administrator, domain serial number, and information on how frequently DNS information for this zone should be refreshed.

Table 1: The most common DNS record types

2 The purpose and relationships between communication protocols, server hardware, operating systems and websites server software.

2.1 Communication protocols

In telecommunication, a communication protocol is a system of rules that allow two or more entities of communications system to transmit information via any kind of variation of a physical quantity. The protocol defines the rules, syntax, semantics and synchronization of communication and possible error recovery methods. Protocols may be implemented by hardware, software, or a combination of both. (wikipedia, n.d.)

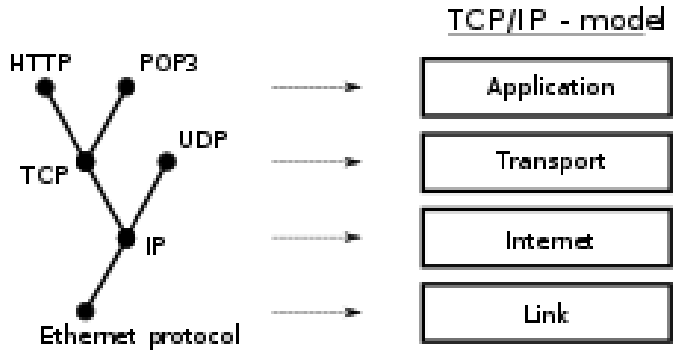


Figure 1: The TCP/IP model or Internet layering scheme and its relation to some common protocols.

- **TCP/IP:** Transmission Control Protocol and Internet Protocol are communication protocols that define how data should travel across the web. TCP helps to connect client devices to the server based on their IP addresses.
- **DNS:** Domain Name System will resolve website's domain names into the server's numeric IP addresses.
- **HTTP:** Hyper Text Transfer Protocol is an application protocol that defines language for client devices and servers to speak to each other.

To implement a networking protocol, the protocol software modules are interfaced with a framework implemented on the machine's operating system. This framework implements the networking functionality of the operating system. When protocol algorithms are expressed in a portable programming language the protocol software may be made operating system independent. The best known frameworks are the TCP/IP model and the OSI model.

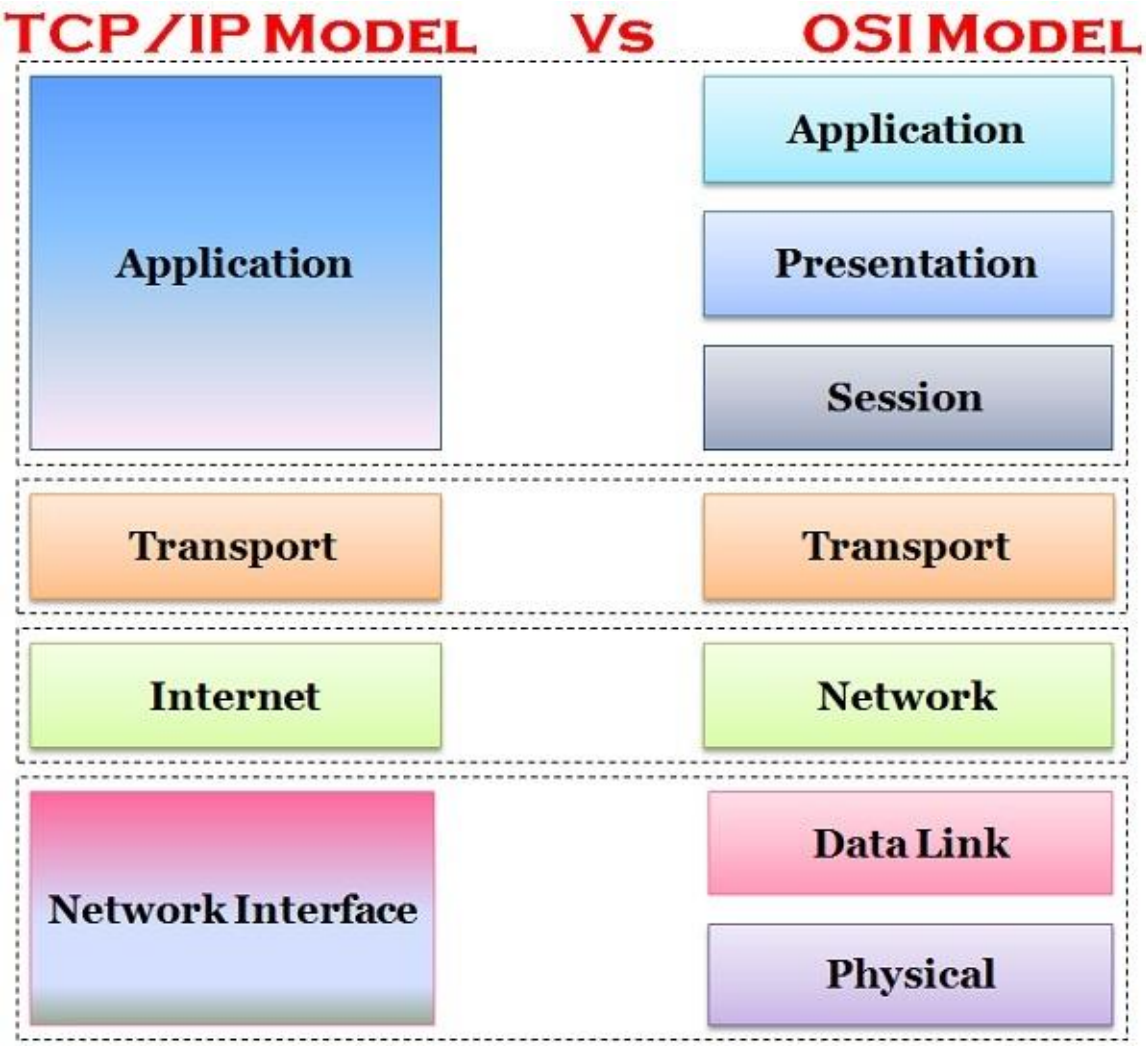


Figure 2: TCP/IP Model and OSI Model

2.2 Website server

Web servers, as well as any other server types, are high-end, powerful computers that are capable to store large amount of data. Nowadays, Web servers are implementing multithreaded server technology, which can serve various requests at the same time.

Web server stores web server's software and website's component files (such as HTML documents, images, CSS stylesheets, JavaScript files, etc.).

Web servers are connected to the Internet and supports physical data interchange with other devices connected to the web.

2.3 Operating system (OS)

An operating system (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs.

Application programs, such as Web browsers, usually require an operating system to function.
There are two types of operating systems commonly used for web servers: Windows and Linux/Unix. Each of them has some major differences, however, have the same purpose to provide a secure and efficient server, where the data resources got secured and managed proficiently.

The web server OS is where the web server software got implemented on.

2.4 Web server software

A web server includes several parts that control how web users access hosted files, at minimum an HTTP server. An HTTP server is a piece of software that understands URLs and HTTP. It can be accesses through the domain names of websites where its stored, and delivers their content to the end-user's device.

The term "Web server" often refers only to the HTTP server software in the machine, which provides the website functionality. HTTP is the protocol of the Web, and HTTP server software, such as Microsoft's IIS and the open source for Apache server, accepts requests from the user's browser and responds by sending back HTML documents (Web pages) and files. It also executes scripts that reside in the server (CGI scripts, JSPs, ASPs, etc.), which perform functions such as database searching and credit card authorization. See IIS and Apache.

2.5 How the website works?

A website is typically a collection of web pages, images and other elements that are linked together to form larger, structured document. Think of a website as a book and each individual page is a web page.
To connect to the Web server, client computers are required to have an internet connection, as well as Web browser application on its OS.

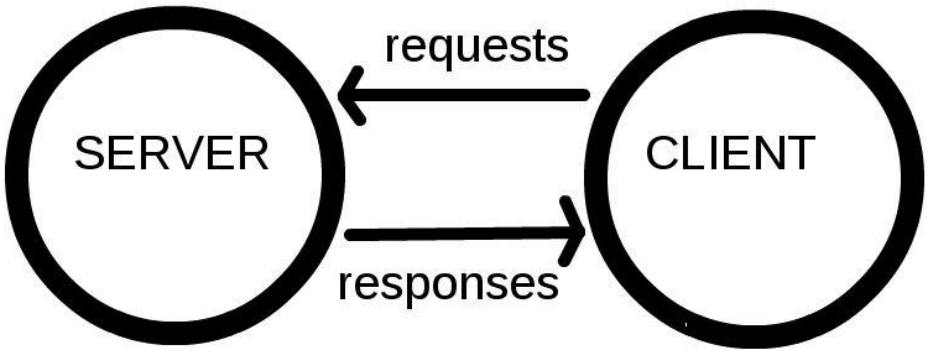
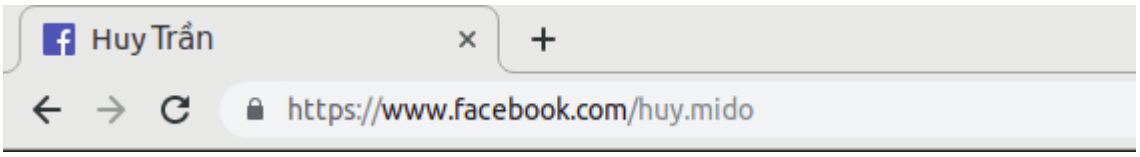


Figure 3: How server and client works



Picture 5: Example Website's URL

Each website's pages are assigned a URL (Uniform Resource Locator) which mainly consist of 3 parts: the communication protocol (or scheme), the domain name of the machine on which the page is located, and the path uniquely indicating the specific page:

Example: https:// is a protocol, **www.facebook.com** is a domain name, /**huy.mido** is a path.

When a user processes a URL into the browser, the browser determines the URL, then asks DNS for the numeric IP address of the website's server.
After located the server's address, the browser makes a TCP connection to that server and send a communication request, such as HTTP request, asking for the page.
On the server side, after accepted the TCP connection from a client's browser and received the client's request, the web server implements its software to indicate the file or resources (from disk) and send the contents of the file back to the client.
Website pages are often written in Hyper Text Markup Language (HTML).
After received the file, the browser will display it.

3 The impact of common websites development technologies and frameworks.

The goal of a framework is to allow designers and developers to focus on building the unique features for their project, rather than re-inventing the wheel by coding common, familiar features found across many websites and web applications

A framework can be considered a pre-built template/structure that handles most of the repetitive or common features. As a result, unlike a CMS, framework will probably not have a user interface (although this is not always the case, as Django provides an administration interface). Most of the activity will be done by writing code and interacting with different parts of the framework itself through code.

Often frameworks take a while to learn, but once you're familiar with them, they should speed up your development time. (ostraining, n.d.)

Website development technologies consists of Front-end and Back-end. The establishment of frameworks that are built to serve those technologies are hitting impact on every website's design, functionality and management. Examples of popular frameworks:

- **PHP:**

Framework	Description
Yii PHP Framework (https://www.yiiframework.com/)	Yii is a fast, secure, and efficient PHP framework. Flexible yet pragmatic Works right out of the box Has reasonable defaults. 
CakePHP (https://cakephp.org/)	CakePHP makes building web applications simpler, faster, while requiring less code. A modern PHP 7 framework offering a flexible database access layer and a powerful scaffolding system that makes building both small and complex systems simpler, easier and, of course, tastier. Build fast, grow solid with CakePHP.

Table 2: PHP Framework

- **Ruby:**

Framework	Description
Rails (https://rubyonrails.org/)	Ruby on Rails makes it much easier and more fun. It includes everything you need to build fantastic applications, and you can learn it with the support of our large, friendly community.
Sinatra (http://sinatrarb.com/)	Sinatra is a DSL for quickly creating web applications in Ruby with minimal effort:
Padrino (http://padrinorb.com/)	Padrino is a Ruby web framework built upon the Sinatra web library. Padrino was created to make it fun and easy to code more advanced web applications while still adhering to the spirit that makes Sinatra great!

Table 3: Ruby Framework

- **Python:**

Framework	Description
Django (https://www.djangoproject.com/)	Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.
Web2py (http://www.web2py.com/)	Free open source full-stack framework for rapid development of fast, scalable, secure and portable database-driven web-based applications. Written and programmable in Python
Flask (https://palletsprojects.com/p/flask/)	Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

Table 4: Python Framework

- **JavaScript:**

Framework	Description
AngularJS (https://angularjs.org/)	AngularJS is toolset for building the framework most suited to your application development. It is fully extensible and works well with other libraries. Every feature can be modified or replaced to suit your unique development workflow and feature needs. Read on to find out how.
EmberJS (https://emberjs.com/)	Ember.js is built for productivity. Designed with developer ergonomics in mind, its friendly APIs help developers efficiently get the job done

Table 5: JavaScript Framework

- **Design/CSS frameworks:**

Framework	Description
Bootstrap (https://getbootstrap.com/)	Build responsive, mobile-first projects on the web with the world's most popular front-end component library. Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.
Foundation (https://foundation.zurb.com/)	A Framework for any device, medium, and accessibility. Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device. Foundation is semantic, readable, flexible, and completely customizable. We're constantly adding new resources and code snippets, including these handy HTML templates to help get you started!

Table 6: Design/CSS Frameworks

4 The impact of search engines on websites performance.

Search engine is a service that allows Internet users to search for content via the World Wide Web (WWW). User enters keywords or key phrases into a search engine and receives a list of Web content results in the form of websites, images, videos or other online data. (techopedia, n.d.)

By the way, **search engines** are answer machines. When person goes an online search, the search engine scours its corpus of billions of documents and does two things:

- First, it returns only those results that are relevant or useful to the searcher's query
- Second, it ranks those results according to the popularity of the websites serving the information.
  SEO stands for **Search Engine Optimization**, which is the practice of increasing the quantity and quality of traffic to your website through organic search engine results. This is also a marketing discipline focused on the practices that affect the visibility, position, and ranking of a websites/web page on a search engine's results page. The higher website is ranked on the search results page(SRP), the better the chances that consumers will interact with it.

There are two broad segments that affect the position of a webpage in SRPs:

On-page SEO:

- Relevance (content, keywords)
- Architecture (page design, UX, performance)

Off-page SEO:

- Social media presence (brand trust, engagement)
- Linking (quality, credibility)

Why Is SEO Important?

In short: search is a BIG source of traffic.

In fact, here's a breakdown of where most website traffic originates:

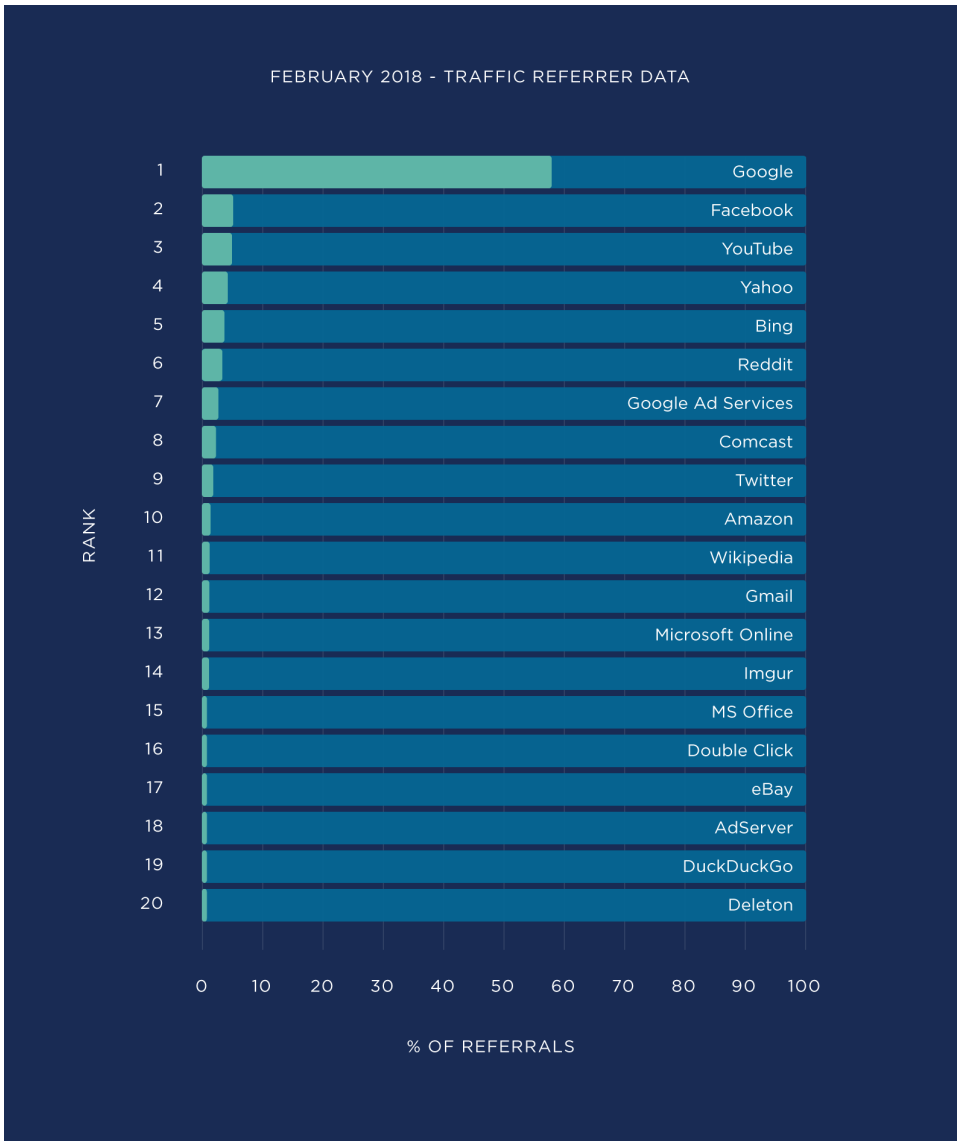


Figure 4: Traffic referrer data in SEO

Nearly 60% of all traffic on the web starts with a Google search. And if you add together traffic from other popular search engines (like Bing, Yahoo, and YouTube), 70.6% of all traffic originates from a search engine.

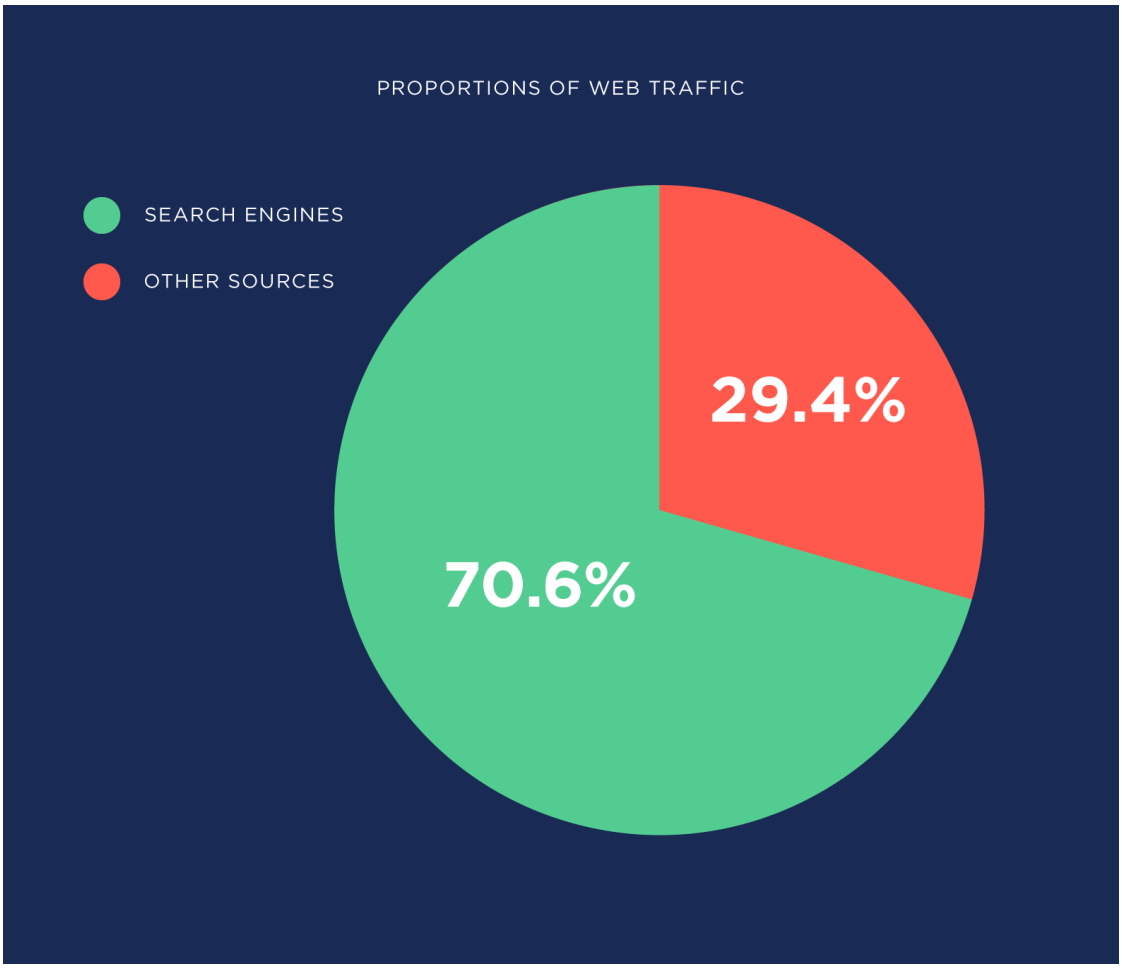


Figure 5: Proportions of web traffic

Because of this, search engines made a great impact on website's development, requiring developers to work harder to improve their site's performance.

According to jeffbullas.com (2018), there are 13 elements that help to improve a site's index value and rank through SEO can support the website:

- **Provide useful, high quality, relevant content:**
  When the site provides useful content, visitors tend to stay longer on it to consume the information and therefore increase the "dwell time"- the amount of time visitors spend on the site that can affect SEO ranking.

- **Improve page load speed:**
  As mentioned earlier in this section about how important of the site's performance is. There are many ways to increase page load speed, some of which include using a caching plug- ins. Making sure the code is clean and streamlined, optimizing image sizes and minimizing redirects.

- **Image optimization:**
  Besides image file format and sizing, developers can signal relevancy of the site's content to search engines by using keywords for the image file name, alt tag, title, description and caption.

- **Using header tags:**
  Good formatting of the content helps to improve the user experience of the website tremendously. It makes readers more willing to spend time to read the content and come back more.

- **Outbound links:**
  To make the content more useful and relevant, linking out to authority sites will not only increase the relevancy of the site's content and time readers spend on them, but it is also believed to send trust signals to Google and improve SEO in ranking.

- **Different multimedia:**
  Images, videos, slideshows and audio can help enrich the UI and allow to deliver information in a way that is most suited to the site's ideal visitor.

- **Broken links:**
  Search engines consider a large number of broken links as a signal of an old, neglected site and this can impact the site's SEO ranking.

- **Readability:**
  Making the site's content easy to read and understand helps make it useful to the readers. "Some experts also believe that Google takes readability into account when ranking webpages".

- **Layout and formatting:**
  Proper formatting and a user-friendly layout can help improve user experience and make the content easy to scan and digest, so readers will stay on the site longer and consume the site's information.

- **Contact Us page:**
  Including a Contact Us page on the website and putting the link in the navigation not only makes for good UI (especially on e-commerce sites) but can also potentially improve the site's SEO ranking.

- **Site architecture and navigation:**
  A well-thought-out site architecture reflected in clear navigation is critical in helping visitors find what they want on the site, accomplish their goals and come back repeatedly.

- **Mobile optimization:**
  As of April 2015, Google has started to penalize sites that are not mobile optimized by bumping down their SEO ranking. Since the number of users accessing the Internet via mobile devices has been surpassing the number using desktop over the years. It is time for developers to make sure their sites are compliant for those devices.

- **Social sharing:**
  Installing social sharing buttons on the website not only makes it easy for users to share the content, and thereby improve the UI. It can also help the site's rank higher on searches.

II. WEBSITE TECHNOLOGIES, TOOLS AND SOFTWARE USED IN WEBSITE DEVELOPMENT

1 The capabilities and relationships between front-end and back-end.

Two terms thrown around a lot in the web industry are front-end and back-end. It can be a little frustrating since the difference between the front-end and back-end isn't always perfectly clear. They are terms often used to describe aspects of the web industry.

Picture 6: Front-end and Back-End

1.1 Front-end websites technologies

The **front-end** is everything involved with what the user sees, including design and some languages like **HTML** and **CSS**

Front-end website technology is a term that involves the building of website pages and User Interfaces (UIs) for web-applications.

Front-end will determine how the website's presentation layer will look like by implements the structure, design, behavior and animation of everything the user sees on the screen when they open up their websites, web-applications or mobile apps.

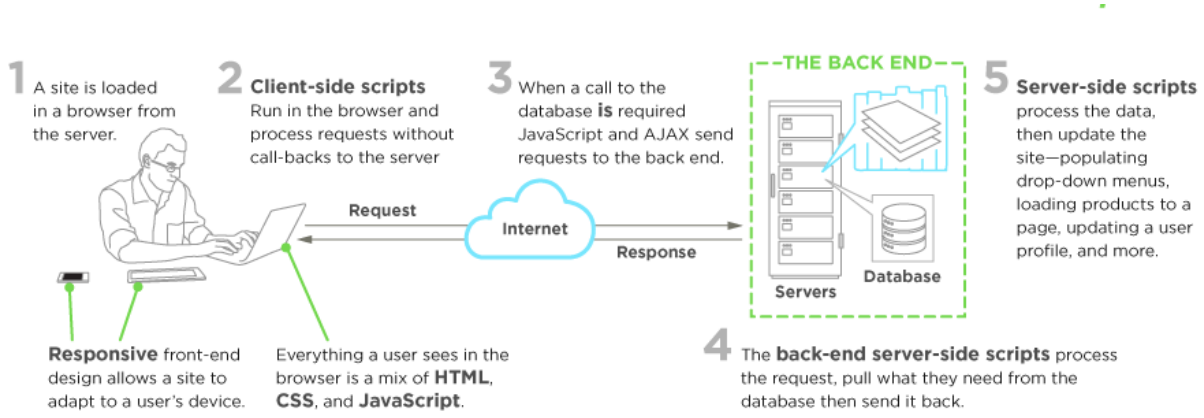


Figure 6: Front-end development

Front-end technology requires client-side scripts, which run in the user's browser and process requests without call-backs to the server, that would put less stress on the server. (frontendmasters, n.d.)
Nearly every website's front-end is a combination of
JavaScript, HTML and CSS, where:

- HTML dictates a site's organization and content
- CSS comprises the code for every graphic element- from backgrounds to fonts- that make up the look and feel of a website
- JavaScript is client-side scripting. By fueled by an array of excellent frameworks that simplify and accelerate the website's agility.

1.2 Back-end websites technologies

The **Back-end** is a mix of the server, databases, APIs, and operating systems that power an app's front end. For an in-depth look at the software that ties it all together, read Server-Side Scripting: Back-End Web Development Technology, or get basic view of back-end technology with our article The Role of the Back-End Developer. In another way, the **back-end**, or the "server-side", is basically how the site works, updates and changes. This refers to everything the user can't see in the browser, like databases and servers. (upwork, n.d.)

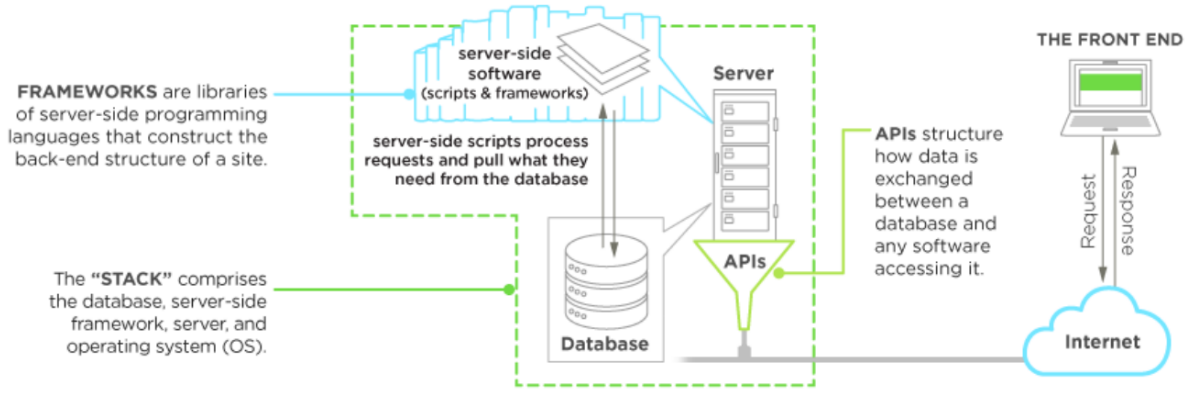


Figure 7: Back-end Development

The data are returned and converted into front-end code a user interacts with, such as filling out a form, creating a profile, shopping online, etc.

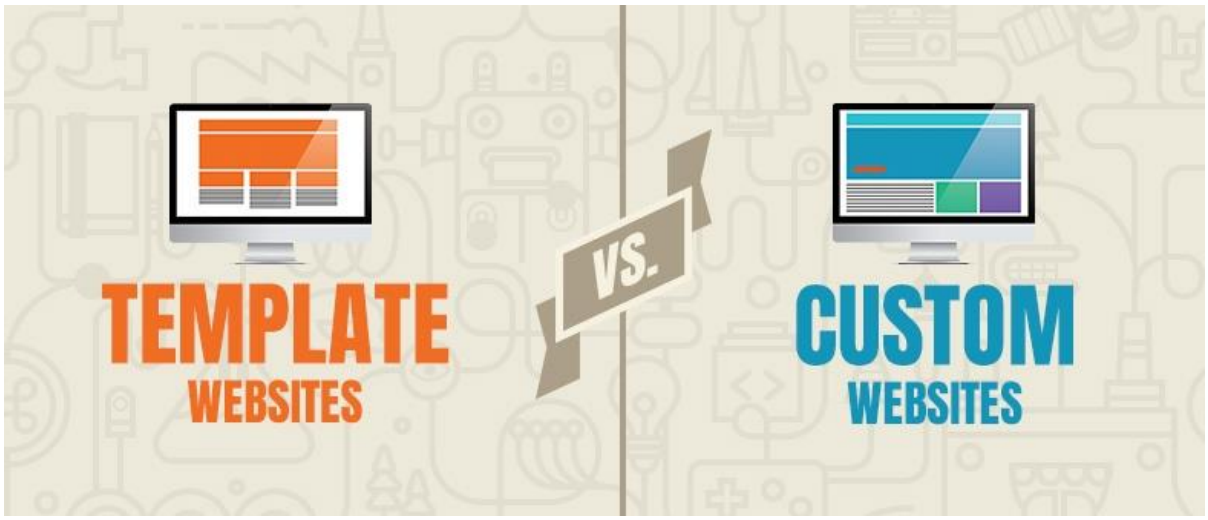
Back-end developers primarily develop and maintain the core functional logic and operations of a software application or information system. Typically, a back-end developer has expert programming skills in C++, C#, Java and other high-level programming languages such as Python, Pearl, PHP, Ruby, JavaScript, etc.

2 The differences between online websites creation tools and custom built sites.

There are two paths one can take when it comes to the initial step of designing a website: the decision to create custom website from scratch or to use a template website. Both options have their pros and cons and serve different purpose for different types of businesses. (csdesignworks, n.d.)

To help people in on the best possible option, there will be outlining the five key variables that people should be taking into consideration to help decide which one is right for create a website:

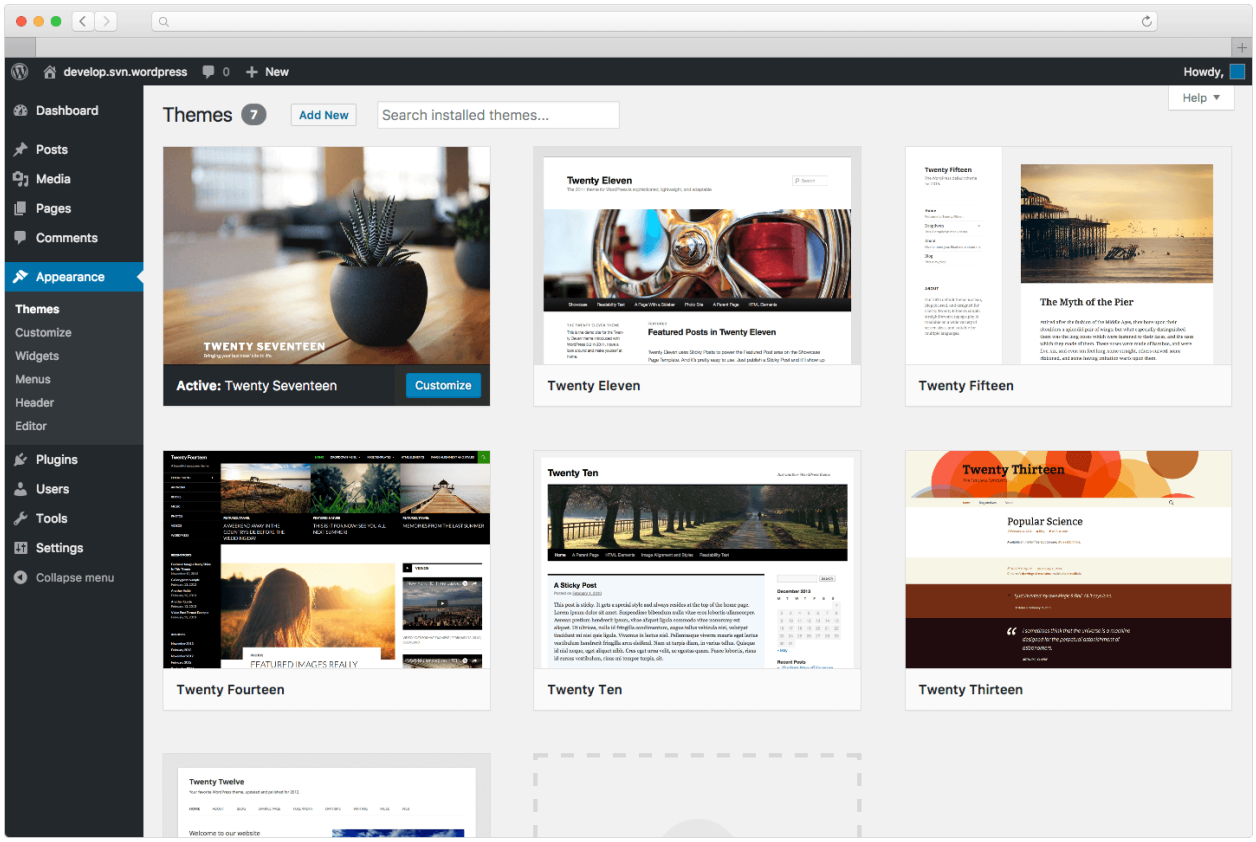
- Budget
- Time
- Brand
- User Experience
- Long Term Scalability



Picture 7: Template websites and Custom website

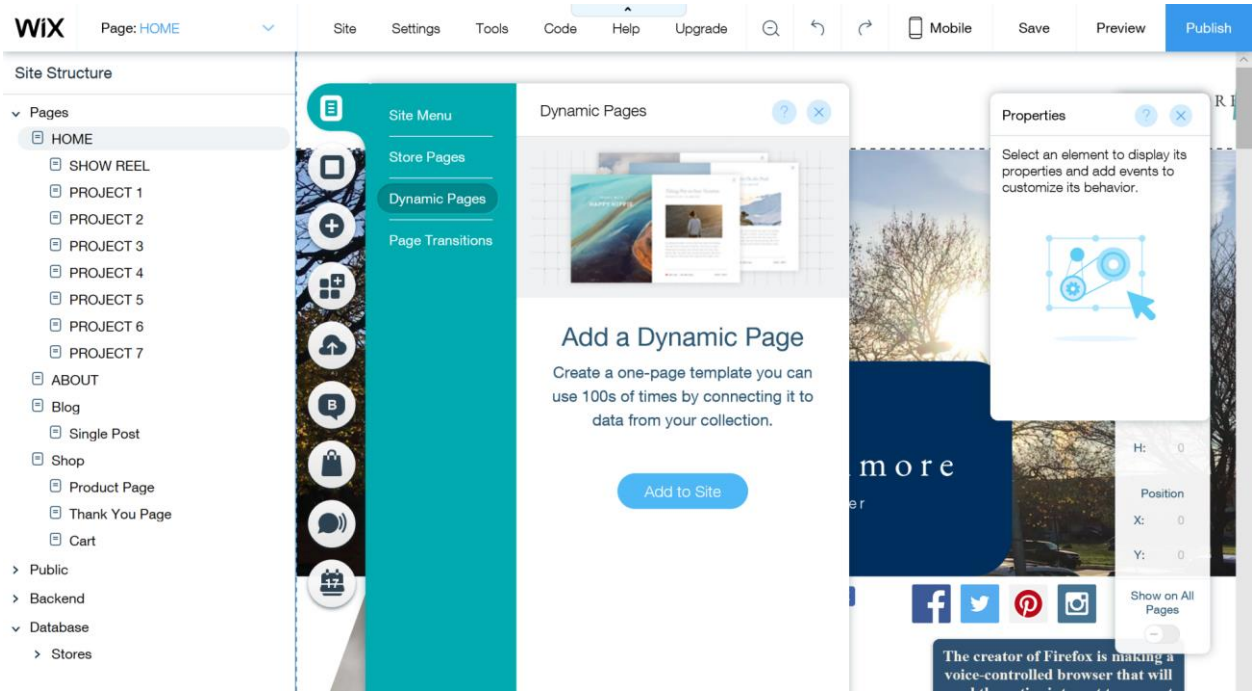
2.1 Using templates

The most common way to start a website would be to set up a WordPress site and apply a template bought from Theme Forest, Template Monster, or Elegant Themes. These websites provide a broad range of different types of website templates available for purchase. Not all of the themes/templates out there are 100% responsive.



Picture 8: WordPress interface

There are also services like **Wix** or **Squarespace** that also provide a theme and allow you to edit within the frame of the website. These can help you get a website up pretty fast and easily. However, there are limitations on the options that are available for more complex features such as forms or e-commerce functions.



Picture 9: Wix interface

One thing to keep in mind is that templates and themes normally come with built-in features so that the user can have more customizability but this ultimately weighs down the website in terms of load speed and **SEO** performance. It's very likely that you'll have to put up with features, designs, and functions that won't necessarily fit your brand or business. You could pay a web development company to fix it, but that's like asking a certified ford mechanic to take a look at your Toyota since the site most likely won't be written in the same code format. And even if the developer does fix the site to your liking… Once an update comes out for WordPress or that theme itself, your website will run into some display issues because of a former customization made by a developer.

2.2 Custom built websites

Custom built websites involve a team behind your business. It starts with a creative process to understand who is your target audience, who do you want to reach, how you want/need the website to function and how do you want to look on the internet.

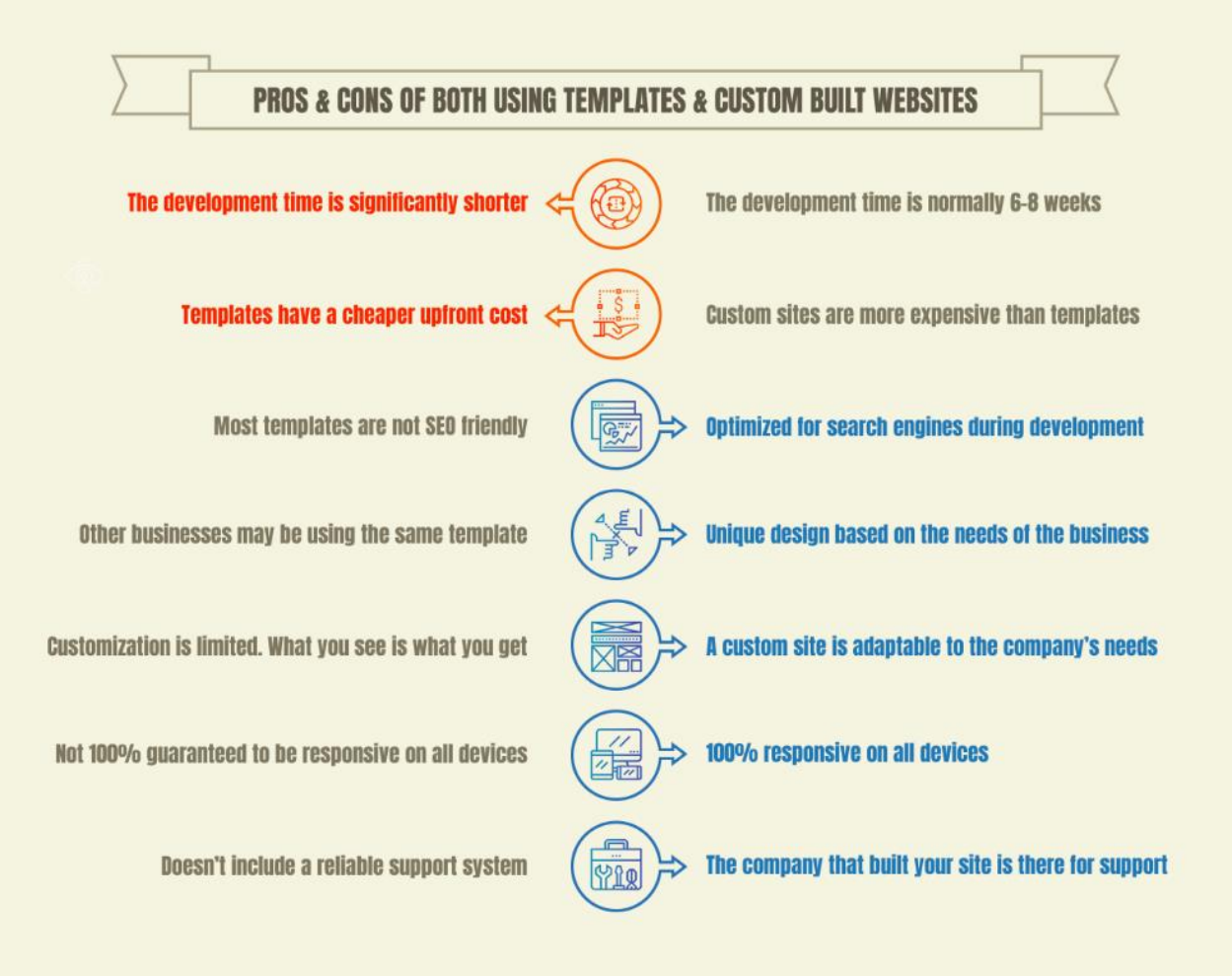
Picture 10: Custom built websites

SEO plays a big role in the success of any website. There are rules to follow when building a site and not all themes are capable of abiding by these rules. This is why custom built websites are so much more superior, the developer is able to optimize the site during creation to allow for google and other search engines to read each page to give the end user a more relevant search result thus, giving custom built sites naturally a better ranking on the system. This goes beyond keyword research.

Picture 11: SEO in custom built websites

Custom built websites tend to take longer than themes, this is because everything is being built and catered to particular business. It will save time in the long run because when its built from the ground up it's a lot easier to customize the site to ensure your site does exactly what you want it to do. Custom designs allow your site to be responsive across all devices (such as phones, tablets, laptops, etc.) & browsers. (atilus, n.d.)

2.3 Compare Using template and custom built websites



Picture 12: Pros and cons of template and custom built website

If you're on a small budget and need quick and simple website, especially for businesses just starting out, a template website is a valid option. However, if you have the budget, need a complex website that can be customized to your liking, and want to go the extra mile to ensure your business rises amongst the rest, then custom-built is the option for you. (csdesignworks, n.d.)

	Online websites creation tools	Custom built websites
Design flexibility	- Fast, easy to use and effective by providing Drag and Drop Interface so that reduce the timeframe. - Does not require any knowledge or skills in coding. - Provide various editable templates to customize. However, websites won't stand out as much from each other if having the same template. - Often also including hosting and domain name solutions.	- Require an amount of timeframe for both creative process and coding process - Require a decent knowledge and skills. coding (such as HTML, CSS, PHP, etc.). - Provide various frameworks on both Front-end and Back-end to customize. Flexible to design a unique website - Flexible to manage and choose hosting and domain name solution
Performance	- The website's bandwidth, which control how fast a website perform, are depends on the cost of hosting solution. - Does not provide any Back-end intervention.	- The website's bandwidth is depending on the cost of hosting solution. - Fully control and customize the Back-end technology to improve the site's performance.
Functionality	- Comes with various built-in features and functions. Sometimes does not including the wanted function. - Too much built-in features may weigh down the web's SEO performance.	- Web developer is able to optimize the site's features and functions to fit the requirements - Optimized for Search Engine with cleaner code and site's structure.
User Experience (UX)	- Lack of control over UX where any custom or added technologies are not possible to be installed as templates run on a structured system.	- Flexible to customize and optimize the site's technologies to capture leads or sell products.
User Interface (UI)	- Each template has its own Customizable UI, developers can custom the sites by working to find the template that fits the requirements. - Sites may not 100% works on all devices.	- Provide a unique UI that not only fits the requirements but also distinct the site from any other sites. - Sites are 100% responsive on all devices.

Table 7: Compare Online website and custom built

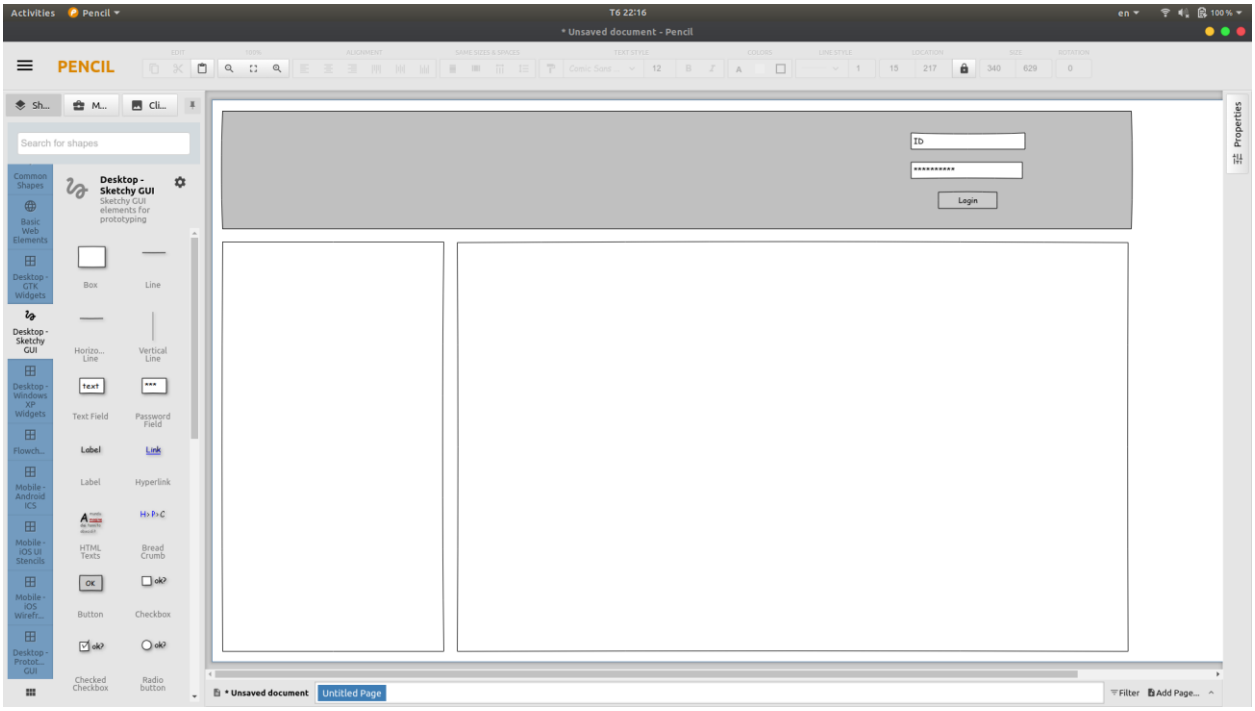
3 Evaluate a range of tools and techniques available to design and develop a custom build website.

Web development tools have come a long way in just a few short years. Those following tools and techniques are the most famous and most-used, with outstanding features and flexibility.

Design tools:

- **Pencil:**

Pencil is built for the purpose of providing a free and open-source GUI prototyping tool that people can easily install and use to create mock-ups in popular desktop platforms.



Picture 13: Pencil interface

- **Sketch:**

Sketch is a vector UI design tool developed by Bohemian Coding that works on MacOS.

Sketch provides a complete toolkit, built-in grid system, prototyping and can even export the designed file into HTML and CSS code files. Those features make the UI design much easier for web developers.

- **Adobe XD:**

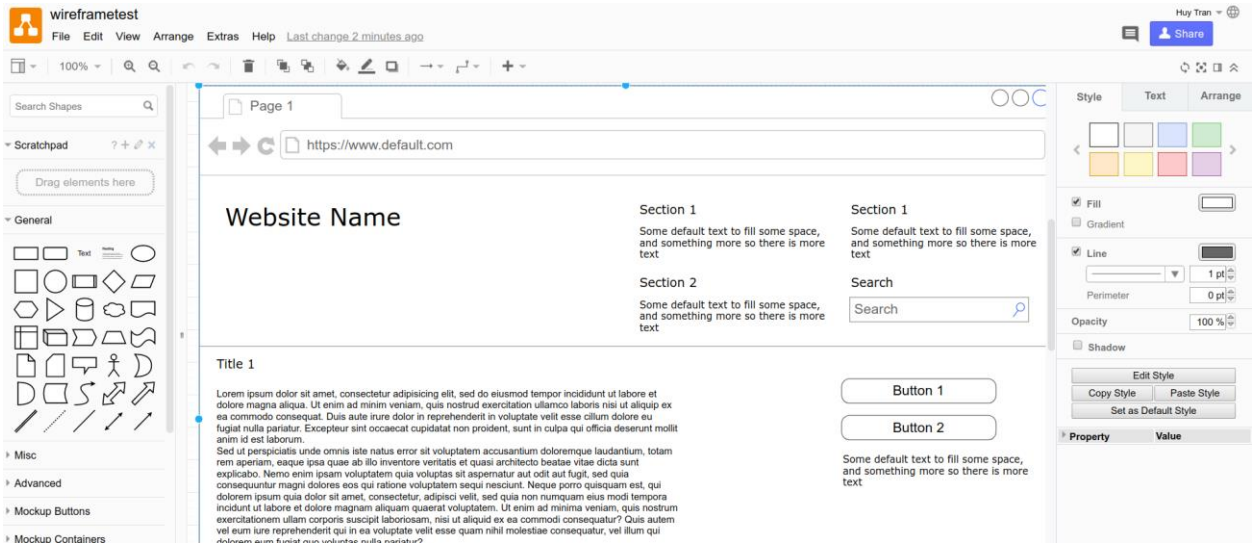
Adobe vector design and wireframe tool, Adobe XD, includes drawing tools, tools that enable developers to define non-static interactions, mobile and desktop previews. Adobe XD integrates with the rest of Adobe Creative Cloud components, such as Photoshop, Illustrator to makes collaboration projects run more quickly.

- **Draw.io:**

draw.io is completely free online diagram editor built around Google Drive(TM), that enables you to create flowcharts, UML, entity relation, network diagrams, mockups and more.

Your data is stored only in Google Drive, so no additional third-party to trust with your data.

draw.io can import from .vsdx, Gliffy(TM) and Lucidchart(TM) files.



Picture 14: Draw.io interface

Front-end frameworks:

- **Bootstrap**

This list would be woefully incomplete without the inclusion of the wildly popular front-end framework, Bootstrap. Created by Twitter developers and initially released in 2011, it's the most used open source framework in the world.

Like any effective front-end framework, Bootstrap includes CSS, HTML and JavaScript, or JS, components. It adheres to responsive web design standards, allowing you to develop responsive sites of all complexities and sizes.

Because it is updated continually, Bootstrap typically includes the latest and best features. For example, it added themes that met Google's material design guidelines shortly after they were published, and it was also upgraded to use Sass as a CSS preprocessor.

Pros:

Responsive web design support (can also be disabled if required)
Extensive documentation

Cons:

Out-of-the-box file size of 276kB due to excessive number of rarely used styles
Excessive number of HTML classes and DOM elements can be messy and confusing

- **Semantic-UI**

A relative newcomer on the scene, Semantic-UI stands out in a number of ways and is poised to become one of the most popular front-end frameworks out there.

This framework's main claim to fame is its simplicity. Because it uses natural language, the code is self-explanatory. Even those with very little coding experience will feel fairly at home working with this framework.

Another notable feature of Semantic-UI is that it is integrated with a dizzying array of third-party libraries. So much so, in fact, that you probably won't need to use any others. Therefore, the development process is a bit easier and more streamlined.

Pros:

Semantic class names make for a low barrier of entry, so even beginners can hit the ground running.

Small file sizes and minimal load times because you can load only the components that you need; each has its own JS file and stylesheet.

Versatile elements make for easy customization.

Cons:

Very large packages when compared to Foundation and Bootstrap.

Those with more complex design and development needs may find this framework lacking.

- **Foundation**

Created by web design company Zurb, Foundation is a highly advanced, enterprise-grade front-end framework that is ideal for developing nimble, responsive websites. Used on sites like Facebook, eBay, and Mozilla, it is also fairly complex and may not be suitable for newbies.

This features-rich framework supports GPU acceleration for smooth, lightning-fast animations and Fastclick.js for fast rendering on mobile devices. It runs on the Sass preprocessor and includes the Foundation-developed data interchange attribute, which lets you load lightweight HTML sections for mobile and "heavier" HTML sections for larger screens. For a comparison between Foundation and Bootstrap, read our complete article, Bootstrap vs Foundation.

Pros:

No style lock-in, so you have greater flexibility

Uses REMS instead of pixels, eliminating the need to explicitly state width, height and other attributes for each device

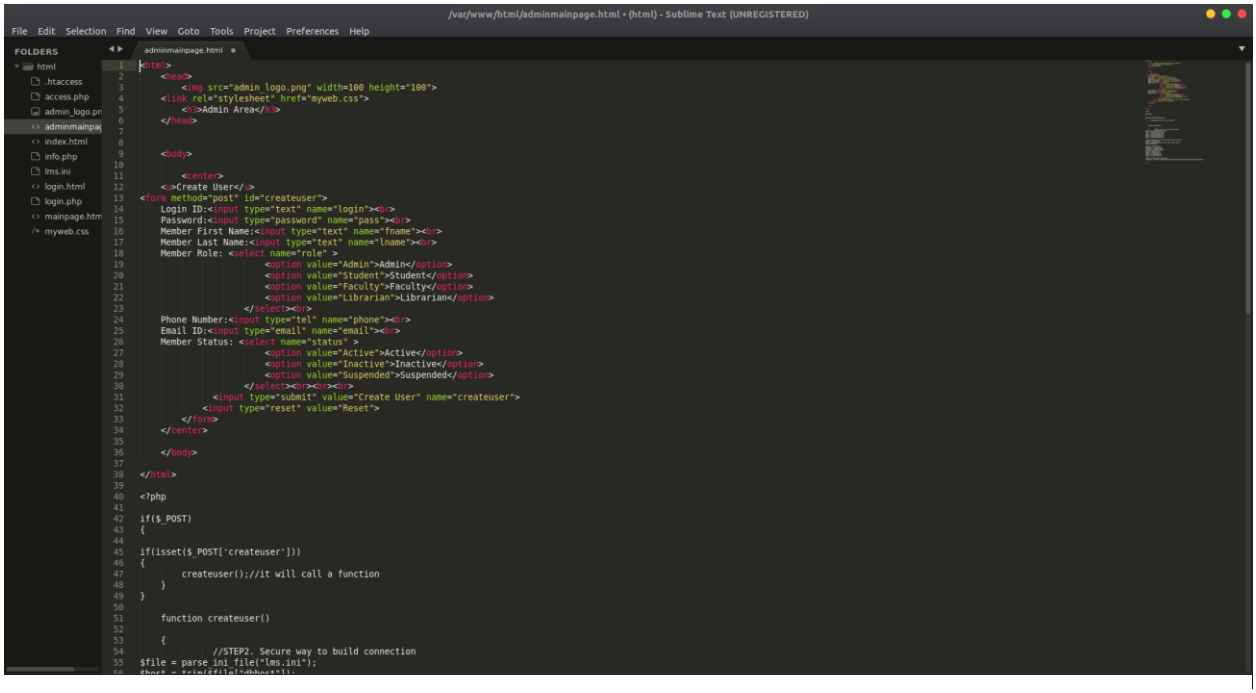
Cons:

Fairly large file size out of the box

A bit too complex for beginners (keycdn, n.d.)

- **Sublime Text**

Sublime Text is a sophisticated text editor for code, markup and prose. You'll love the slick user interface, extraordinary features and amazing performance.



Picture 15: Sublime text interface

Sublime Text is available for Mac, Windows and Linux. One license is all you need to use Sublime Text on every computer you own, no matter what operating system it uses.

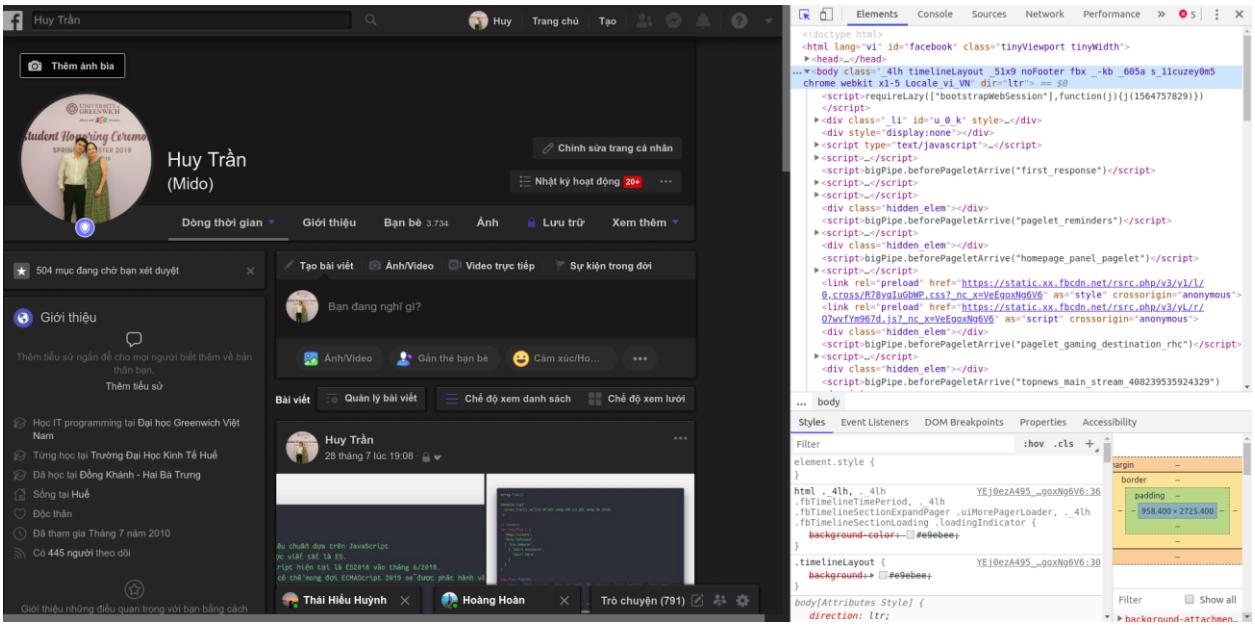
Sublime Text uses a custom UI toolkit, optimized for speed and beauty, while taking advantage of native functionality on each platform.

Sublime Text is built from custom components, providing for unmatched responsiveness. From a powerful, custom cross-platform UI toolkit, to an unmatched syntax highlighting engine, Sublime Text sets the bar for performance. (sublimetext, n.d.)

- **Chrome Developer Tools**

Wouldn't it be great if you could edit your HTML and CSS in real-time, or debug your JavaScript, all while viewing a thorough performance analysis of your website?

Google's built-in Chrome Developer Tools let you do just that. Bundled and available in both Chrome and Safari, they allow developers access into the internals of their web application. On top of this, a palette of network tools can help optimize your loading flows, while a timeline gives you a deeper understanding of what the browser is doing at any given moment.



Picture 16: Chrome Developer Tools interface

- **GitHub**

It’s every developer’s worst nightmare – you’re working on a new project feature and you screw up. Enter version control systems (VCS) – and more specifically, GitHub.

By rolling out your project with the service, you can view any changes you’ve made or even go back to your previous state (making pesky mistakes a thing of the past). The repository hosting service also boasts a rich open-source development community (making collaboration between teams as easy as pie), as well as providing several other components such as bug tracking, feature requests, task management, and wikis for every project.

Many employers will look for finely honed Git skills, so now’s the perfect time to sign up – plus it’s a great way to get involved and learn from the best with a wide array of open-source projects to work on. (careerfoundry, n.d.)

Back-end frameworks:

- **Ruby:**

Ruby is a language of careful balance. Its creator, Yukihiro "Matz" Matsumoto, blended parts of his favorite languages (Perl, Smalltalk, Eiffel, Ada, and Lisp) to form a new language that balanced functional programming with imperative programming.

Framework which has everything required to build database-backed web apps, and the pattern for MVC.

- **Express:**

> Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

With a myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and easy.

Express provides a thin layer of fundamental web application features, without obscuring Node.js features that you know and love.

Many popular frameworks are based on Express. (expressjs, n.d.)

- **AngularJS:**

HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.

Other frameworks deal with HTML's shortcomings by either abstracting away HTML, CSS, and/or JavaScript or by providing an imperative way for manipulating the DOM. Neither of these address the root problem that HTML was not designed for dynamic views.

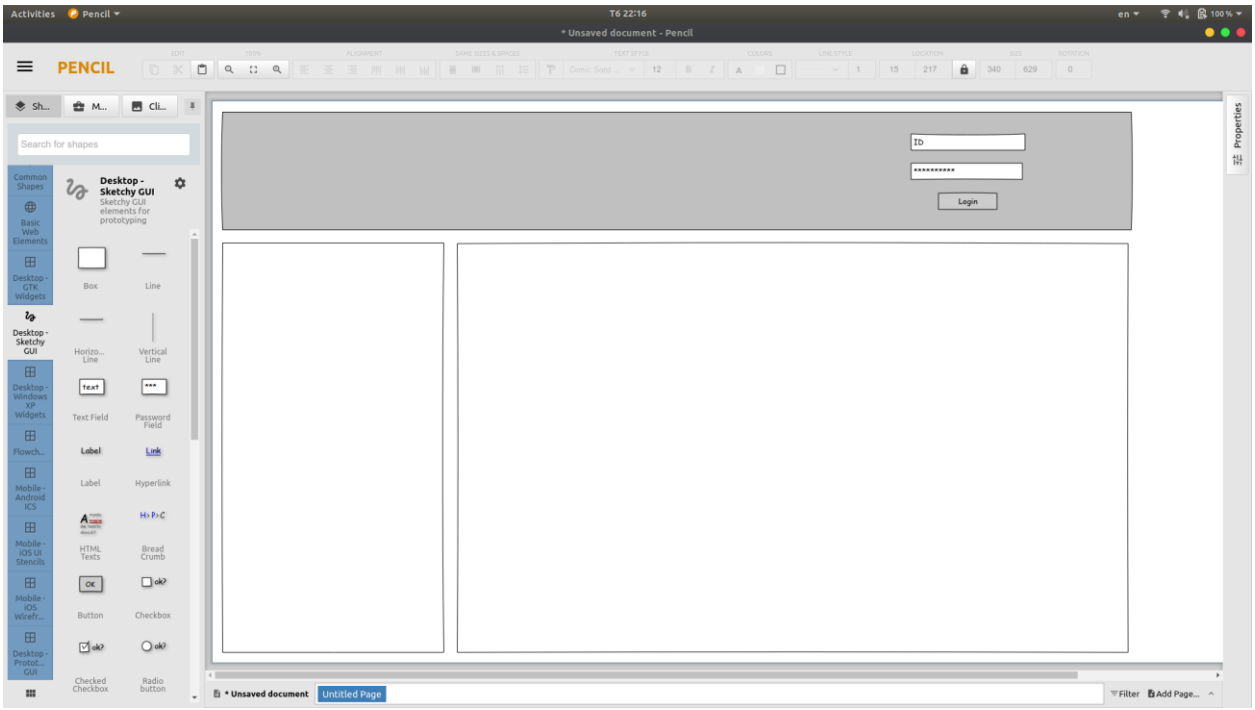
AngularJS is a toolset for building the framework most suited to your application development. It is fully extensible and works well with other libraries. Every feature can be modified or replaced to suit your unique development workflow and feature needs. Read on to find out how. (angularjs, n.d.)

III. CUSTOM BUILT WEBSITE DESIGN WITH TOOLS AND TECHNIQUES

1. Design tools:

Pencil:

Pencil is built for the purpose of providing a free and open-source GUI prototyping tool that people can easily install and use to create mockups in popular desktop platforms.



Picture 17: Pencil tools interface

With pencil, users can easy draw wire-frame and custom the websites.

Built-in Shape Collections: Pencil has even more shape collections included by default. The list of built-in collections now includes general-purpose shapes, flowchart elements, desktop/web UI shapes, Android and iOS GUI shapes. There are also many other collections created by the community and are distributed freely on the Internet. You can easily grab a collection and install it into Pencil with a simple drag-and-drop operation.

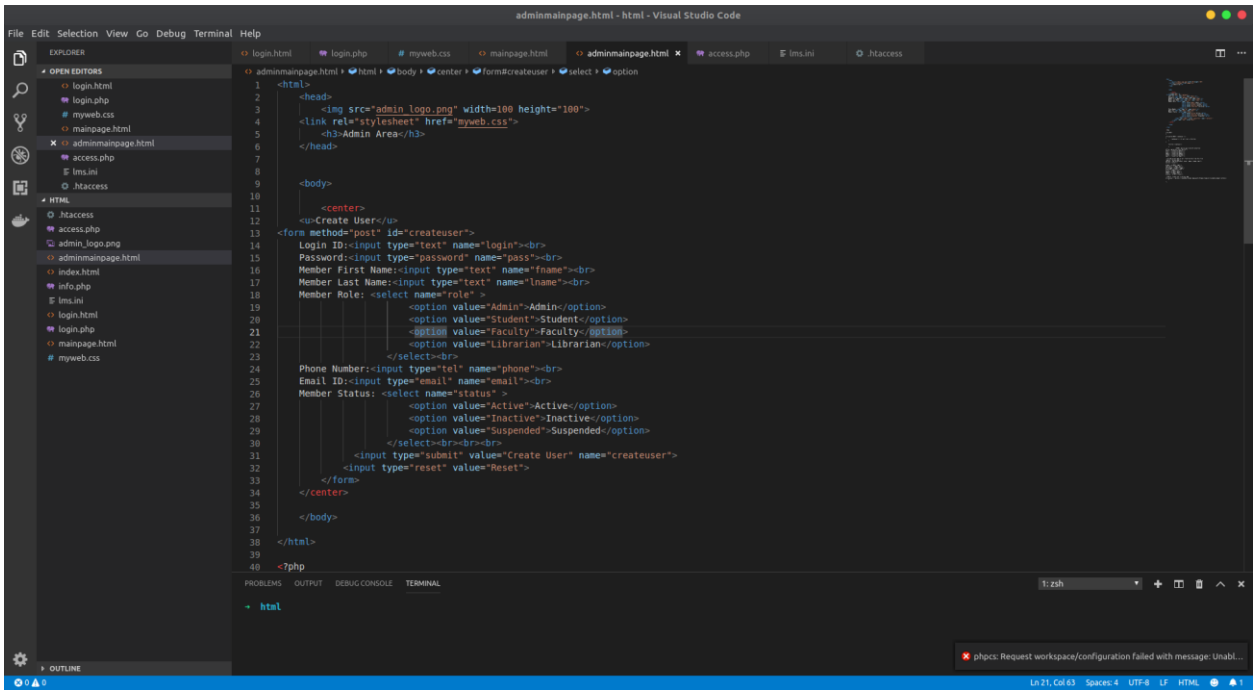
Diagram Drawing Support: Pencil now supports connectors which can be used to "wire" shapes together in a diagram. A collection of flowchart shapes is also available for drawing diagrams.

Exporting to Different Output Formats: Pencil supports outputting the drawing document into different types of formats. You can have your drawing exported as a set of rasterized PNG files or as a web page that can be delivered to the viewers. Pencil also supports exporting documents into popular formats including OpenOffice/LibreOffice text documents, Inkscape SVG and Adobe PDF.

2. Front-end:

Visual studio code:
Over the years, I have used a variety of different Code Editor; TextMate, Coda, Sublime Text, Atom, and each have different advantages and disadvantages. However, Visual Studio Code I was completely convinced until now.

Start by downloading the version for your system (Windows, Mac OS, Linux) at **code.visualstudio.com** (of course it's free).



Picture 18: Visual studio code interface

IntelliSense is a feature inside VSC that I wanted to try as soon as I first opened Editor. It's an improved auto-undo feature, syntax assistant and more than you'd expect from this feature. Moving the mouse over the CSS selector will give you an example of how you can use the corresponding HTML (very useful for beginners).

Multi-mouse: When you mark a code sample in your file, VSC will automatically mark all similar strings. You can press Command-D to add a pointer to the next section, or Shift-Command-L to add a pointer to all instances. Bulk editing, easily.

Emmet has been integrated into VSC. No installation or configuration is required, just start using it (with Tab is the default hotkey to activate your snippets) right after you open the editor.

Git: Other standard Git commands such as sync, pull, and so on, are available in the Git dashboard.

3. Back-end:

Server-side language: PHP, C#, Java, Python, Ruby.

PHP: Most web pages of World Wide Web format use PHP as backend. And this is also the language that will top any list of popular programming languages. PHP owns an open source community that is very active and can operate continuously on many platforms on UNIX, Mac and Windows.

Python: Python open source language has emerged as one of the most popular and important languages for developers. The increasing popularity has become a part of Python becoming an important programming language to learn. Thanks to easy syntax, this is an apt programming language that needs to be learned even for those who are not programmers.

Java: Java has long been dubbed the versatile programming language. Java is used for software development on desktop, web and Android. Even if you love new technologies, languages and frameworks, you can't ignore the importance of Java. Currently there are many Java-based frameworks. Spring framework is the most flexible framework used by developers.

Database SQL: MS SQL Server, MySQL.

MySQL: The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows. PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform)

4. Server:

AWS: With Amazon Web Services, you can get your website up and running quickly. Website applications like WordPress, Drupal, and Joomla! can be launched with a click, and pre-configured solutions from thousands of vendors in the AWS Marketplace are ready to start running when you are.

Platforms: Simple LAMP "Hello World" Application: The acronym LAMP refers to first letters of the four components of a solution stack (Linux, Apache, MySQL, PHP) composed entirely of free and open-source software, suitable for building high-availability, dynamic, websites capable of serving tens of thousands of requests simultaneously. The links below deploy LAMP in your AWS account using CloudFormation bootstrap scripts. (amazon, n.d.)

I used **LAMP stack** on a single **Amazon EC2** instance with a local MySQL database for storage.

CONCULISION

Hopefully by now you've got a little better of a grasp on the differences between the front-end and back-end in the web industry. It can be confusing topic, especially since there's not really an industry standard for what's always in the front-end and what's always in the back-end. However, there are also a lot of people who work in and understand both the front-end and back-end. Those people are often called "full stack developers". If you're still a little confused about the difference between the front-end and back-end, always remember that the front-end is related to the browser and everything sent to it. If it's got something to do with a database, then it's back-end related.

People who have skill in both frontend and backend development are often referred to as full stack developers. They have a full range of skills that can be applied to the user interface and everything that makes it work in the background.

Some people consider a full stack developer not as good as a frontend or backend developer. As developer, having both frontend and backend proficiency means more opportunities. You will be able to apply to more contract, part-time, or full-time employment positions. As a freelancer, you will be able to take on more projects without being limited to frontend only or backend only.

References

amazon. (n.d.). Retrieved from amazon: https://aws.amazon.com/websites/getting-started/start-your-project/

angularjs. (n.d.). Retrieved from angularjs: https://angularjs.org/

atilus. (n.d.). Retrieved from atilus: https://www.atilus.com/templates-vs-custom-built-websites/

careerfoundry. (n.d.). Retrieved from careerfoundry: https://careerfoundry.com/en/blog/web-development/7-essential-tools-for-front-end-development

cloudns. (n.d.). Retrieved from cloudns: https://www.cloudns.net/blog/what-is-dns/

csdesignworks. (n.d.). Retrieved from csdesignworks: https://www.csdesignworks.com/blog/2016/08/template-vs-custom-website

csdesignworks. (n.d.). Retrieved from csdesignworks: https://www.csdesignworks.com/blog/2016/08/template-vs-custom-website

ducanhplus. (n.d.). Retrieved from ducanhplus: https://www.ducanhplus.com/ten-mien-quoc-gia-cap-cao-nhat/

expressjs. (n.d.). Retrieved from expressjs: http://expressjs.com/)

frontendmasters. (n.d.). Retrieved from frontendmasters: https://frontendmasters.com/books/front-end-handbook/2018/what-is-a-FD.html

keycdn. (n.d.). Retrieved from keycdn: https://www.keycdn.com/blog/front-end-frameworks

ns1. (n.d.). Retrieved from ns1: https://ns1.com/resources/dns-types-records-servers-and-queries

ostraining. (n.d.). Retrieved from ostraining: https://www.ostraining.com/blog/webdesign/frameworks/

sublimetext. (n.d.). Retrieved from sublimetext: https://www.sublimetext.com/

techopedia. (n.d.). Retrieved from techopedia: https://www.techopedia.com/definition/12708/search-engine-world-wide-web

upwork. (n.d.). Retrieved from upwork: https://www.upwork.com/hiring/development/a-beginners-guide-to-back-end-development/

wikipedia. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Communication_protocol

wikipedia. (n.d.). *wikipedia*. Retrieved from wikipedia: https://en.wikipedia.org/wiki/Domain_Name_System