Final Year Project Report

Full Unit – Final Report

Advanced Web Development

Danielle Andriyanov

A report submitted in part fulfilment of the degree of

BSc (Hons) in Computer Science

Supervisor: Daniel O'keeffe

Department of Computer Science
Royal Holloway, University of London

March 31, 2023

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 11,801

Student Name: Danielle Andriyanov

Date of Submission: 31 March 2023

Signature: Danielle Andriyanov

Table of Contents

Abstract

The following report represents the website I have created and the learning outcome. Advanced web development was the project I was allocated to. Advanced web development involves building a functional website that uses the latest frameworks, large organisations and businesses create these applications to provide online services like video streaming, social media and information websites. Nowadays, various large projects are most likely to include web development within it, therefore this is a prime skill each computer scientist should have. However, with the online tools available, it is easy to learn if there is no previous experience. Web development is divided into two parts, Front-end development is coded using HTML, CSS and JavaScript allowing the user to load the web application and interact. Back-end development, which is the server part of the web application, controls and uses a database. It is written in languages like Java and Python. In order to present a fully functioning web application and meet the client's requirements, it is important to create a plan with a timeline and tasks to follow accordingly as it can help with later problems that may occur. When milestones are not reached on time or are missed, a delay is caused to the project and in some cases, this costs money as well. To avoid as many challenges as possible that occur during the process of web creation, it is strongly suggested to follow a specific life cycle.

The Lifecycle of website development:

- Discovering and researching:

The first step of the cycle is an extremely important one where the requirements and risks are written and agreed upon with the client prior to starting the project. The key is to gain a shared understanding of the client's requirements in order to proceed to the next step. Failing to do so, may result in bigger problems later in the project.

- Planning:

The second stage of the cycle is where the agreed requirements are documented and the steps to how to make the website, which kind of people are in charge, and goals are defined in this document. This step allows for predicting the outcome of the requirements (the website).

- Design and development:

The third stage of the cycle is where the design is chosen. Such as the kind of images, layout, colours and fonts that should attract customers and be user-friendly. To create the design the targeted users are taken into consideration to fulfil their needs. When the design is agreed, the development process begins where a full-function and interactive website is created using programming languages like HTML, CSS and JavaScript.

- Testing:

The fourth stage of the cycle is where the website is tested. Although there are other kinds of tests done during the process of development there are different ones at the end of the development which will be explored later in this report. Every page is thoroughly tested for its content and functionality in order to ensure there are no errors that can come up after the website is launched.

- Launching:

The fifth stage of the cycle is where the web application is launched. It is published online and available for users to browse and use it.

- Maintenance:

The final stage of the cycle is where the website undertakes maintenance. After it launched there should be a maintenance team as technology evolves and requirements change the web application must stay up-to-date. It also checked for errors and fixes bugs during the time when the web application is online. also, the maintenance team ensures the content on the website is accurate if it is an online shop stock always changes or new products are added, therefore it should be updated.

This report will present the outcome of the website I have created. It is an online shop that sells coffee bean bags using HTML, CSS and JavaScript. It has a simple structure which will be analysed and discussed throughout this report. It will talk about why I chose specific design patterns, how it was written, and the software engineering process behind the website. Creating and developing a website from scratch is an exciting opportunity as it allows expressing creativity through the design process that reflects you personally and develop and learning new skills as it requires a combination of different programming languages. The personal skills developed from creating a website are design skills which make a website visually appealing, expanding knowledge in coding, expanding the knowledge in writing and also project management skills. These will all aid me in the upcoming future when I start my career. The reason behind why I chose to make a shop specifically about coffee is due to the passion I have for coffee. One day I am hoping to export my own and sell it in a shop, therefore this is a great opportunity to learn more about how to create an online shop in the future.

This report is divided into seven chapters where each chapter discusses a different topic and stage. In the first chapter I will be providing an introduction to this report speaking about aims and goals, objectives, survey of related literature and the reasons behind some of my designs. In the second chapter I will expand the state or art and design patters used in my web application. In the third chapter I will provide methodologies and go in depth about testing stage as I mentioned earlier and provide documentation. Then in the fourth stage I will be discussing the end system development that includes architecture, featured, running it, work logs and future enhancement. In chapter five I will provide information of professional issues I have encountered and self-evaluation. Then in chapter 6 bibliography will be provided, all the websites, articles and journals that I used to aid me in writing this report, using Harvard style. Lastly the final chapter will provide a timeline of the project development and appendixes.

Project Specification

The project specification is taken from the final year project list located in moodle.

Aim: To design and implement a web application of your choice.

An example of a service is an online store that provides the ability to authenticate, post items for sale, search, choose items to buy and perform payments.

The design must follow an N tier architectural pattern consisting of web browser interface, web and application server layers and a database.

Must include application logic and user interface using existing web framework. A rational database schema equipped with sound constraints.

It is expected that prior to selecting the concrete technologies to be used, an evaluation of the state-of-the-art options will be performed along a number of comparison criteria (e.g. overall set of capabilities, support for aspects such as concurrency, security, modularity, performance, UX practices). The student is expected to (i) adequately establish such criteria (ii) investigate and reflect on a number of candidate technologies and (iii) appropriately compare and justify the concrete choices made.

Chapter 1: **Introduction**

1.1 The Problem

Advanced web development is the development of any kind of website using different frameworks that are available today. It is crucial for any business to scale its digital growth by increasing the number of customers getting exposed to and subscribing to its content, as a fully functioning website includes information about the company, support, and goods and services for sale. Anyone could build a website using online resources available on the Internet, however, it imposes major challenges such as performance, design, compatibility, security, and maintenance.

Performance: users get frustrated easily and therefore leave websites that take a long time to load or are hard to use. The common performance issues currently challenging web developers are poorly written codes, slow servers and issues in handling large amounts of traffic that can lead to websites crash. There are free tools available online to resolve these kinds of challenges.

User Interface design: to achieve a high-quality website the deliverable user interface should be smooth and easy to use. This can increase the number of users accessing and browsing the website which should be a convenient and easy process for the user.

Compatibility: The website should be able to run smoothly on any kind of browser, platform and device but it can become extremely challenging to do so without performing testing.

Security: websites that have weak security are more prone to cyber-attacks therefore it is a top priority to ensure the security level is high and data is stored safely.

Maintenance: many websites that go live have issues that can only be discovered at this stage, as a result even when a website is live, it should continue to be supported and receive appropriate maintenance.

Building a robust web application arises difficulties and challenges therefore it is important to plan ahead, make a realistic project plan with different milestones and time to completion, and use free online available tools and strategies to ensure the best delivery result of a web application.

1.2 Aims and Project Goals

The aim of this project is to create a web application for an online shop which will be selling coffee beans. The website includes 6 different pages. Firstly, a home page which is the introduction to the shop displays a summary, images and quick links to best sellers and subscriptions. Secondly, the shop page where different types of coffee beans are displayed. Users will be able to click on the image leading to a different page including information about the product the price and the possibility to add the item to the cart. Next, the subscription page will offer users to sign up for a coffee subscription, there will be an option to choose how often it should be delivered (once a week, one in two weeks or once a month), which kind of coffee and the kind of grind. This will allow users to save money as this option will be cheaper than a usual coffee bag. Then, the about page informs the users of the story about the coffee. Also, a help page where users can find frequently asked questions. There are 4 different categories: shipping, ordering, subscription and coffee. Under each category, a number of questions are displayed. Each question has a drop-down answer one clicked. Lastly, the contact us page displays useful information on how to make contact, on the left side there is information like email, phone number and address and on the left side there is a box form where the user can input a name, email address and the message to contact customer service team.

When an item is added to the cart, there will be a cart logo that can be clicked on to view all the items in the cart. Then on the checkout page, a user is able to fill out details and shipping information, choose a delivery method and make a payment.

Goals:

- Complete the project on time.
- Have all the pages fully functioning and working.
- Test pages.
- Have a friendly user interface that allows the user to enjoy browsing through the website.
- Learn how to add a cart and purchases on the website.
- Write the project report with all the sufficient requirements provided on moodle.
- Have the website working on smaller screens like phones and when the size of the website decreases the content will too.
- Enjoy the experience and gain new skills.
- Create a subscription plan page where users chose the kind and every how long.
- Create a login page.
- Finish the project on time.
- The website communicates its purpose

1.3 Survey of Related Literature

Advanced web development just like any other technology keeps evolving at a rapid speed. There are numerous literature topics available to explore and learn about the latest technology trends, new technology and best practices. In advanced web development, the most important topics as of today are responsive design, web accessibility, web security and the use of modern web frameworks. These topics will be further explored in more depth below.

Responsive design:

Responsive design is essential in any type of web application. It is a development approach that allows web pages to adjust dynamically to the size of the device being used. This could be a mobile device, tablet, or computer. When developers create a website they should aim in making it responsive and easily accessible on different browsers without the assumption that users will only access it through a computer. The design of the website like colours, fonts and layouts will be viewed differently and might not be accessible at all on different browsers if it was created with the intended purpose of a computer (Almeida and Monteiro, 2017).

Today, users use different screens at the same time and access websites with a device that is convenient for them at this specific moment. For example, a user usually uses the phone to browse through the internet but the phone is currently charging and the user watches tv but wants to search for something therefore, he/she will use the browser on the tv. As a result, it is significant for websites to adjust their sizes according to the device. See figure 1 for an example of how web application changes in size

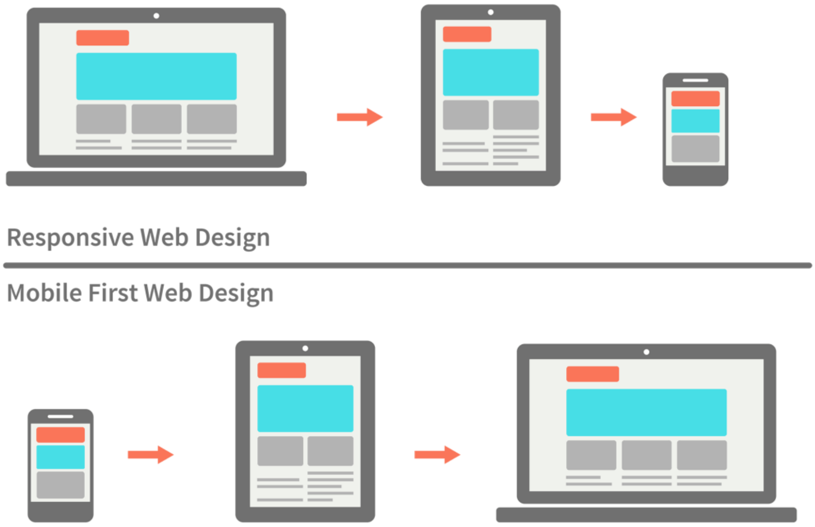


Figure 1.

To make an application responsive a few lines of code should be added to the end of the file. This code provides instructions and parameters to activate the changes and make them responsive. According to statistica mobile phones are the most used devices for web browsing. It is estimated that around 92% of users access the internet using a mobile device, see figure 2 (Petrosyan, 2023). Therefore when making a website the focus should first be on creating for browsing on the phone and then adding the needed code to make it work on bigger screens.

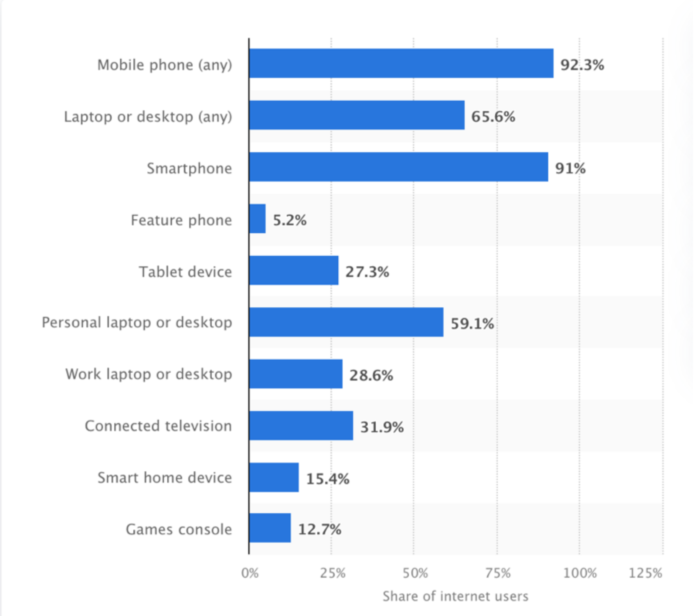


Figure 2.

Having a responsive application is not just improving the amount of traffic but has other advantages. Google's algorithm can increase an organisation's website's rank if it is fully responsive and user-friendly, this can allow more visits for users and increase growth. It also makes the website easier to use, if a customer was satisfied with their visit and did not encounter any problems, they are most likely to return again. Responsive design improves the speed of the pages making them load a lot faster. It is important to continue adapting new approaches for responsive design which can adapt to any device to continue the growth of the website.

Web accessibility:

Some websites remain completely inaccessible to some parts of the public due to their disabilities. The reason they are not able to have access is developers have little knowledge of how to expand accessibility. There are two principal ways available to improve this, both are equally important by using assistive technology and adopting good practices. This is an important topic spoken today as many people with disabilities feel discriminated against by not being able to access certain information as it was not designed for their sector. There are other factors contributing to it like economic, technical and legal issues.

Assistive technology is devices designed specifically for people with disabilities like vision hearing, communication and mobility loss. These devices aid in accessing websites for people with difficulties. An example of aiding devices is screen readers, keyboards, and screen magnifiers (see figure 3). When a developer develops a website it is important to research these assistive tools to find ways on how to make the website works with these devices and allow people with disabilities to browse as well as people without disabilities.

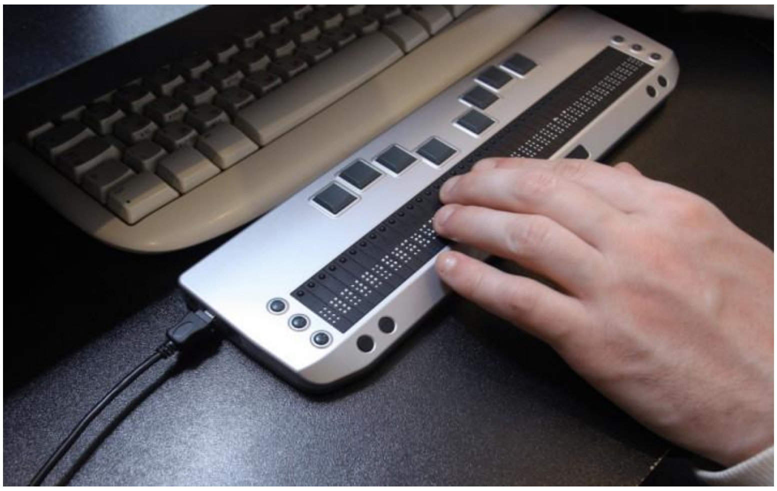


Figure 3.

Providing this service has proven that it provides capital growth. The more users access the website the more popularity is achieves and this can bring higher sales. Search engine optimisation also increases therefore the website will be found mor easily on the internet. Also, it increases the useability, users are more likely to visit the website again. It can also save major costs like reduced maintenance, reduced server capacity and decreases costs of upgrades.

Making a website accessible to every sector of the public will mainly bring advantages in finance, user experience and cost reduction. Therefore, it is crucial for developers to take this into account when creating a web application.

Web security:

As everything moves to the web, there are numerous services available like transactions, trading, and personal information stored online. This brings a threat to security. There are many groups which use different techniques like ransomware and viruses to gain unauthorised access to this information therefore the importance of web security is crucial. Businesses are at high risk of malware attacks if security strategies for their websites are not implemented. Attackers usually use a program which can display websites with vulnerabilities. These vulnerabilities open the door for the attacker to get in and place its malware.

When a website suffers an attack it imposes a bad reputation, financial loss and loss of trust. A user who had their personal information stored in a website which was attacked, would likely not use to website again although the breach was fixed due to a fear that if there was a vulnerability that was not addressed this could happen again in the future.

There are basic security components that aid in website protection,

SSL Certificate: this protects any data transfer process when it transfers to the server from the website. For example, google labels websites as insecure if they do not have this SSL certificate and ask the user if they are sure they want to enter the website although it is not fully secured.

Web Application Firewall: this aid in stopping automated attacks by bots. Bots can have the capabilities of searching through the web and detecting their vulnerabilities and cause a ddos attack which usually crashed the website by increasing its traffic significantly fast by using other users' phones without their knowledge.

Software updates: when software is not updated in time or is delayed, the website is at large risk of being compromised.

It is highly necessary for website developers to ensure their websites use some of the security components available today as every component strengthens the security to avoid future damage from compromises.

Modern web frameworks:

Web frameworks provide additional features to achieve desired structure for a web application helping in work and time reduction. It also aids in formatting code helping it become more readable to the person working on it next.

The frontend part of the web application is the interactive part of the website where the user engages, clicks, and browsers through the website. The most common languages used are HTML, CSS and JavaScript. The following frameworks can be used for frontend:

- Vue
- Angular JS
- Angular 2+
- React
- Ember

The backend part uses Pyton, Ruby and Node JavaScript mainly to connect the frontend code to the server. The following frameworks can be used:

- Spring MVC

- Django
- Flask
- Ruby on Rails
- Express

Choosing which framework should be used, depends on what kind of language is used, therefore research should be done before using any for best optimisation.

1.4 Rationale

As a web developer, you receive project requirements which you must act upon them to achieve the desired outcome which is a web application. There are a number of factors that influence the decision of website development, which is user experience, technical requirements, security, robust content management capabilities, payment procession, marketing strategy and maintenance. These factors will be discussed deeper below.

Firstly, User experience is a dominant factor when developing online stores. The store would need to be intuitive, visually appealing, and easy to navigate. Research must be made about the targeted customers to understand what kind of needs and preferences they may have to maximise traffic. This data will help in the decision-making of the design which includes colours, fonts, layouts, images, and size. It is vital to use a design framework like Bootstrap to ensure the website is accessible across all devices. User experience is the whole package which makes the customer experience, if it is good then the experience would be positive, and the user will come back again. It is important to make it as simple as possible as users get easily frustrated when a website doesn't load fast, is not accessible and is not appealing to the eye (Ferrer, 2022)

The next factor to consider is the technical requirements of the online store. An online store must integrate specific functionalities like a shopping cart, product search and check out process to handle complex transactions. There are content management systems available like WooCommerce to handle these requirements.

Web security is another critical consideration when developing an online store. Developers must implement security measures like HTTPS, two-factor authentication, SSL certificate, firewalls and more to protect users' data and prevent unauthorised access from attackers that could potentially cause major harm. It is important to also implement strict password policies, so it is not easily guessed.

An online store also requires robust content management capabilities to ensure that every product available on the website is up to date, the stock is managed, prices are changed, and shipping is handled. Usually, shipping platforms are used that provide automated management, creation, and printing of shipping labels and importing tracking information. This software is great for time management (Kohler, 2015).

Another important factor to consider is payment processing. Developers have to integrate payment gateway functionalities to allow customers to pay securely and efficiently. PayPal, Stripe, and Braintree provide these services. This is usually don’t by using software that accepts card payments when customers make a purchase (Kohler 2015).

Online stores also require an effective marketing strategy to increase traffic and sales. Developers have many options like customer reviews, recommendations and email marketing which can be integrated to increase user's visits and ensure their loyalty. It is also vital to implement search engine optimisation to increase the visibility and rank in the search engine.

Finally, developers must consider having a maintenance team for the online store after the website goes live. As the store grows, they must ensure that the website can handle the increased traffic and data volume. Developers may choose solutions like AWS or Azure which is a cloud hosting system offering infrastructure and deployment options.

In conclusion, developing an online store requires deep consideration of the elements mentioned above. By implementing these techniques and approaches, the website will be fully optimised for performance security and user experience.

Chapter 2: **Web Frameworks**

2.1 State of the art of Web Development

Due to advancements in technology, the state of the art of web development is constantly evolving. There are constant changes in user preferences and behaviour and a growing demand for web applications and services. Here are the key technologies:

Responsive design: as there is an increase in the use of mobile devices, users access the internet more often than a phone therefore a responsive design is essential for delivering a consistent user experience across different screen sizes.

Serverless architecture: these types of computers allow developers to build and run applications without the need for managing infrastructure resulting in faster development, deployment and scalability.

Progressive Web Apps: these applications are installed on users' devices and offer the ability to use applications without the internet.

Artificial intelligence: natural language processing and machine learning are integrated into web applications to provide personalised experiences and automate repetitive tasks.

Microservices: its architecture can break down applications into smaller independent services making it easier to maintain complex web applications.

Security: as the threats of cyber security and breaches are growing, we developers are focusing more on security and implementing best practices like secure coding and authentication.

Web frameworks: there are many types of web frameworks available today, and each has its own unique features advantages and disadvantages.

- Djanjo: this is a high-level python web framework that emphasises rapid development, pragmatic design and doesn’t repeat yourself principle. It includes the following build in features: ORM, URL routing, templating engine and a robust admin interface. Django gained its popularity due to the large community it has. The way it was designed aids in faster cleaner and more practical design for website development. It comes with common libraries to build common functionalities, includes a built-in administration interface, and no imports are needed. A large number of developers in the community release their projects as open-source packages for others to use.

- Flask: a lightweight Python web framework allowing developers to have the flexibility and is recommended for beginners. It is mainly used for small to medium applications and includes features such as URL routing, template rendering, and support for various extensions. It doesn’t require large amount of learning to get started with it, it is simple but extensive.

- Ruby on Rails: a popular web framework written in the Ruby programming language. It follows the convention over the configuration principle. it includes features such as an ORM, URL routing, and built-in security measures. It is an open source and used for web applications, data processing and automation. Due to its high flexibility, it can be run on any operating system available today. It can also provide default structure. It comes with a built in code that could be played around with to reach the desired goal.

- Laravel: is a PHP web framework that is known for its modern syntax and easy use. It includes many features such as an ORM, URL routing, template engine, and built-in authentication. Due to access to this library developers can build a robust web application a lot quicker and reduce the amount of coding needed.

- React: is a JavaScript library for building user interfaces. Often used in conjunction with other web frameworks such as Express.js or Rube on Rails. Allows providing tools for managing and handling user interaction. It was first created by Facebook before it became an open source. Their best feature is components that show data, changes to data, data updates and generate new data.

- Express.js: a popular Node.js web framework that is used for building web applications and API also single page, multipage and hybrid web pages. it has a minimalist approach providing a flexible foundation. It is a layer which is built on top of Node aiding in servers and routs management.

2.1.1 Advantages and Disadvantages

	Advantages	Disadvantages
Djanjo	• Implemented in Python • Better content management • Better connectivity • Batteries in framework • Fast processing • Scalable • High security	• Cant be used for small projects • Can lead to slow websites • Lack of convetion
Flask	• Flexible and easy • Allows unit testing	• Scaling time • Cant use third party modules
Ruby on Rails	• Time efficiency • Active community • Helpful tools and libraries • Cost effective	• Shortage of flexibility • Performance time • documentation
Laravel	• simple coding • secure	• high cost • limited support

	• data mitigation • easy to learn	• slow • frequent updates
React	• high performance • easy to learn • Facebook support	• Lack of documentation • Development pace • Incomplete tooling set
Express.js	• Code sharing and reuse • Speed and performance • Large number of free tools	• Call-back issues • Lack in library support • Unstable API

2.1.2 Concurrency

Concurrency is the ability to execute a couple of problems or tasks at the same time. Various approaches can be utilized for executing concurrent programs or designing programming languages that support concurrency. These include creating individual operating system processes for each computational execution or implementing a group of threads within a single operating system process to carry out computational processes. It is divided into 4 levels: instruction, statement, unit and program.

In web frameworks, concurrency utilizes packages that offer a robust and adaptable structure of threading utilities that perform well, including thread pools and blocking queues. This package aids programmers in creating utilities more efficiently instead of having to manually create them, similar to how the collections framework eliminated the need for constructing data structures.

- Django: when developers run a couple of processes in Django at the same time, concurrency issues will occur. However, this can be prevented by using select_for_update to fetch the quiryset in order to make it locked till the process is complete.
- Flask: it allows to handle many concurrent requests.
- Ruby on Rails: can achieve concurrency by using only one thread.
- Laravel: it is possible to make up to 1000 concurrent executions.
- React: it is possible for concurrency however reach had its architecture more complex.
- Express.js: can handle 15,000 requests at once using a single thread

2.1.3 Security

Website security is vital for any application to keep the data stored securely and prevent a future attack that may occur. The most frequent attack is a denial of service when there is a large amount of traffic resulting in a website crashing or a ransomware attack where data is stolen and to receive it back an attacker requests ransomware to exchange for the data back. To prevent these types of attacks, website security aims to protect websites from unauthorized access, modification, destruction, or disruption. This requires a comprehensive approach that considers the design of the web application, web server configuration, password policies, and client-side code. However, if you use a server-side web framework, it's likely that it will come with robust defence mechanisms against common attacks. Other attacks can be prevented by enabling HTTPS in your web server configuration. You can also use publicly available vulnerability scanner tools to identify any potential security gaps. While website security may seem daunting, there are effective measures you can take to safeguard your website and protect its users.

Django: queries constructed using query parameterization therefore it is protected against SQL injections. There is also protection against cross-scripting which is when attackers inject client-side script into the browsers of another user. Also, there is clickjacking protection by a supporting browser.

Flask: flask can have protection against cross-site scripting however it security doesn’t cover everything, users should still be careful in other places like generating HTML without the help of Jinja2 or calling Markup on data submitted by users. Another complex attack is cross-site request forgery and flask does not offer protection for it.

Ruby on Rails: there are many security issues such as CSRF, SQL injection, command injection, XSS and authentication vulnerabilities. Ruby on Rails does not offer protection against these issues, therefore users should take extra precautions to avoid being attacked.

Laravel: one of the most secure web frameworks it protects against various attacks and takes care of vulnerabilities as soon as they are addressed.

React: react offers high protection as well against common attacks.

Express.js: a secure framework as well however third-party open-source packages in the system may contain malicious code therefore users should be careful.

2.1.4 Modularity

Modularity is an essential aspect of web frameworks that involves breaking down a web application into smaller, more manageable parts. These small parts and modules are small scripts written to perform a certain task. This allows developers to work on different components simultaneously, facilitating a faster development process, easier testing, and more efficient deployment. By using a modular architecture, developers can maintain and update individual modules without impacting the rest of the application.

Web frameworks are designed already with modularity function providing developers with tools and best practices for building modular web applications. With these frameworks, developers can focus on building the core features of their applications while relying on the framework to handle the underlying infrastructure and management of the modules. Modularity makes web development more scalable and efficient. By creating independent modules, developers can isolate and fix issues in specific parts of the application without affecting the entire project. This also makes it easier to maintain and troubleshoot the application, especially as the application scales and becomes more complex.

The modular design in Django includes a minimalist design template aiding in the module creating and saving developers a lot of time. Open source becomes easier to access as codes for models can be used without understating how it works or its technology and the integration process is a lot simpler due to its design.

Flash modularity is used by programming beginners to build single-module apps.

Ruby on Rails is one of the greatest for modularity use as it uses interpreted language like reopening classes and adding more logic into them, bundler which is a dependency management tool that requires a download, deface which could extend rails views, rail engines which is a new feature that is integrated into ruby on rails.

Laravel uses a package to manage modules. It includes views, controllers, and models.

React uses a component structure for modularity. This requires more time and effort as import and export are used to access code in the files. This is due to every component containing its own file.

In express.js, developers may import their own code and files and express js uses module pattern that exports everything to the main files.

In conclusion, modularity is an essential concept in web frameworks that enables developers to build efficient and scalable web applications. By using a modular architecture, developers can focus on building individual components while relying on the framework to handle the underlying infrastructure and management of the modules. Every framework works differently and has its own pros and cons, therefore developers must consider which framework will work best for their application.

2.1.5 Performance

Performance is a crucial aspect of web frameworks that directly affects the user experience. It refers to how quickly a web application responds to user requests, loads web pages, and performs other operations. A fast and responsive web application is essential for ensuring a positive user experience and keeping users.

The following frameworks Django, Flask, Ruby on Rails, Laravel, React, and Express.js are designed with performance in mind, providing developers with tools and best practices for optimizing web applications. Performance can be improved by developers by reducing the number of database queries, caching frequently accessed data, and minimizing the size of web pages. To optimize performance, web frameworks use various techniques such as server-side rendering, lazy loading, and code minification. Additionally, developers can use profiling and benchmarking tools to identify problems and areas for improvement. Performance is a critical factor in web application development, and it requires continuous monitoring and optimization. Developers need to be aware of performance issues and actively work to improve the speed and responsiveness of their web applications.

To reduce overall load time a strategy that reduces files sizes is used. This is completed by reduce the number of HTTP requests made. Load time affects users and their experience. If the desired file takes long time to download since it is large in size, users are likely to get frustrate and download the file from a different website.

To make an application smooth, different strategies are used to make scrolling up and down fast, when buttons are clicked, they respond fast and animation is running smoothly. These are all factors that make performance better.

In conclusion, performance is one of the critical aspects of web frameworks that directly affects the user experience. Web frameworks aid developers by providing tools and best practices for enhancing performance, such as decreasing the number of database queries and the size of web pages. To ensure a fast and responsive web application, developers need to be aware of performance issues and continuously work to improve their web application's speed and responsiveness.

2.1.6 UX practices

User experience (UX) is a critical component of web development, as it directly affects user engagement and satisfaction. The frameworks I have mentioned before offer developers a range of UX practices to ensure that their web applications are easy to use, intuitive, and visually appealing.

Web frameworks provide different kinds of features structures and tools to enhance user experience design. It includes the usage of responsive design, accessibility, and usability testing. Developers can use responsive design to ensure that their web applications look and function well on different devices and screen sizes. Accessibility features can make web applications more accessible to people with disabilities which was spoken about earlier, and usability testing can identify areas for improvement in user interaction and experience and to fix any issues arising. Frameworks also offer built-in support for user experience design elements such as forms, navigation menus, and input validation. Additionally, frameworks often come with pre-built templates and user interface components that developers can customize to meet their application's specific needs. To ensure a positive user experience, developers must keep in mind user experience throughout the entire development process, from design to deployment and also research about their targeted users to fulfil their requirements. By using user experience best practices and taking advantage of web framework features, developers can create web applications that are both functional and user-friendly. This will bring more traffic and profit to the organisation managing the website.

In conclusion, user experience is another extremely important part of web development that highly affects user engagement and satisfaction. To create a great responsive website that includes accessibility and usability testing, developers may use existing web frameworks to help with the implementation. Developers are highly recommended in taking advantage of these features provided within the frameworks to successfully create web applications that are easy to use, intuitive, and visually appealing.

2.2 Sound Normalisation Rules

In database management, sound and normalization rules are used to ensure data consistency and integrity. Soundness refers to the accuracy and correctness of the data stored in a database. This means that data should not contain any inconsistencies, contradictions, or errors. Soundness rules can be defined based on the data domain and business requirements, and they typically involve data validation checks, such as data type constraints, range constraints, and referential integrity constraints. On the other hand, normalization is the process of organising data in a database to reduce redundancy and improve data consistency. Normalization rules define how data should be structured and how the relationships between data entities should be represented. The most common normalization rules are first normal form (1NF), second normal form (2NF), and third normal form (3NF). Each of these rules builds upon the previous one and ensures that data is organized in a way that eliminates duplication and improves efficiency.

The first normal form should satisfy two conditions, contain only atomic values and contain no repeating words. In the second normal form, it should satisfy being in the first normal form and all non-key attributes are fully functionally dependent on the primary key. In the third normal form, it should satisfy being in the second normal form and have no transitive functional dependency.

The application of soundness and normalization rules is important for maintaining data consistency and reducing errors in a database. By following these rules, developers can ensure that data is organised and presented in a structured form which makes it easier to maintain and query. Additionally, soundness and normalization rules can improve application performance and reduce the risk of data corruption.

In conclusion, soundness and normalization rules are important concepts in database and should be used to manage the database. Soundness rules ensure the accuracy and correctness of data, while normalization rules help to organize data in a structured and efficient manner. By adhering to these rules, developers can ensure that their databases are highly consistent, reliable, and efficient.

2.3 Architectural Paradigms and Design Patterns

Every piece of architecture is built upon paradigms. In current software systems OOA, CBA and SBA are the three key concepts which lead to successful design in projects.

Object-oriented architecture (OOA) is the foundational block of any software application. Hardware and software that may manipulate data structures in memory are called objects. They have descriptors that provide information about the type and capabilities, it provides protection for computer security.

Component-based architecture (CBA) is a framework used to build software using reusable components. Each of these components is built to perform a task defined by architecture. They are created by developers and stored in the library. The communication between the components is through a software bus. These components have a few features such as swapping components with similar functionality, designed to be plugged into applications without modification needs, and minimal dependencies making them dominant architectures.

Service-based architecture (SBA) creates business applications by using services which are software components. services communicate with each other to provide business capabilities and perform complex tasks by reusing services.

As there are plenty of languages and architectures which help in the website creation process, it may be confusing to start the project therefore design patterns are used. Design patterns are set of guidelines to design a user interface. When designing a website the space is limited, not a lot of information can be put on one webpage, this can result in clutter. Solution such as dropdown menu can provide help in organising links. Grid layout is used to divide the page and equally spacing it. Lastly a responsive web page that can be also used on smaller screen.

Chapter 3: **Software Engineering**

3.1 Methodology

Software methodology provides a better structure to the software development workflow. It organises the project consisting of different phases such as design and development. Examples of a few methodologies:

Agile – this methodology minimises the risk of cost, bugs, and change of requirements when new functionality is added and is time effective. This methodology is newer than waterfall. It was specifically designed due to the high failure rate of the waterfall method. The stages are:

- Concept
- Inception
- Iteration
- Release
- Maintenance
- Retirement

Waterfall – this is the most traditional methodology used, it is a linear model including sequential phases that focuses on the goals. before moving to the next phase, the previous phase must be finished. It is impossible to go back and modify once the phase has been completed. Before starting this phase it requires a lot of structured documentation as in the first step and understanding by both developer and customer must be met, otherwise it may lead to problems in future phases which will be harder to revert. The five stages are:

- Analysis
- Design
- Implementation
- Testing
- Maintenance

Feature-driven development – this methodology is derived from agile. Its main focus is to deliver working software as frequently as possible. Although it a modern method of agile it still remains to be an old methodology. The only down side is if a feature takes longer than 2 weeks to be implemented, it must be broken down further to meet the two weeks deadline. The steps are:

- Develop a model
- Build feature list
- Plan by each feature
- Design by feature

- build by feature

During each step, the progress is tracked with its possible errors

Scrum – very similar to agile methodology. To meet the requirements developers must communicate and collaborate daily, and hold meetings once a week to discuss the progress made. To succeed in this method it is recommended for smaller team use. The stages are:

- Initiation
- Planning and estimates
- Implementation
- Review
- Release

3.2 Testing

Testing occurs to check whether the produced software meets the requirements and is error-free. Manual or automated tools are used when the software's component is executed. If there are any gaps, errors, or requirements missing the system will alert the developers. Testing ensures the high performance of the software, resulting in time and cost savings. There are over 150 types of tests and they are divided into 3 categories which are functional testing, non-functional testing and militance testing.

In the functional testing, the following tests are performed:

- Unit testing: testing individual units of code
- Integration testing: test if integrated units work together
- Smoke: checking of deployed software is sable
- UAT: tested by end users before release
- Localisation: verifying software behaviour
- Globalisation: validating if a website is delivering user experience across the globe

In the non-function testing category, the following are performed:

- Performance: tests the speed, stability, reliability and response time
- Endurance: testing if a software can resist the load it should endure for a long period of time
- Load: tests the performance of the system
- Volume: checks if the software endured a large volume of data

- Usability: when users use the software and are observed

In the militance testing category, the following are performed:

- Regression: verifies modification and updates made to the software without affecting the functionality

- Maintenance: test to identify new problems or confirm that the problem was solved

3.3 Risks and Mitigations

Underestimating time:

- Although a plan with a timeline where I am giving each task a timeframe is created before starting the project, there is a possibility of having a task which can take longer than expected to complete. As a result, the rest of the tasks can get pushed. To mitigate this, in the event where I take longer to complete a task, I will spend extra hours the following week to not fall behind.

Unmet requirements:

- Unmet requirements can result from underestimating time. In the worst-case scenario, if I fall too far behind on the tasks, I would not have a fully functioning program. Unmet requirements can also occur from a lack of knowledge in certain coding languages. To mitigate this, If I find a task hard to complete I will get the help I need to get it done. For example, get advice from the project supervisor or search for online tutorials.

Poor quality of end product:

- Poor quality product is the worst-case scenario but still possible. This could result due to misunderstanding of the assignment/requirements. If I find myself having questions or issues, I will seek advice to solve the problem.

Failed test cases:

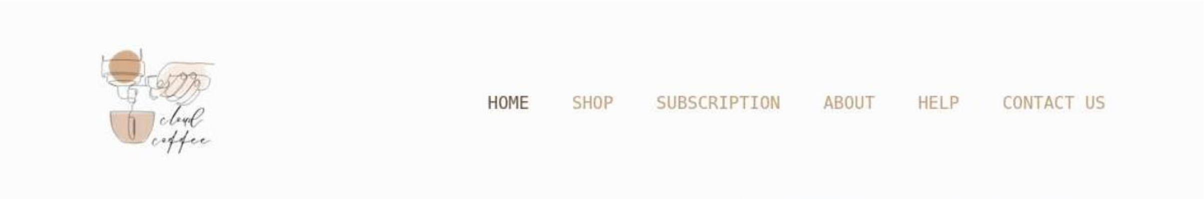
- When approaching the testing phase, some may fail. Failing can result in bad web application functionality. For example, buttons not clicking, nothing happens when trying to purchase products or the web pages are jumping and it is not possible to interact. If in the testing phase, I notice that some parts of the code do not function as should I will take action to resolve it.

Chapter 4: **End System Development**

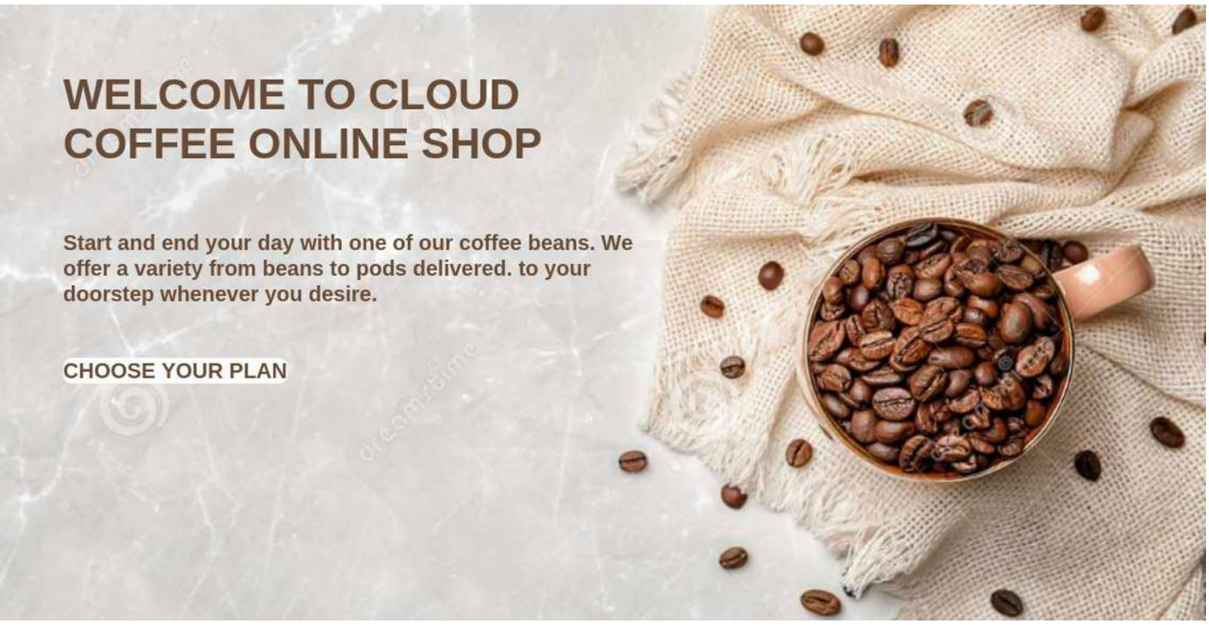
The end of system development refers to the final stage of the software development life cycle (SDLC) where the development of the software product is complete, and the system is ready for deployment. The software development cycle is a model used to manage projects consisting of seven stages to create a successful project. The phases are planning, system analysis, system design, development, implementation, integration and testing, and operations and maintenance. The advantage of this cycle is that more requirements can be added in later stages without jeopardising the project. End system development is also commonly known as the release stage. During this stage, the software product undergoes thorough testing and quality assurance processes to ensure that it meets the requirements of the stakeholders and is ready for deployment. The software may undergo various types of testing such as functional testing, performance testing, security testing, and user acceptance testing to ensure that it is reliable, stable, and meets the specified requirements. Once the testing and quality assurance processes are complete, the software product is packaged and prepared for deployment. This may involve creating installation packages, user manuals, and other supporting documents that will assist users in using the software. The end of system development also marks the beginning of the maintenance phase, where the software product is released to the users and any issues that arise are addressed through bug fixes, patches, and updates. Moreover, the maintenance phase is an ongoing process that continues throughout the software product's life cycle and involves regular monitoring and support to ensure that the software remains reliable and effective for its users. Finally, the end of system development is a vital stage in the software development life cycle. This is an exciting phase for the developers and clients as the project reaches its end and is ready to be interacted with and used by the targeted users.

4.1 The architecture of the End System

Home page:



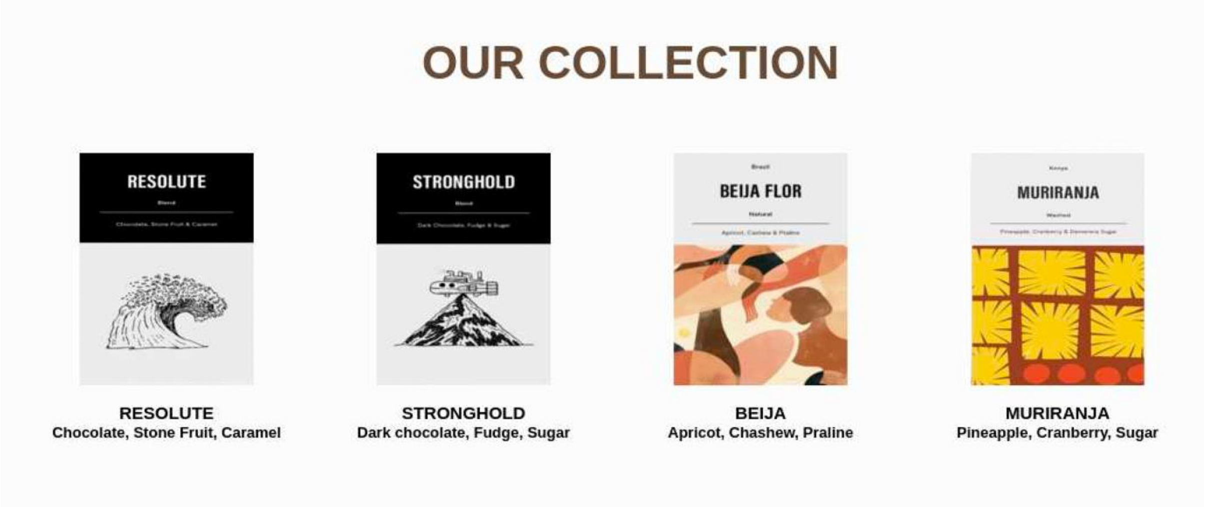
Navbar navigation menu, the background is white and the words are in a beige colour except for the home word in brown. When you move the mouse over the words they turn brown too. Each click on the page leads to a different page.



WELCOME TO CLOUD COFFEE ONLINE SHOP

Start and end your day with one of our coffee beans. We offer a variety from beans to pods delivered. to your doorstep whenever you desire.

The main screen after navigation menu, text on top of it and choose your plan menu button, when clicked leads to the subscription page.



OUR COLLECTION

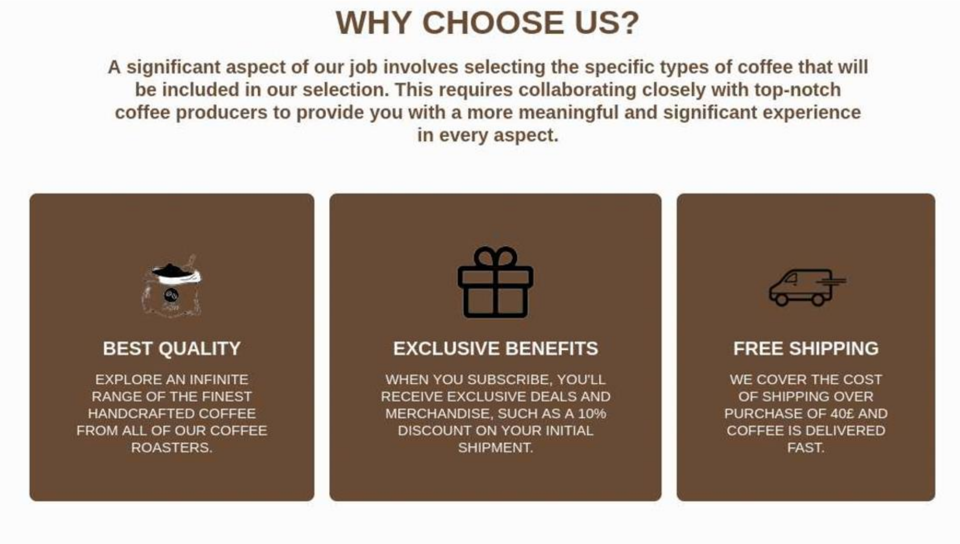
RESOLUTE
Chocolate, Stone Fruit, Caramel

STRONGHOLD
Dark chocolate, Fudge, Sugar

BEIJA
Apricot, Chashew, Praline

MURIRANJA
Pineapple, Cranberry, Sugar

The content seen after the picture, a grid of columns with images and text below, these are supposed to so to be some of the coffee sold in the online store.



WHY CHOOSE US?

A significant aspect of our job involves selecting the specific types of coffee that will be included in our selection. This requires collaborating closely with top-notch coffee producers to provide you with a more meaningful and significant experience in every aspect.

BEST QUALITY

EXPLORE AN INFINITE RANGE OF THE FINEST HANDCRAFTED COFFEE FROM ALL OF OUR COFFEE ROASTERS.

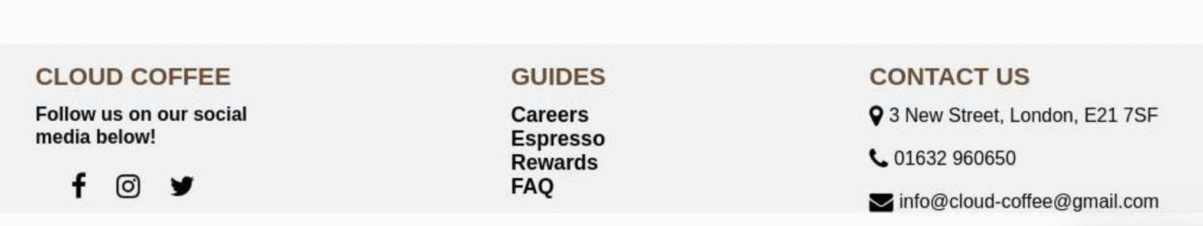
EXCLUSIVE BENEFITS

WHEN YOU SUBSCRIBE, YOU'LL RECEIVE EXCLUSIVE DEALS AND MERCHANDISE, SUCH AS A 10% DISCOUNT ON YOUR INITIAL SHIPMENT.

FREE SHIPPING

WE COVER THE COST OF SHIPPING OVER PURCHASE OF 40£ AND COFFEE IS DELIVERED FAST.

Following that, there is a grid of columns again with space between them, each containing an image and description.



CLOUD COFFEE

Follow us on our social media below!

GUIDES

Careers
Espresso
Rewards
FAQ

CONTACT US

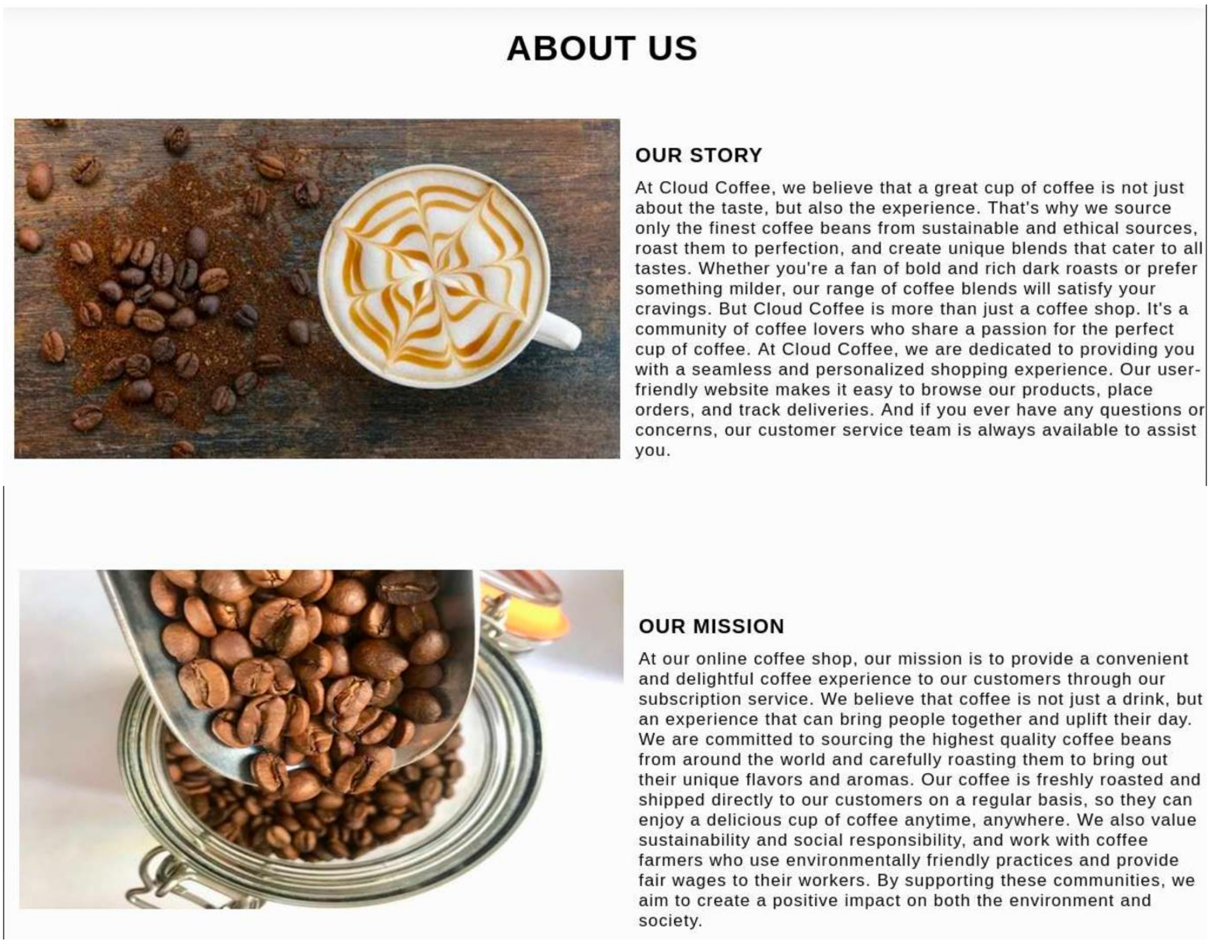
3 New Street, London, E21 7SF

☎ 01632 960650

✉ info@cloud-coffee@gmail.com

Lastly a footer part at the bottom of the page on the right there are icons with contact information, in the middle are guides but they do not lead to any other page and on the left there are social media icons and some text.

About us:



ABOUT US

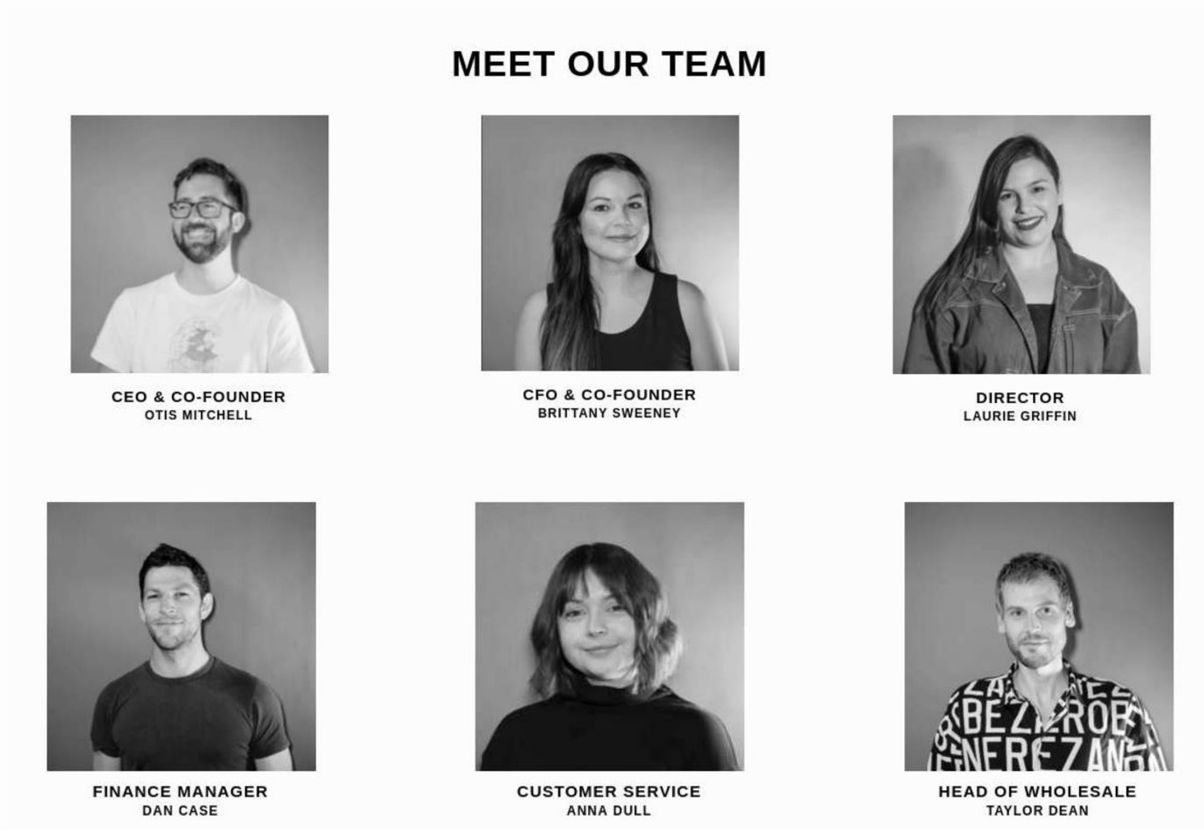
OUR STORY

At Cloud Coffee, we believe that a great cup of coffee is not just about the taste, but also the experience. That's why we source only the finest coffee beans from sustainable and ethical sources, roast them to perfection, and create unique blends that cater to all tastes. Whether you're a fan of bold and rich dark roasts or prefer something milder, our range of coffee blends will satisfy your cravings. But Cloud Coffee is more than just a coffee shop. It's a community of coffee lovers who share a passion for the perfect cup of coffee. At Cloud Coffee, we are dedicated to providing you with a seamless and personalized shopping experience. Our user-friendly website makes it easy to browse our products, place orders, and track deliveries. And if you ever have any questions or concerns, our customer service team is always available to assist you.

OUR MISSION

At our online coffee shop, our mission is to provide a convenient and delightful coffee experience to our customers through our subscription service. We believe that coffee is not just a drink, but an experience that can bring people together and uplift their day. We are committed to sourcing the highest quality coffee beans from around the world and carefully roasting them to bring out their unique flavors and aromas. Our coffee is freshly roasted and shipped directly to our customers on a regular basis, so they can enjoy a delicious cup of coffee anytime, anywhere. We also value sustainability and social responsibility, and work with coffee farmers who use environmentally friendly practices and provide fair wages to their workers. By supporting these communities, we aim to create a positive impact on both the environment and society.

After the navigation menu, this page includes 2 images one after and another on the left side of the page and text on the right side of the page.



MEET OUR TEAM

CEO & CO-FOUNDER
OTIS MITCHELL

CFO & CO-FOUNDER
BRITTANY SWEENEY

DIRECTOR
LAURIE GRIFFIN

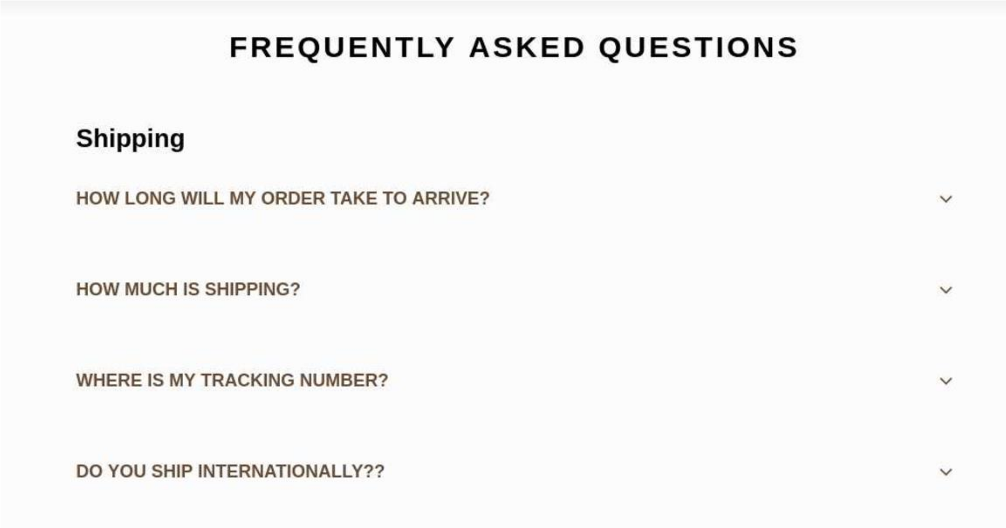
FINANCE MANAGER
DAN CASE

CUSTOMER SERVICE
ANNA DULL

HEAD OF WHOLESALE
TAYLOR DEAN

At the bottom of the page, there are images in 2 rows and 3 columns of team members, their positions, and names.

Help page:



FREQUENTLY ASKED QUESTIONS

Shipping

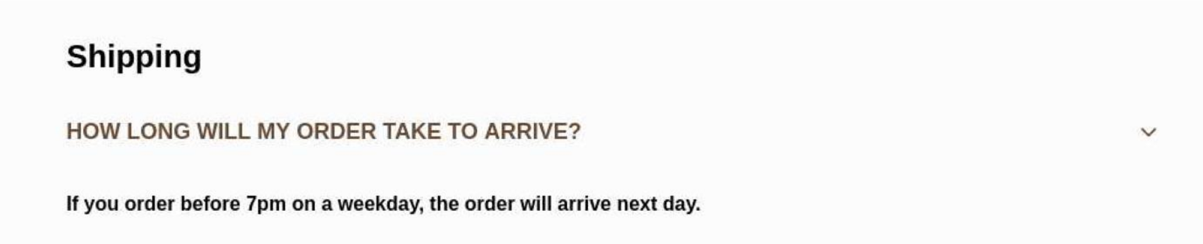
HOW LONG WILL MY ORDER TAKE TO ARRIVE?

HOW MUCH IS SHIPPING?

WHERE IS MY TRACKING NUMBER?

DO YOU SHIP INTERNATIONALLY??

This is part of the help page there are more questions at the bottom, when clicked answer is displayed see below.

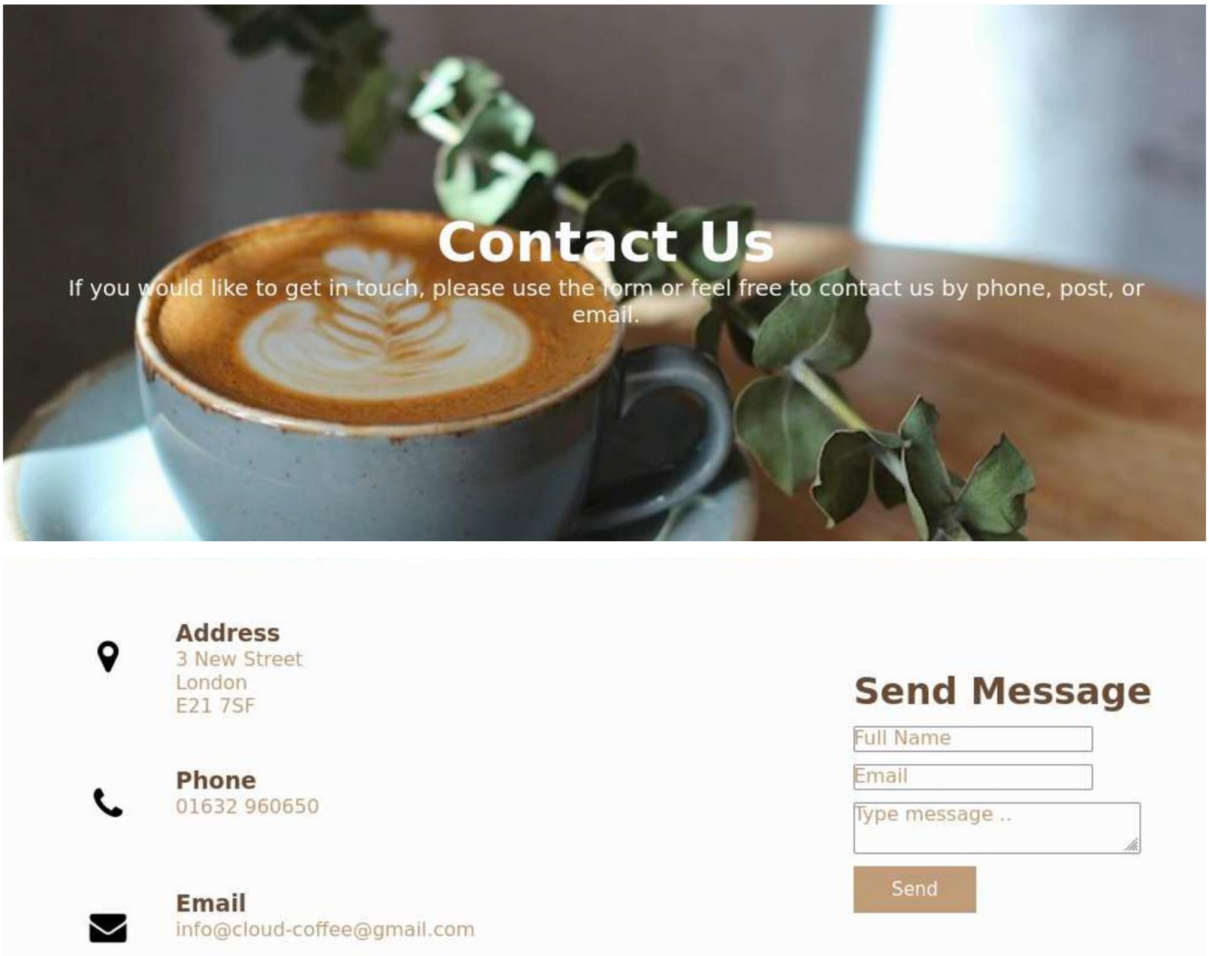


Shipping

HOW LONG WILL MY ORDER TAKE TO ARRIVE?

If you order before 7pm on a weekday, the order will arrive next day.

Contact us:



Contact Us

If you would like to get in touch, please use the form or feel free to contact us by phone, post, or email.

Address
3 New Street
London
E21 7SF

Phone
01632 960650

Email
info@cloud-coffee@gmail.com

Send Message

The page begins with an image and text on top as well and at the bottom, on the right side you can find contact information and, on the left, there contact form box that can be filled in to send an enquiry.

4.2 Features of the End System

```
<div class ="coffee-collection">
<h1>Our Collection</h1>
<div class = "row">
<div class ="column">
<img src = "resolute.png">
<h3>Resolute</h3>
<p>Chocolate, Stone Fruit, Caramel</p>
</div>
<div class ="column">
<img src = "stronghold.png" >
<h3>Stronghold</h3>
<p>Dark chocolate, Fudge, Sugar</p>
</div>
<div class ="column">
<img src = "beija.png" >
<h3>Beija</h3>
<p>Apricot, Chashew, Praline</p>
</div>
<div class ="column">
<img src = "muriranja.png" >
<h3>Muriranja</h3>
<p>Pineapple, Cranberry, Sugar</p>
</div>
</div>
```

-

```
<header class ="header">
<nav class = "navbar">
<a          href          =          "#"><img          src          =
"/home/cim/ug/wfis117/git/project/EclipseProjects/project/src/main/java/c
loud coffee1.png" class = "logo" alt =""></a>
  <div>
  <ul id = "nav">
    <li><a class = "active" href = "index.html">HOME</a></li>
    <li><a href = "shop.html">SHOP</a></li>
    <li><a href = "subscription.html">SUBSCRIPTION</a></li>
    <li><a href = "about.html">ABOUT</a></li>
    <li><a href = "help.html">HELP</a></li>
    <li><a href = "contact.html">CONTACT US</a></li>
</ul>
</div>
</nav>
```

- The code represents a header section of a webpage. It includes a navigation bar with a logo and a menu that allows the user to navigate through different pages of the website. The header element has a class called header. Within the header, there is a nav element with a class navbar. The navbar contains a logo image and a div element that wraps an unordered list with the id of nav. The list has six items: index, shop, subscription, about, help, and contact. Each contains href a link to connected page displayed when clicked.

```
<div class = "menu">
<h2>Frequently Asked Questions</h2>
<div class = "ship">
```

```
<h3>Shipping</h3>
<button class = "question">How long will my order take to arrive?
<i class="fa fa-angle-down" aria-hidden="true"></i>
</button>
<div class = "answer">
<p>If you order before 7pm on a weekday, the order will arrive next day.
</p>
</div>
<button class = "question">How much is shipping?
<i class="fa fa-angle-down" aria-hidden="true"></i>
</button>
<div class = "answer">
<p>We provide free Royal Mail tracked shipping on all UK orders with a
basket value of over 40£.<br> If it is total value is under 40£ then
shipping is a fixed fee of 5£. </p>
</div>
<button class = "question">Where is my tracking number?
<i class="fa fa-angle-down" aria-hidden="true"></i>
</button>
<div class = "answer">
<p>You can find your tracking number on the shipping confirmation
email.<br>The link provided will direct to Royal Mails Tracking website
once it is clicked.</p>
</div>
<button class = "question">Do you ship internationally??
<i class="fa fa-angle-down" aria-hidden="true"></i>
</button>
<div class = "answer">
<p>No, at the moment we only deliver anywhere within the United
Kingdom.</p>
</div>
</div>
```

- This code creates a frequently asked questions section about shipping policies for a website. The section is contained within the class menu and includes an h2 heading that says Frequently Asked Questions. The section is divided into several sub-sections, each represented by a div element with the class ship. Each sub-section includes a h3 heading that specifies the topic of the question, followed by a button with the class question that can be clicked to reveal or hide the answer to the question. The button also includes an icon with an arrow pointing down that indicates that it can show the answer to the question below it. Each question has the class answer that contains the answer to the question. The answer is represented by a p element that includes the relevant information. The questions and answers cover topics such as how long it takes for an order to arrive, how much shipping costs, where to find a tracking number, and whether or not the company ships internationally.

```
<div class = "footer">
      <!-- Social media box bottom left -->
      <div class = "box1">
      <h2>Cloud Coffee</h2>
      <p>Follow us on our social <br> media below!</p>
      <div class = "social-media-icon">
      <a href ="#"><i class ="fa fa-facebook" aria-hidden="true"
style="color:black"></i></a>
      <a href ="#"><i class ="fa fa-instagram" aria-hidden="true"
style="color:black"></i></a>
      <a href ="#"><i class ="fa fa-twitter" aria-hidden="true"
style="color:black"></i></a>
```

```
      </div>
      </div>
      <!-- Guides box after social media -->
      <div class = "box1">
      <h2>Guides</h2>
      <ul>
     <li><a href = "#">Careers</a></li>
     <li><a href = "#">Espresso</a></li>
     <li><a href = "#">Rewards</a></li>
     <li><a href = "#">FAQ</a></li>
     </ul>
      </div>
      <!-- Contact information box -->
      <div class = "box1">
      <h2>Contact Us</h2>
      <div class = "contact">
      <span><i class = "fa fa-map-marker"></i>3 New Street, London, E21
7SF</span>
      <br>
      <span><i class = "fa fa-phone"></i>01632 960650</span>
      <br>
      <span><i      class      =      "fa      fa-envelope"></i>info@cloud-
coffee@gmail.com</span>
      </div>
      </div>
      </div>
```

- This code defines footer section of a website. Made out of three sections each represented by class box1. This code defines a footer section for a website. The footer is composed of three sections, each represented by a div element with the class box1. The first section is a social media box that includes the Cloud Coffee name, a short message, and icons that link to the company's social media pages on Facebook, Instagram, and Twitter. The second section is a guide box that lists four clickable links to pages on the website related to careers, espresso, rewards, and FAQs. The third section is a contact information box that includes the company's address, phone number, and email address.

```
<script>
var acc = document.getElementsByClassName("question");
var i;

for (i = 0; i < acc.length; i++) {
  acc[i].addEventListener("click", function() {
    this.classList.toggle("active");
    var answer = this.nextElementSibling;
    if (answer.style.display === "block") {
      answer.style.display = "none";
    } else {
      answer.style.display = "block";
    }
  });
}
</script>
```

- This code defines javascript function that adds a click event listener to the HTML element with the class question. When one of these elements is clicked the function toggles the active class on the clicked element. Then it checks the next sibling of the clicked element to see if it is visible or not. If it is visible then the function hides the element by setting its display property to none, however, if it is not visible the function shows the element by setting its display property to block.

4.3 Potential Future Enhancement

- Make a proper subscription plan page where users are able to choose within 3 different subscriptions. When the desired subscription is chosen, the user is then able to choose what type of grind is needed, every how long to deliver and the size of the coffee.

- Make a better about us page where there are more images and text displayed, decrease and size, have more gap between the image and text.

- In the footer part, make the guide section and social media icons accessible, when clicked it leads to different page with information.

- Make a fully functioning shop page where products are displayed, users are able to add items to the cart, click on the product that will lead to another page with in depth description of the product.

- Make a login page where customers can create an account, login to the account view their recent order and have details saved.

- Search button, where users can search what they are looking for and it will lead them to it.

- Expand frequently asked questions.

- Make is fully accessible when using different devices.

Chapter 5: **Assessment**

5.1 Professional issues

During the project, I encountered a couple of professional issues. Firstly, time management, looking back I wish I have spent more time prioritising this project than other assignments. I have left little time to complete it which caused stress and reduced the quality of my work, However, I still tried my best ability make it the best version I could possibly make. Secondly, I had many issues with coding there were certain things that I could not understand how to achieve and would easily get frustrated over them. there were times when I would have a vision in my head of how I would like the page to look but then when using CSS and JavaScript to implement it, some stuff would not work. Also, I had issues in implementing the subscription plan page unfortunately it didn’t go as planned and is not functioning at all. Lastly, due to running out of time, I was not able to meet some of the requirements I set at the beginning of the project. Although this project is soon to be over, I would still pursue trying to finish it and make it the way I envisioned it to be.

5.2 Self Evaluation

During this project, I have learnt numerously about different topics like the kind of testing, design patterns, different types of web frameworks available and in-depth information about them, and state of art of web development. Skills that I can apply in the upcoming future. I have also expanded my knowledge about the creation and development of web applications. I learnt to use new functions in HTML and CSS that I did not know before and am very proud of the progress I made although coding wise it is not sufficient. This has been a long and exhausting journey but I am extremely happy with the process and the result.

Chapter 6: **Planning and timescale**

6.1 Timeline

My initial plan from the project plan was this:

Term 1:

Week 1: Research and make a choice for a website.

Week 2: Write the project plan.

Week 3: Sketch UI design of my website

Draft a UML diagram.

Week 4: Start code implementation.

Template pages UI

Typography & colour scheme

Logo & sketch my own coffee products for artwork.

Week 5: Displaying content.

Choosing a development framework

Designing the database structure

Week 6: Prepare for interim report and presentation.

Week 7-8: Continue working on the UI and DB structure.

Week 9: Markups for social media

Week 10-11: Authentication system & payment performance

Term 2:

Week 1 – 3: Using SE tools to improve code quality.

Week 4: Go over the website and initial design sketches and adjust.

Week 5: Apply a debugger.

Week 6-7: Start final website testing for features.

Week 8: Revisit UML diagram

Week 9: Create a user manual.

Week 10 -11: Prepare to submit a fully functioning website and final report.

However, this plan did not go accordingly. I do not have a new plan to provide as I did not keep track of my progress.

Bibliography

[1] A dictionary of computing. . encyclopedia.com. 29 Nov. 2022 . (2022) Encyclopedia.com. Encyclopedia.com. Available at: https://www.encyclopedia.com/computing/dictionaries-thesauruses-pictures-and-press-releases/object-oriented-architecture (Accessed: December 7, 2022).

[2] Admin (2023) Why is website security important?: Secure website 2023, cWatch Blog. Available at: https://cwatch.comodo.com/blog/website-security/why-is-website-security-important/ (Accessed: March 28, 2023).

[3] Appelbaum, B. (2022) Top 6 software development methodologies, Planview Blog. Available at: https://blog.planview.com/top-6-software-development-methodologies/ (Accessed: December 8, 2022).

[4] Aspittel (no date) What is a web framework, and why should I use one?, · We Learn Code. Available at:https://welearncode.com/what-are-frontend-frameworks/#:~:text=Web%20frameworks%20help%20us%20achieve,features%20rather%20than%20configuration%20details. (Accessed: March 28, 2023).

[5] Architecture paradigms and their influences and impacts on component ... (no date). Available at: https://www.researchgate.net/publication/221178754_Architecture_Paradigms_and_Their_Influences_and_Impacts_on_Component-Based_Software_Systems (Accessed: December 7, 2022).

[6] A., V. (2023) Why is react a library and next.js a framework? and which is better?, Medium. Bits and Pieces. Available at: https://blog.bitsrc.io/why-is-react-a-library-and-next-js-a-framework-and-which-is-better-cee342bdfe8c (Accessed: March 28, 2023).

[7] Beedie, G. (2020) What's involved in a website discovery and definition phase? - 93digital WordPress blog, 93digital. Available at: https://93digital.co.uk/marketing/whats-involved-in-a-website-discovery-and-definition-phase/ (Accessed: March 29, 2023).

[8] Best Laravel Security features for your application (2022) The Official Cloudways Blog. Available at: https://www.cloudways.com/blog/laravel-security/ (Accessed: March 30, 2023).

[9] Bigby, G. (no date) Website development planning process, Create Sitemaps - Sitemap Generator - Visual Sitemap Generator. Available at: https://dynomapper.com/blog/487-website-development-planning-process (Accessed: March 28, 2023).

[10] Bird, D. (2023) What is assistive technology?, Scope for business. Available at: https://business.scope.org.uk/article/assistive-technology-devices-definitions-how-disabled-people-use-the-web (Accessed: March 28, 2023).

[11] Brophy, P. and Craven, J. (2007) Web accessibility, Library Trends. Johns Hopkins University Press. Available at: https://muse.jhu.edu/pub/1/article/216646 (Accessed: March 28, 2023).

[12] Chawre, H. (2022) Top web development challenges & solutions to overcome them, Top Web Development Challenges & Solutions to Overcome Them. Turing Enterprises Inc. Available at: https://www.turing.com/resources/top-web-development-challenges (Accessed: March 28, 2023).

[13] Component-based architecture (no date) Tutorials Point. Available at: https://www.tutorialspoint.com/software_architecture_design/component_based_architecture.htm (Accessed: December 7, 2022).

[14] Craven, J. (2008) “chapter 4 the importance of web accessibility ,” in Web accessibility practical advice for the Library and Information Professional. London: Facet Publishing, pp. 41–46.

[15] Dev, B.R.M.W. (no date) Testing: Phase six in our web design process, BRM Web Dev.Availableat:https://www.brmwebdev.com/webdesign/testing#:~:text=After%20the%20development%20phase%20we,are%20no%20errors%20or%20bugs. (Accessed: March 28, 2023).

[16] Detimo (2019) Python flask: Pros and cons, DEV Community. Available at: https://dev.to/detimo/python-flask-pros-and-cons-1mlo (Accessed: March 30, 2023).

[17] Django Framework (no date) Tutorials Point. Available at: https://www.tutorialspoint.com/python_web_development_libraries/python_web_development_libraries_django_framework.htm (Accessed: March 28, 2023).

[18] Django (no date) Django Project. Available at: https://docs.djangoproject.com/en/4.1/topics/security/#:~:text=Django%20contains%20clickjacking%20protection%20in,the%20exact%20header%20value%20sent. (Accessed: March 30, 2023).

[19] Express/node introduction - learn web development: MDN (no date) Learn web development | MDN. Available at: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction (Accessed: March 28, 2023).

[20] Ferrer, J. (2022) The power of User Experience, Medium. Bootcamp. Available at: https://bootcamp.uxdesign.cc/the-power-of-user-experience-business-user-interface-customer-b21cFrederic Gonzalo (2020) Understanding the difference between mobile-first, adaptive and responsive design, Frederic Gonzalo. Available at: https://fredericgonzalo.com/en/understanding-the-difference-between-mobile-first-adaptive-and-responsive-design/ (Accessed: March 29, 2023). 2d4bf54d (Accessed: March 28, 2023).

[21] Ferati, M. and Vogel, B. (2020) “Accessibility in web development courses: A case study,” Informatics, 7(1), p. 8. Available at: https://doi.org/10.3390/informatics7010008.

[22] First Normal Form (1NF) (no date) First Normal Form (1NF) - Database Normalization. Available at: https://www.1keydata.com/database-normalization/first-normal-form-1nf.php (Accessed: March 30, 2023).

[23] Gillin, P. (2022) What is component-based architecture?, Mendix. Available at: https://www.mendix.com/blog/what-is-component-based-architecture/ (Accessed: December 7, 2022).

[24] Introduction to web development (2015) Code Conquest. Available at: https://www.codeconquest.com/what-is-coding/web-programming/ (Accessed: October 8, 2022).

[25] Ijaz, U. (no date) How to modularize code in react, Educative. Available at: https://www.educative.io/answers/how-to-modularize-code-in-react (Accessed: March 30, 2023).

[26] Joseph (2022) What are the advantages and disadvantages of REACT JS, FastComet Blog. FastComet. Available at: https://www.fastcomet.com/blog/advantages-and-disadvantages-of-react-js (Accessed: March 30, 2023).

[27] Juviler, J. (2022) The beginner's guide to web design patterns, HubSpot Blog. HubSpot. Available at: https://blog.hubspot.com/website/design-patterns (Accessed: December 7, 2022).

[28] Kadam, S. (2020) How modular programming works in Node.js, freeCodeCamp.org. freeCodeCamp.org. Available at: https://www.freecodecamp.org/news/modular-programming-nodejs-npm-modules/#:~:text=In%20simple%20terms%2C%20a%20module,one%20of%20these%20in%20detail. (Accessed: March 30, 2023).

[29] Kelly, P. (2020) A modular django, Medium. Medium. Available at: https://medium.com/@w.patrick.kelly/a-modular-django-aba5a74117b2 (Accessed: March 30, 2023).

[30] Kohler, N. (2017) 6 technical aspects to set up before selling online, WooCommerce. Available at: https://woocommerce.com/posts/technical-aspects-sell-online/ (Accessed: March 28, 2023).

[31] Laravel-modules V5 v1 v2 v3 v4 v5 v6 (no date) Introduction | Laravel Modules Docs. Available at: https://nwidart.com/laravel-modules/v5/introduction (Accessed: March 30, 2023).

[32] Managing concurrency in Django using select_for_update (no date) RSS. Available at: https://www.sankalpjonna.com/learn-django/managing-concurrency-in-django-using-select-for-update#:~:text=When%20you%20run%20multiple%20workers%20of%20your%20Django%20application%2C%20you,until%20the%20transaction%20is%20completed. (Accessed: March 30, 2023).

[33] Miller, S. (2021) What is ruby on rails?, Codecademy Blog. Available at: https://www.codecademy.com/resources/blog/what-is-ruby-on-rails/ (Accessed: March 28, 2023).

[34] Musgrave, D. (2022) Lambda, Amazon. Europa Editions. Available at: https://aws.amazon.com/lambda/serverless-architectures-learn-more/#:~:text=What%20is%20a%20serverless%20architecture,management%20is%20done%20by%20AWS. (Accessed: March 28, 2023).

[35] Nicholas Hunt-Walker April 2 et al. (no date) An introduction to the flask python web app framework, Opensource.com. Available at: https://opensource.com/article/18/4/flask (Accessed: March 28, 2023).

[36] Ogundele, S. (2021) Concurrency in web development, Hashnode. Hashnode. Available at: https://hashnode.com/post/concurrency-in-web-development-ckkdrgymr02wn96s18ddhhlzf (Accessed: March 30, 2023).

[37] Pixel506 (2023) PIXEL506 - A design thinking and Digital Branding Co.., Nearshore Digital Solution Company. Available at: https://pixel506.com/insights/how-much-traffic-can-nodejs-handle#:~:text=threading%20concerns%2C%20Node.-,js%20uses%20a%20single%20thread%20with%20an%20event%2Dloop.,module%2C%2070K%20requests%20per%20second. (Accessed: March 31, 2023).

[38] pavic, bojan and anstey, chris (2020) Why does speed matter?, web.dev. Available at: https://web.dev/why-speed-matters/ (Accessed: March 30, 2023).

[39] Pekkola, S., Kaarilahti, N. and Pohjola, P. (2006) “Towards formalised end-user participation in Information Systems Development process,” Proceedings of the ninth conference on Participatory design: Expanding boundaries in design - Volume 1, pp. 21–30. Available at: https://doi.org/10.1145/1147261.1147265.

[40] Raj, P., Raman, A. and Subramanian, H. (no date) Architectural patterns, O'Reilly Online Learning. Packt Publishing. Available at: https://www.oreilly.com/library/view/architectural-patterns/9781787287495/1ade4017-ec96-4827-a189-4191de57022a.xhtml (Accessed: December 7, 2022).

[41] Queirós, A., Faria, D. and Almeida, F. (2017) Strengths and limitations of qualitative and quantitative research methods, Zenodo. Available at: https://zenodo.org/record/887089 (Accessed: March 39, 2023).

[42] Rak, V. (2023) Pros and cons of ruby on rails, Sloboda Studio. Available at: https://sloboda-studio.com/blog/pros-and-cons-of-ruby-on-rails/ (Accessed: March 30, 2023).

[43] Ratnasingam, P. (2002) The Importance of Technology Trust in web services security, Information Management & Computer Security. MCB UP Ltd. Available at: https://www.emerald.com/insight/content/doi/10.1108/09685220210447514/full/html (Accessed: March 28, 2023).

[44] Ruby on Rails Secure Development Guidelines (no date) Veracode. Available at: https://www.veracode.com/security/ruby-security#:~:text=Since%20Ruby%20on%20Rails%20is,XSS)%2C%20and%20authentication%20vulnerabilities. (Accessed: March 30, 2023).

[45] Staff, G. (2022) The Pros and cons of Laravel - guru blog, - Guru Blog. Available at: https://www.guru.com/blog/the-pros-and-cons-of-laravel/ (Accessed: March 30, 2023).

[46] S., A. (2022) What is HTML? hypertext markup language basics explained, Hostinger Tutorials. Available at: https://www.hostinger.co.uk/tutorials/what-is-html (Accessed: December 7, 2022).

[47] Second normal form (2NF) (no date) Second Normal Form (2NF) - Database Normalization. Available at: https://www.1keydata.com/database-normalization/second-normal-form-2nf.php (Accessed: March 30, 2023).

[48] Software engineering 'the establishment and use of sound engineering principles (methods) in order to obtain economically software that is reliable and. - ppt download (no date) SlidePlayer. Available at: https://slideplayer.com/slide/7053656/ (Accessed: March 30, 2023).

[49] Team, D.F. (2021) Django Advantages and disadvantages - why you should choose django?, DataFlair. Available at: https://data-flair.training/blogs/django-advantages-and-disadvantages/ (Accessed: March 30, 2023).

[50] Team, E. (2023) 5 main node.js advantages and disadvantages: EPAM anywhere business, 5 Main Node.js Advantages and Disadvantages | EPAM Anywhere Business. EPAM Anywhere. Available at: https://anywhere.epam.com/business/node-js-pros-and-cons (Accessed: March 30, 2023).

[51] The Laravel PHP Framework – Web App Construction For Everyone (2023) Kinsta®. Available at: https://kinsta.com/knowledgebase/what-is-laravel/ (Accessed: March 28, 2023).

[52] The state-of-the-art tools for website construction - works blog (no date) Works. Available at: https://www.works.so/blog/developers/the-state-of-the-art-tools-for-website-construction (Accessed: March 28, 2023).

[53] Third normal form (3NF) (no date) Third Normal Form (3NF) - Database Normalization. Available at: https://www.1keydata.com/database-normalization/third-normal-form-3nf.php (Accessed: March 30, 2023).

[54] Top cloud providers: AWS, Microsoft Azure, and Google Cloud, hybrid, SAAS players (no date) ZDNET. Available at: https://www.zdnet.com/article/the-top-cloud-providers-of-2021-aws-microsoft-azure-google-cloud-hybrid-saas/ (Accessed: March 28, 2023).

[55] Top 4 software development methodologies (2021) Application Security Blog. Available at: https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/ (Accessed: December 8, 2022).

[56] Top 10 node.js security best practices - risks & prevention (no date) Snyk. Available at: https://snyk.io/learn/nodejs-security-best-practice/ (Accessed: March 30, 2023).

[57] The web and web standards - learn web development: MDN (no date) Learn web development | MDN. Available at: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/The_web_and_web_standards (Accessed: December 7, 2022).

[58] Valencia, A. (2022) Top 7 SEO benefits of Responsive Web Design, Search Engine Journal. Available at: https://www.searchenginejournal.com/seo-responsive-web-design-benefits/211264/ (Accessed: March 28, 2023).

[59] W3C Web Accessibility Initiative (WAI) (no date) Financial factors in developing a web accessibility business case for Your Organization ◦ Web Accessibility initiative ◦ w3c, W3C Web Accessibility Initiative (WAI). Available at: https://www.w3.org/WAI/business-case/archive/fin (Accessed: March 28, 2023).

[60] Web development - cybercraft - offshore software development company (2022) CyberCraft. Available at: https://cybercraftinc.com/expertises/web-development/ (Accessed: October 8, 2022).

[61] Website security - learn web development: MDN (no date) Learn web development | MDN. Available at: https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Website_security (Accessed: March 30, 2023).

[62] What is CSS? - learn web development: MDN (no date) Learn web development | MDN. Available at: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS (Accessed: December 7, 2022).

[63] What is javascript? - learn web development: MDN (no date) Learn web development | MDN. Available at: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript (Accessed: December 7, 2022).

[64] What Is Service-Oriented Architecture? (no date) Amazon. The University. Available at: https://aws.amazon.com/what-is/service-oriented-

architecture/#:~:text=Service%2Doriented%20architecture%20(SOA),other%20across%20platforms%20and%20languages. (Accessed: December 7, 2022).

[65] What is web performance? - learn web development: MDN (no date) Learn web development | MDN. Available at: https://developer.mozilla.org/en-US/docs/Learn/Performance/What_is_web_performance (Accessed: March 30, 2023).

[66] *What is Amazon Cloudfront? - amazon Cloudfront* (no date). Available at: https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html (Accessed: March 30, 2023).

[67] (2023) *Devices used to access the internet 2022, Statista*. Available at: https://www.statista.com/statistics/1289755/internet-access-by-device-worldwide/#:~:text=As%20of%20the%20third%20quarter,nearly%2066%20percent%20of%20respondents. (Accessed: March 29, 2023).

Appendix: Diary

- Created the following pages: index.html, subscription.html. about us.html, contact us.html, help.html
- Created the following pages: index.css, subs.css, help.css, contact.css, styles.css, about.css
- Added images for the website including logo
- Added navbar menu on every page
- Designed the navbar menu using styles.css
- Coded contact us page, includes an image, text on top of it, a row with 2 columns where right side has social media icons and contact information generated by an online website with fake contact details
- Used contact css to design this page and make visually pleasing
- Coded help page which is the frequently asked questions that are divided into 4 different sections each has a heading and then a drop-down menu. On each menu, there is a question and when clicked the answer drops down, some small javascript was added in the html file
- Used help css to design this page and make it visually pleasing.
- Coded about page that includes 2 pictures on the right side and text on the left side that tells the story of the online store, below are pictures of the team that are working in the shop.
- Used about css to design this page and make visually pleasing
- Coded subscription html plan however it is not functioning it currently includes 3 large boxes explaining the plans and have a button
- Used subs css to design t his page and make visually pleasing
- Coded home page, the main page of the website. Large picture displayed, text on the image then there are small pictures of selection of coffee available and at the bottom there are boxes columns with information.
- Used index css to style to home page
- Added a footer to some of the pages
- Added some content into the shop page