FINAL DOCUMENT
CIPHERCRAFT

By Josh Casey C00261828

(Student) - Josh Casey
[Email address]

Acknowledgements

I would first like to express my thanks to my project supervisor, Dr Keara Barrett who guided me throughout the entire project. Helping me improve my documentation from the structure and layout, to giving feedback for presentation slides and research posters, her insights were very beneficial in achieving what I did during the project. The guidance provided during the development stage was especially helpful, without them I never would have got as much done as I did.

Next, I would like to express my thanks, to all my lecturers, and classmates who provided ideas, and solution in overcoming challenges, and feedback during the presentations. I would like to thank my friends and classmates for supporting me throughout this project, from giving feedback, and ideas on how to better CipherCraft, or just providing moral support, thank you.

Table of Contents

1.1 Project Overview

CipherCraft is a web-based application with a design aimed to provide an engaging learning experience focusing on the fundamentals of cryptography. It wishes to create a free online learning tool for those interested in cryptography, particularly targeting third-level education students in computing fields like Cyber Security, Software Development, and IT Management.

The platform offers different methods of learning, includes interactive components and progress tracking through quizzes, to ensure users grasp the basics before advancing to the more complex topics. Inspired by CrypTools' visual and interactive approach, CipherCraft aims to cater to various learning styles.

With a structured curriculum, CipherCraft guides users through the foundational concepts before introducing advanced topics, preventing overload and frustration. By providing the theoretical knowledge required, the platform equips students with the ability to further their study in the practical world of cryptography.

CipherCraft is built using a stack of technologies including Python, Flask, MySQL, Docker, HTML, JavaScript, CSS, JSON, and Jinja. These technologies enable the platform to deliver interactive content, handle data storage, and provide a dynamic user experience.

Overall, CipherCraft serves as a starting point for learning cryptography, empowering users to explore their interest in cyber security and gain essential knowledge and skill for their academic and professional pursuits.

1.2 Introduction

CipherCraft, a platform dedicated to teaching cryptography fundamentals. This document outlines the development journey of CipherCraft. Focusing on the different achievements such as the learning style integration, the course structure, technical to personal achievements, project review insights, and how CipherCraft is relevant to cyber security.

Exploring the different technical challenges faced throughout the development process of CipherCraft, such as creating a dynamic website, including dynamic functions, and the solutions implemented to overcome these challenges.

Discover how CipherCraft accommodates various learning styles, visual, auditory, kinaesthetic, and the implementation process of each, such as images, videos, text-to-speech, and interactive functions.

Gain some insight into the curriculum structure CipherCraft follows, and its assessment strategies, with potential enhancements for future development. Reflect on the project review which includes refining the research approach taken previous, thoughts on the modules used, and the missing features.

Understand how CipherCraft is relevant to cyber security with emphasis on how cryptography relates to cyber security, and how CipherCraft is built to provide fundamental knowledge of cryptography.

1.3 Technologies

The technological framework stack used to develop CipherCraft, include the use of Python version 3.11.3, with it I used Flask framework, and Jinja came along with the installation of Flask as a dependency. The development of CipherCraft also needed a database, as I am most comfortable with MySQL, it was the no brainer choice, using docker, I created a container to house the MySQL database, and the MySQL Workbench to view and alter the information within the database. HTML and CSS where used in conjunction to provide both a layout and design of the platform. JavaScript was used for frontend functionalities, as well as the development of the interactive functions, while a JSON file was used to contain the course content.

Python:

Throughout the progress of the development of CipherCraft, all backend functionality was created using Python. This allowed for me to split my code up into classes and import the required functions when needed. The libraries and packages I used include, the cryptography library, for developing my encryption functions, os, random, json, re, and a few others that are pre-installed with python to achieve certain functionality that was required for the development of CipherCraft.

Flask:

Flask is a lightweight, flexible framework for Python, used for web development. It provides the basic tools and features needed without imposing strict rules, or dependencies. (Barguzar, 2024) The primary reason for using Flask was its simplicity, and how allowed for a minimalist approach when developing CipherCraft. The use of Flask for the development of CipherCraft was to create routes, that handle different sections of the application, and to determine where users would end up, based on there interactions within the application. Flask was also used for session implementation and management.

Jinja:

Jinja is a templating engine, it is very commonly used throughout web development, especially with Flask. (Flask, 2010) The main use of Jinja throughout CipherCrafts development, was the use of its variable substitution, and control structures such as if statements and loops. Jinja also provided the ability of preventing cross-site scripting, as it automatically escapes content by default.

MySQL:

MySQL is a database management system, that is widely used for storing, and structuring data. CipherCraft used a MySQL database to store the user information, such username, and passwords. It was also used to store user progress of the content, the quiz questions, and there results. The use MySQL Workbench was used to setup the tables, and manage the data.

Docker:

The use a docker container to house the MySQL database, was used to allow for consistency of database, while working from a laptop and desktop machine. As the docker container, contains all of the configurations and dependencies, it allowed for me to work off two different machines with the same information, simply updating the container when needed.

HTML & CSS:

HTML was used to create the skeleton of the website, providing different things like forms, for login/signup, buttons for navigation, and placing actually word on the webpages. While CSS was used to design and layout of these HTML elements, adding colour, text style, etc. The use of both HTML and CSS allowed for the ability to provide a user interface, that looks clean and easy to navigate.

JavaScript:

The role of JavaScript in the development of CipherCraft was to add frontend functionality, such as providing colour selecting within the quiz, so when a user selects an answer the selected option is coloured. JavaScript was also used in developing the functions for the interactive components, as it was easier to access a JavaScript file with the dynamic aspect of the interactive functions.

JSON:

The use of a JSON file throughout the development of CipherCraft was to house and store the content related information, from text, images, videos, and the required input fields of an interactive function alongside the function name. The JSON file was structured in a hierarchy from Module to Topic to Pages.

2.1 Achievements

This section is to disclose the accomplishments achieved throughout the development of CipherCraft. From the many technical milestones such as dynamically displaying the content to the personal growth of skills such as problem-solving and the considerations of different learning styles. Exploring the wide array of achievements, I have attained during the creation of this online learning platform. This development process allowed for me to advance my technological level, problem solving skills, project management capabilities, and the integration of learning styles to enhance user experience.

2.2 Learning Styles

Methods of learning otherwise known as learning styles, are a form of learning and understanding. These types of learning styles can be grouped differently, such as words, pictures, speech or written, visual, and listening, either way, these methods of learning are used to determine which method best suits an individual's strongest form of understanding and or greatest level of engagement.

In this section, we explore the distinct types of learning styles and their implementation within the development of CipherCraft, an online learning platform focused on the fundamentals of cryptographic principles. Drawing from the research conducted prior and theoretical frameworks such as the meshing hypothesis, exploring the characteristics of each learning styles and the implementation process.

In developing CipherCraft, careful consideration was given to integrating various learning styles to enhance user engagement in hopes that the knowledge provided was absorbed. From a technical standpoint, this involved implementing features and functionalities that catered to different learning styles. Below I will discuss how each learning style was incorporated into the platform.

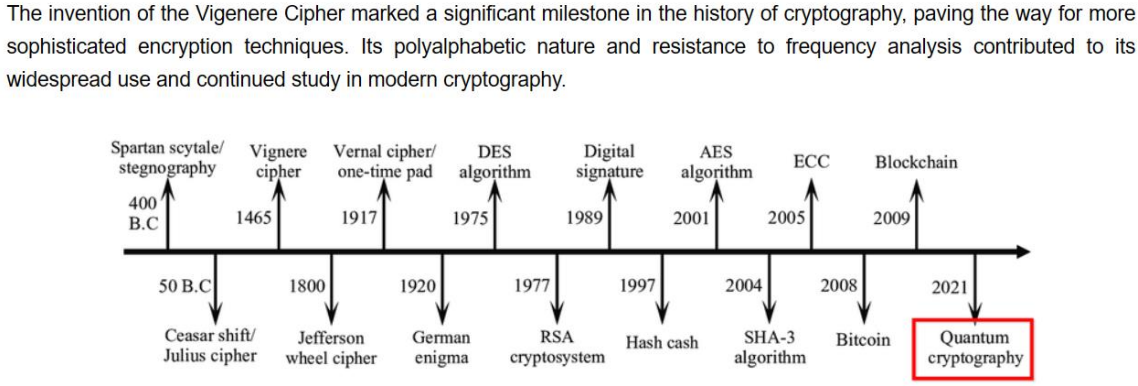
Visual Learning:

Visual Learning is a style of processing information from nonverbal manners such as images, graphs, maps, etc., it allows for the learner to view the information in a non-written form for a stronger understanding, for these types of learners the use of diagrams and colour would be best suited. Visual learners would tend to visualise objects or have a photographic memory's, allowing them to recall information from a mental image. Visual learners will also have their preference in the way data can be visualised, some may prefer graphs while others may prefer maps. Understanding the diverse types allows educators to tailor the information to the learner. (Casey, 2024)

Visual Implementation:

Ensuring that CipherCraft had the ability to allow for visual learners to truly engage with the application was not a small achievement. The implementation for this was quite laborious, from gathering the visual components that best suit the content, to successfully displaying them on the screen in a dynamic manner, lead to its own difficulties. Establishing a method that allowed for the visual components to be dynamically displayed with the correct corresponding content, was the most troublesome aspect of this implementation. Overcoming this challenge, lead to the solution of other challenges the development of CipherCraft took on. By using different tags/placeholders within my JSON file allowed for the content and visual components to be correctly displayed side by side.

The image above shows a snippet of code from the JSON file, that houses the content. The "id" in this case represents the page number, the "content" equates to the written information, and "image," the visual component of that page. Following this method allowed for the synchronisation between different types of information content. The image below displays how the content and images are displayed with in the application side by side seamlessly.



Written Learning:

Written Learning or read/write learning is a method of absorbing information through the means of reading notes, handouts, and textbooks. These types of learners tend to retain information best from reading and rereading to writing and rewriting information. To increase the success rate for a learner with this style they should include the following in their studies, using lists, headings, notes verbatim in class, and writing statements for diagrams. Written learners nowadays have the benefit of different online tools to assist with their notetaking, as they have places such as Microsoft Word, which allows for automation of creating lists and highlighting of keywords, these tools help the learners with creating their notes for later usage. (Casey, 2024)

Written Implementation:

Knowing that CipherCraft, requires dynamic content loading, flexibility, and scalability, the only choice was to store the content in a secondary location and fetch it when required. Which brought along the challenge of which is the best method of storing the information and how exactly it should be call upon the issue.

When making the decision on how to store the content, figuring out the best way for not only the written learning but how each method would affect all other learning styles was taken into consideration, such as using a database did not seem applicable due to the need of interactive functions, images, videos, etc. Storing all of these within a database would lead to complications down the road, such as the numerous amounts of querying back and forth with the database making sure the information is correct before displaying it.

Wanting CipherCraft to be quick and responsive, left only one true option, which was to use a JSON file, and structure the content accordingly so it becomes almost intuitive when displaying the content. Structing the content based on a Module, Topic, Page hierarchy, allowed for full control on what information would be displayed on each page. This method also allowed for the content to become dynamic, scalable, and flexible.

Listening Learning:

Listening Learning, otherwise known as auditory learning, is the method of learning where individuals best retain information through verbal communication. People who prefer this method of learning have a strong memory for remembering what they have heard. The components of speech are particularly important for these learners, i.e., the tone, pitch, and loudness are all key aspects in benefiting these learners in retaining and understanding the information, these learners may also read aloud or quietly to themselves to assist them in absorbing the information. These learners may struggle with written instructions but may gain a stronger understanding if verbally explained. Auditory learning is an asset as the ability to understand from listening can be used in an academic, personal, and professional manner, benefiting the learners through life. (Casey, 2024)

Listening Implementation:

Providing an avenue for auditory learners on CipherCraft, the implementation stage brought forward two conclusions, the use of videos to provide a layer of audio to the application, and the second being use of a text-to-speech API. The inclusion of videos came about by adding a tag/placeholder within JSON file, then storing the value of the URL to the video.

The text-to-speech component of CipherCraft was implemented using Responsive Voice API, this is a free text-to-speech API, which allowed for the inclusion of audio based on CipherCraft's content. As auditory learners respond to different tones and pitches, the best way to provide the text-to-speech functionality was to incorporate different voices. CipherCraft includes two different audio buttons, Male and Female, demonstrating the different tones and pitches.

```
<div class="audio-button-container">
  <button id="audioFemaleBtn" onclick="responsiveVoice.speak(pageContent, 'US English Female')">Audio Female</butt
  <button id="audioMaleBtn" onclick="responsiveVoice.speak(pageContent, 'US English Male')">Audio Male</button>
</div>
<div id="pageContent">{{ content_data.content }}</div>
```

The code above demonstrates how the text-to-speech implementation works. Firstly, setting up buttons for Female, and Male. To using the onclick call to the responsiveVoice.speak, with the information in "pageContent", using the voice US English Female and Male. Below this the "pageContent" is the content_data.content, which is the content fetched from the JSON file.

As the responsive voice text-to-speech API is free, it is not perfect. It is sometimes slow to respond when the buttons are clicked, and highlighted text within the application is also sent to the responsive voice API. Further development of CipherCraft would lead to a change in text-to-speech API, or even the inclusion of mp4 files which contain audio of the content. To enhance the learning experience for auditory learners.

Interactive Learning:

Interactive learning, also known as kinaesthetic, tactile, and hands-on learning, is the method of absorbing information through physical aspects such as interacting with an object or participating in a lab activity. Interactive learners prefer movement and interaction with their environment to best learn and understand, i.e., if an individual wishes to learn something new like riding a bike, they have the options of verbally understanding from an explanation, watching a video online demonstrating it, or reading instructions, but kinaesthetic learners would prefer to start peddling the bike straight away. These types of learners prefer areas that use hands-on activities instead of passively listening, watching videos, or viewing graphs/diagrams. During the hands-on activities, the engagement level of these learners will be captured, allowing them to process and understand information. (Casey, 2024)

Interactive Implementation:

The implementation of interactive functions to accommodate for kinaesthetic learners, was by far the most challenging. From decision making on what type of functions to include which technology to use for these functions, and how to dynamically include these functions while allowing for scalability and flexibility within platform.

Deciding on interactive components that best suit the topic, demonstrating how the encryption process works, by allowing users to enter their own message and key values. These functions were then programmed using JavaScript, primarily to prevent any delay in passing the information back and forth between the client and server. Using a frontend programming language allowed for faster response time within the function calls, but also had brought the downside of a lack of proficiency in using JavaScript to perform these types of tasks.

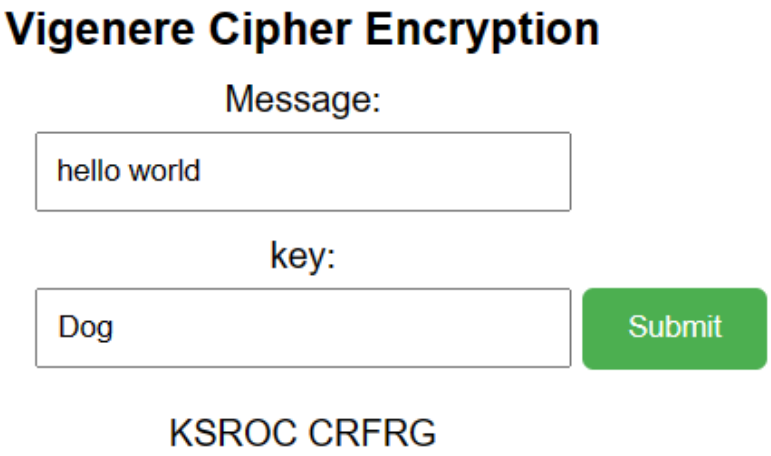
Finding a solution that includes the dynamic calling of these functions, and their respective fields was by far the most challenging. As the only solution I could realistically see working was including a div/id tag for each function and based on the function call to also call upon its div/id tag. This method would have included the process of developing a div tag for every interactive function and including JavaScript code to handle the data flow of these functions.

Following that method would have made the dynamic aspect of CipherCraft fall short, as this would include static information in idle wait of being called. Instead, the solution came clear when I was able to populate field ids within my HTML using a combination of Jinja and placeholders in my JSON file. Figuring that I could transport information from my JSON file to HTML tags/ids using Jinja created the solution.

```
{% for input_field in content_data.interactive_component.input_fields %}
<label for="{{ input_field.id }}">{{ input_field.label }}:</label>
<input type="{{ input_field.type }}" id="{{ input_field.id }}">
{% endfor %}
```

```
"interactive_component": {
    "title": "Vigenere Cipher Encryption",
    "input_fields": [
      {
        "label": "Message",
        "type": "text",
        "id": "message"
      },
      {
        "label": "key",
        "type": "text",
        "id": "key"
      }
```

By structing the different input fields within my JSON file, I was able to transport them over to the HTML page using Jinja. The two images above show's how the process operates. First assigning the input fields using a label, type, and id, which has no limit, meaning I could generate as many of them as I need. Then using Jinja templating engine, embedding of a loop for the number of input fields within the received number of input fields, populating the labels and inputs becomes easy. The image below shows the outcome of this approach.



The integration of these learning styles into CipherCraft's course design and functionality, ensure that users with diverse learning preferences can effectively engage with the platform and gain a strong understanding of cryptography. During the research of learning styles, research into assessment types was also taken into considerations. The research focused on two styles objective and subjective. In the research document the weighted consideration and insights into the assessment style best suited for CipherCraft was presented, being objective.

Objective Assessment:

This is a form of assessment where the answer is either correct or incorrect and does not contain any essays or long descriptive answers. It determines the individual's knowledge based on a quantitative approach. This assessment type is more frequently used in subjects such as Maths, Science and Computer Science, where answers are not up for interpretation and are either correct or incorrect. (Casey, 2024)

Objective Assessment Implementation:

As CipherCraft adapted to an objective assessment style, to incorporate a form of testing the user's knowledge, in an interactive way. This assessment was also used to prevent users from further exploring different modules with more complex topics.

The implementation of the objective assessment underwent a few implementation considerations, such as what type of questions should be asked, the best method of storing the questions, how to display the questions, and determining what result is necessary to show that a user is ready to progress to the next level.

When it came to what type of questions should be asked, referencing the research document, it showed a few different question styles that are suited for an objective assessment, these question styles include:

- Multiple Choice
- True or false
- Fill in the blank
- Matching
- Assertion and reasoning

Incorporating all these questions styles would have dramatically increased the overall assessment of users, as each style comes with their own benefits. Unfortunately, this implementation fell short, due to time constraints and other commitments, leaving the assessment questions that CipherCraft provide to users as exclusively multiple choice. (Casey, 2024)

This made the storage aspect of the questions straightforward, as they all required the same form of storage. The conclusion being a database, storing all the questions, using columns such a module, question, the different options, and the correct option. This allows for all the questions to be stored in a singular location but having that module column allows for a unique identifier.

Figuring out how to display the questions was not a simple query to the database returning all questions based on a module. The questions had to be picked out, randomised, categories, and stored correctly on the server to ensure that the results were accurate and true. Through trial and error, implementing a random quiz that will only display 15 questions from a pool was successfully implemented.

If the opportunity presented itself, allowing for more time, additional considerations in the quiz would have been taken. For example, considering the different learning styles, visual representations of the questions would have been included, a text-to-speech functionality on the questions would have been included adding an element of audio that would read out the question, and its accompanying selection of answers.

Additional interactive functionality such as an optional quiz which would put the users cryptographic programming skills to the test, having them create a function in a programming language of their choice that preforms a certain cryptographic function. If more time was allocated these are the functionalities considered for future development of CipherCraft's assessment strategy.

2.3 Content Strategy and Organisation

The content strategy for CipherCraft underwent a meticulous approach in gathering and structuring information, from many diverse sources, and the experience of the cryptography module, in last year's curriculum hoping to instil a strong learning experience for its users. The course curriculum is structured into five modules, each designed to build upon the previous one and provide users with a good foundation in the fundamentals of cryptography.

Module 1: The Start of Cryptograph

The first module contains a brief introduction to cryptography and its history. It contains a description of different terminology used throughout the course so that the users do not begin to feel lost in later stages. Module one also contains some historical cryptographic principes so users can see how they work and function, so they have a general idea as to how an encryption process works. (Casey, 2024)

The different historical principles that are covered include the Caeser Cipher, Vigenère Cipher, and the Fence Cipher. Demonstrating the differences in transposition, substitution, and polyalphabetic, encryption methods.

Each topic follows the general approach:

- Historical overview
- The encryption processes
- The decryption processes
- Weakness

Each encryption process includes an interactive component allowing users to engage with the platform and for a better understanding through visual methods. For example, the user enters a message, decides upon a key, hits submit and returns the encrypted message. This module is also full of visuals such as diagrams, and videos for further explanations. (Specification Document)

Module 2: Where it is now

Module two jumps into the more complex aspects of cryptography, such as the different types, symmetric and asymmetric. The topic explains what symmetric and asymmetric encryption is, using diagrams to explain how both operate, how they contrast, the advantages and disadvantages of each, and a video providing further information on these methods of encryption. (Casey, 2024)

The next topic covered in module two is Block Ciphers VS Stream Cipher. This topic details what a block and stream cipher is, how the encryption and decryption process operate, using visuals throughout to provide a visual learning aspect. Jumping to the strengths their strengths, to real world examples, and finally touching on hybrid encryption as a form of incorporating both block and stream ciphers to achieve their strengths and remove their weakness. (Casey, 2024)

The one-time pad was used as an example of the perfect encryption, while explaining the unrealistic aspects it requires in proving to be completely random. The one-time pad was primarily used to demonstrate how X-OR works and operates. Using videos and visuals in hopes of portraying the information so users gain a strong understanding of the perfect encryption. (Casey, 2024)

The Diffie-Hellman key exchange protocol was covered in depth, providing an example of an asymmetric encryption algorithm. Diffie-Hellman provided knowledge about how secure communication is possible, while also touching on the strengths and weaknesses. With the use of video explanations, and diagram to display how the algorithm/principle operates. (Casey, 2024)

Module 3: Advanced Encryption Standard (AES)

Module three provides an in-depth breakdown of how the AES encryption functions, i.e., the number of rounds, sub bytes, shift rows, mix columns, etc. This provides the user with knowledge of symmetric encryption, using the most popular encryption algorithm used today. Discussing the real-world sectors that utilise AES encryption, giving the users some knowledge of different areas that require symmetric encryption. (Casey, 2024)

This module also contains information on different modes of operations, Electronic Codebook, Cipher Block Chain, and Counter, providing an in-depth explanation, their corresponding strengths, weaknesses, vulnerabilities, and mitigations. Below is a general layout followed when creating the content for each of the modes of operations.

- How they work
- Strengths/Weakness
- Vulnerabilities
- Mitigations

By following this general layout, each mode of operations presents in a similar style in hopes for the users to follow along much easier. (Casey, 2024)

Module 4: Hashing Algorithms

The primary focus of module four was to provide the users with a break from encryption principles and explore a different cryptographic technique. The core topic of module four is hashing.

The module starts off with topic one explaining the ins and outs of what a hashing algorithm is, a one-way function. Explaining to users that when something is hashed it cannot be unhashed. While touching on sections a person may consider using a hashing algorithm. Brief overview of the history of hashing, and mentioning preimage and collision resistance, which setup the next topic. (Casey, 2024)

The second topic discusses preimage resistance, and collision resistance. Explaining what both means, the implications of them, and methods to mitigate them. Discussing how new advancements are made every single day, informing users that developers need to stay up to date on these new advancements. There was consideration of implementing different interactive functions for displaying preimage resistance and collision resistance but due to time constraints and other commitments this was not possible. (Casey, 2024)

The next section of module four provides information about the most popular hashing algorithms, MD5, SHA1, SHA256, and SHA-3. Giving the users insights into which of them are used, standardised, and which are vulnerable. Which an interactive section allowing the users to input a message and see the results hash value. (Casey, 2024)

Ending the module is a topic providing use cases for hashing algorithms, such as password storage, digital signatures, and file integrity. Leaving room at the end of the module of interactive components, such as displaying how passwords are hashed and salted, allowing a user to upload a file and download the hashed version of it. (Casey, 2024)

Module 5: Key Management

The last module puts emphasis on the importance of key usage. It explains how the key is major vulnerability to encryption. Demonstrating how properly handled, generated, and stored keys equate to the strength of the encryption. While providing the users information about the life cycle of cryptographic keys, generation, storage, distribution, rotation, and revocation. Explaining how key management relates to integrity and confidentiality for protecting data during transmission and in storage. (Casey, 2024)

Following this is a topic the delves into the necessity, of truly randomised key generation. As non-random keys may exhibit patterns or biases that could potential be exploited by threat actors, as well as become susceptible to different vulnerabilities such as brute force. Explaining how truly random keys are generated, using unpredictable sources of entropy, and no patterns or biases. (Casey, 2024)

The third topic covered in this module covers aspects of key storage, demonstrating how it is a critical component in maintaining the confidentiality and integrity of sensitive data. Covering different methods of key storage, Hardware Security Modules (HSMs), software-based key vaults, cloud-based key vaults, and others. Exploring the advantages and disadvantages of each. Exploring the option of hybrid storage methods to maintain the strengths while limiting the weaknesses. (Casey, 2024)

The last topic covered in CipherCraft, goes into explaining all the various stages a cryptographic key goes through, from its birth to retirement. Explaining how proper management of keys creates security and the effectiveness of cryptographic systems. Exploring the different avenues that keys are used in, from encryption to decryption, or authentication to digital signatures. Disclosing information such as key rotation as a best practice to mitigate any security risk. Essentially informing the users of CipherCraft about the best practices to follow, to ensure confidentiality, and integrity. (Casey, 2024)

The course content is organised and stored in a JSON file format, utilising a hierarchical structure of modules, topics, and pages. Each module contains a selection of topics, and each topic contains a selection of pages. The pages then contain the information for the content, images, video, and interactive components. This structured approach allowed for CipherCraft to provide users with an engaging and dynamic learning experience, with their preferred learning style. (Casey, 2024)

2.4 Technical

2.4.1 User Interface

The use interface for CipherCraft was developed to prioritise usability, enhance visual appeal, and cater to diverse learning styles. Drawing inspiration from the Cisco learning platform and other education resources, CipherCraft's interface is designed to create intuitive navigation and engaging interactions.

Usability:

Usability was the key principle of focus when designing CipherCraft. The layout provides a clear navigation located at the top right-hand side of each page. Buttons are used and coloured to stand out so users can clearly see them, correctly labelled so users know where that button will take them.

Visual Appeal:

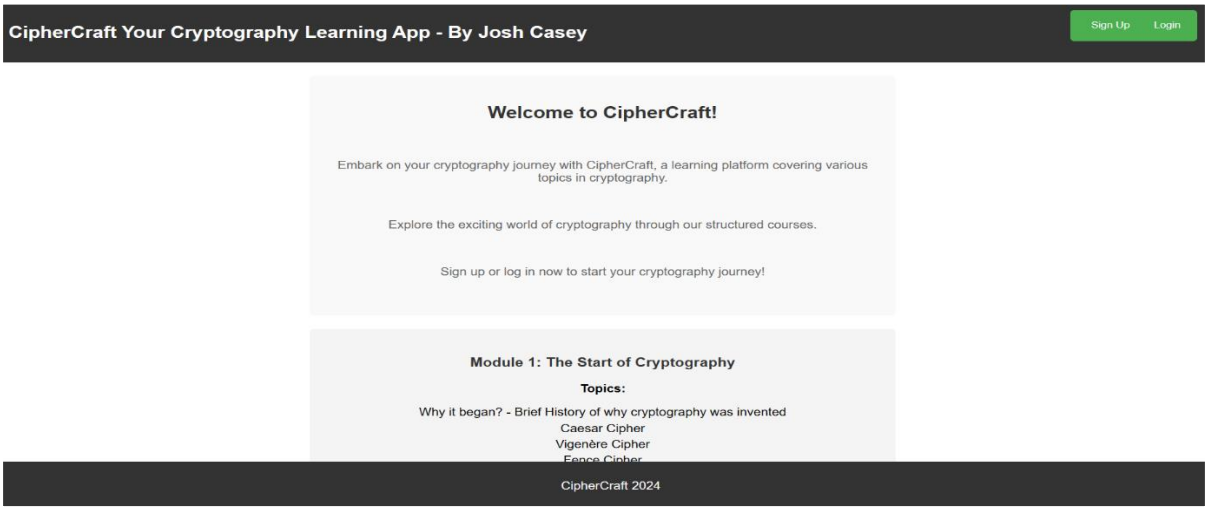
The visual design of CipherCraft was created in hopes of providing an engaging and immersive learning environment. A clear colour scheme, consisting of tones that contrast well with each other, ensuring that readability of the content is possible. Typography choices enhanced the readability of the information.

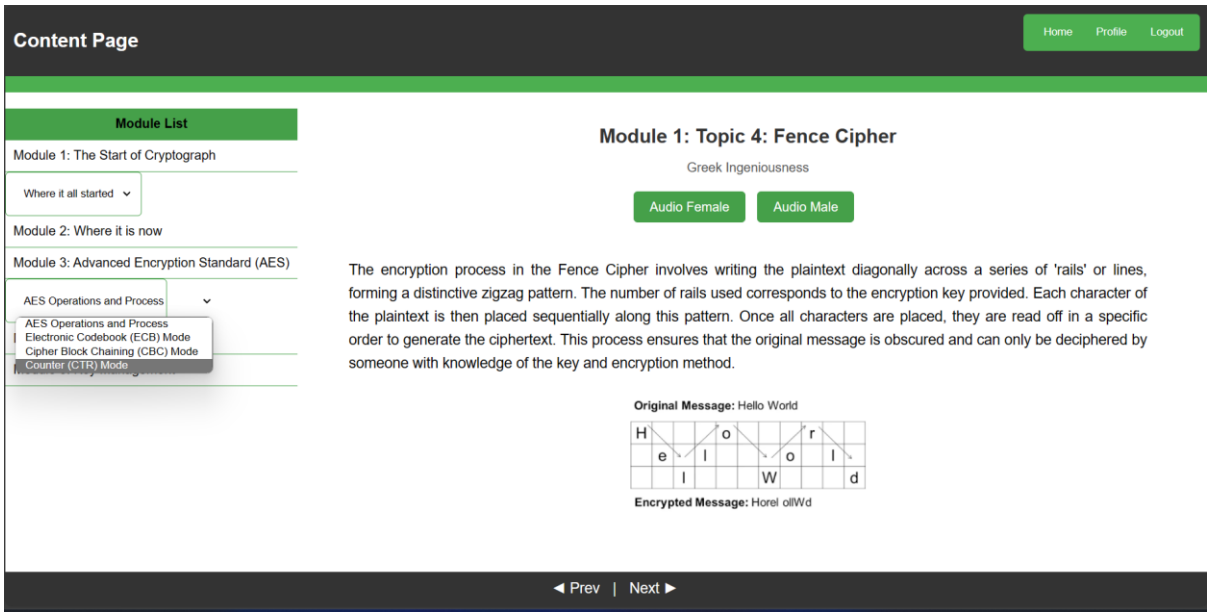
Consideration for Learning Styles:

CipherCraft's content interface caters to a variety of learning styles, providing different forms of processing information. For visual learners, images are used consistently throughout the application. The auditory learners benefit from the different audio aspects such as videos, and text-to-speech functionality. Interactive components such as the encryption process for the Caesar Cipher are included for kinaesthetic leaners.

Navigation:

The navigation within CipherCraft, is aimed to be user-friendly. The placement of navigation buttons at the top-right hand side of each page ensures consistent access to traversing the application. With the use of dropdown menus within the content page for swiftly going from one module to another. Progress bars included within the quiz also providing the user with a navigation method within the quiz, knowing how many questions they have left before completion.





The images above are of the unauthenticated home page, and the content page. As described previous both pages follow the same style, navigation at the top with the page heading at the left-hand side, with navigation buttons on the right. Following the same colouring scheme, of white, black, and green. The home page displays the information in the centre, as this is the location most people will be staring at. The content page has the information slight right of the centre due to the dropdown "Module List," where users can swiftly navigate between modules and topics. Footer containing the next and previous buttons to utilise the space.

2.4.2 Knowledge of Flask increased

The development of CipherCraft expanded my knowledge of the Flask framework, from using sessions tokens, to utilising Flask's after request functionality which will run after every request, passing information to HTML pages by including them after the render of a template. The use of sessions allowed me to determine authenticated and unauthenticated users, the users progress within the course, and if a user is taking a quiz or not. The after-request functionality allowed for me to clear the cache of the quiz, preventing users from accessing the same quiz after they have finished it. Utilising the built-in protections against cross-site scripting, such as template rending and Jinja2, assisted in securing CipherCraft.

2.4.3 CipherCraft becoming Dynamic

How to store the information and the different methods of displaying the content? These are challenges that came with developing CipherCraft. Many different solutions presented themselves, from simply statically storing the content within separate HTML pages calling to them based on the learner's navigation path, which would have led to the creation of hundreds of HTML pages which in return seemed tedious, and not to mention removes any level of scalability, or flexibility within the application.

While another solution to overcome this put me on the path of developing a Single-Page Application (SPA). A SPA requires only one HTML file, which in return stores all the information, a SPA determines which information to display based on its implementation. Access to different sections can be determined through server routes, session IDs, or any other form. The downfall to using a SPA is how it renders the HTML page every single time. Meaning any time the server refreshes, it requires the massive file to be re-rendered. (Adobe Experience Cloud Team, 2023)

The downfall of using a SPA for CipherCraft, even just for the content, comes down the re-rendering of the entire page, due to the large amount of information that CipherCraft wants to present this method did not seem applicable. Although it still statically storing the information it provides a more dynamic form of displaying the information, and leaves room for flexibility, but holds back on any chance of scalability.

Knowing that CipherCraft, requires dynamic content loading, flexibility, and scalability, the only choice was to store the content in a secondary location and fetch it when required. Which brought along the challenge of which is the best method of storing the information and how exactly should I call upon the information.

When making the decision on how to store the content, figuring out the best way for not only the written information but how each method of learning would be affected. All other learning styles were taken into consideration, meaning use of a database did not seem applicable due to the need of interactive functions, images, videos, etc. Storing all of these within a database would lead to complications down the road, such as the numerous amounts of querying back and forth with the database making sure the information is correct before displaying it.

Wanting CipherCraft to be quick and responsive, left only one true option, which was to use a JSON file, and structure the content accordingly so it becomes almost intuitive when displaying the content. Structing the content based on a Module, Topic, Page hierarchy, allowed for full control on what information would be displayed on each page. This method also allowed for the content to become dynamic, scalable, and flexible.

Developing CipherCraft as a dynamic platform was a significant milestone in my journey, requiring careful consideration of various technologies and methodologies to ensure a seamless user experience. Flask, Jinja, JSON, and Python, provided the requirements of turning CipherCraft into an interactive and responsive learning platform.

- **Flask integration:** Flasks played a key role in managing routes, handling user requests, and orchestrating the data flow within CipherCraft. By utilizing Flask's session management capabilities, I could personalise the user experience based on their interactions ensuring the content was dynamically delivered.
- **Jinja Templating:** The use of Jinja allowed me to embed Python logic directly into the HTML templates, facilitating dynamic content rendering. This allowed for the effortless integration of dynamic data retrieved from the JSON file into the frontend, enhancing the platform's flexibility.
- **JSON Data Storage**: The hierarchical structure of the JSON file served as the backbone of CipherCraft's content management system. Organising content, images, videos, and interactive functions within the JSON file enabled efficient data retrieval and seamless integration into the platform.
- **Python Backend:** Python served as the driving force behind CipherCraft's dynamic functionality, enabling real-time updates and interactions. Through Python scripts, I retrieved and processed data from the JSON file, dynamically generating content based on user sessions and interactions.

Developing a dynamic platform presented several challenges, including data retrieval, session management, and integration of interactive components. After trying different approaches and experimenting, I managed to overcome these challenges, hoping users have a seamless and easy experience.

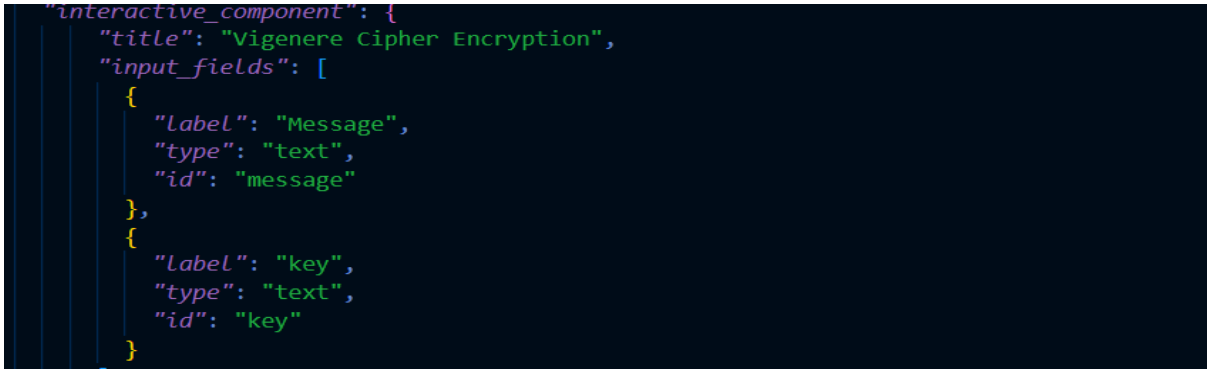
The JSON file employed a hierarchy structured approach, utilising placeholders for content, images, videos, and interactive functions. This organised structured facilitated efficient data storage and retrieval, allowing for the different learning styles to be applied within CipherCraft.

2.4.4 Learning Jinja

Learning Jinja was a beneficial achievement for the development of CipherCraft. Jinja is a powerful templating engine for Python, allowing for the integration of dynamic content into the HTML templates. Here is how my experience with Jinja enhanced the development of CipherCraft:

Dynamic Content Rending: Jinja's ability to embed Python code directly into HTML templates allowed me to dynamically render content based on user interactions and session data. This ensured that CipherCraft could adapt to users' progress. Additionally, using Jinja and the JSON file I was able to customise each of the interactive components CipherCraft includes, by stating which fields each component required.

```
{% for input_field in content_data.interactive_component.input_fields %}
<label for="{{ input_field.id }}">{{ input_field.label }}:</label>
<input type="{{ input_field.type }}" id="{{ input_field.id }}">
{% endfor %}
```



Being able to populate field ids within my HTML using a combination of Jinja and placeholders in my JSON file. Figuring that I could transport information from my JSON file to HTML tags/ids using Jinja was a great technical advancement to me. The two images above display the code used to achieve this. First assigning the input fields using a label, type, and id, which has no limit, meaning I could generate as many of them as I need. Then using Jinja templating engine, embedding of a loop for the number of input fields within the received number of input fields, populating the labels and inputs becomes easy.

2.4.5 Quiz Functionality

Implementing a quiz feature within CipherCraft was a significant technical achievement, requiring integration with a MySQL database and management of user sessions. Here is how the quiz functionality was implemented:

MySQL Database Integration: The quiz questions are stored in a MySQL database, allowing for efficient retrieval and management of quiz data. Utilizing a database for quiz storage ensures scalability and ease of maintenance.

Session Management: Upon entering the quiz, users are assigned session tokens to show they are currently taking a quiz, and to store the questions they are to be asked. These sessions allow for personalised quiz experience and ensures that quiz results are accurately assigned to the correct user.

Result Storage and Evaluation: Upon completion of the quiz, users' results are saved in the database, the result is then used to determine if a user can progress to the next module or not. If a user does not pass the quiz, they can retake the quiz using the retake quiz button, which will update the sessions and put the user back into the quiz.

Session Cleanup: To maintain system integrity and security, sessions are appropriately destroyed upon completion of the quiz. Additionally, an after-request route is utilised to clear the cache and ensure that users session data is securely managed.

The implementation of the quiz adds an interactive and engaging element to CipherCraft, allowing for its users to determine their level of knowledge of the content provided to them. The quiz additionally allows for CipherCraft to track how well users are doing within the quiz, providing a level of feedback on the questions and content, and if there is a lack of correlation between both. As well as providing a form of restricting users access to modules that may be to advance for them, which could lead to frustration and confusion.

2.4.6 Progress Tracking

The implementation of a progress tracking functionality within CipherCraft is another technical achievement, demonstrating the platforms commitment to enhancing the user experience. This feature allows CipherCraft to monitor and store user progress. Contributing to a more personalised and engaging learning platform. Key aspects of the progress tracking implementation include:

Integration of Flask Sessions: Using Flasks session management, CipherCraft tracks user interactions and progress throughout the learning platform. By establishing and maintaining session variables, the platform tracks each user's progress accurately and record it while preforming real-time updates. This functionality is implemented across various routes within the Flask application ensuring consistent tracking and synchronisation of user progress data.

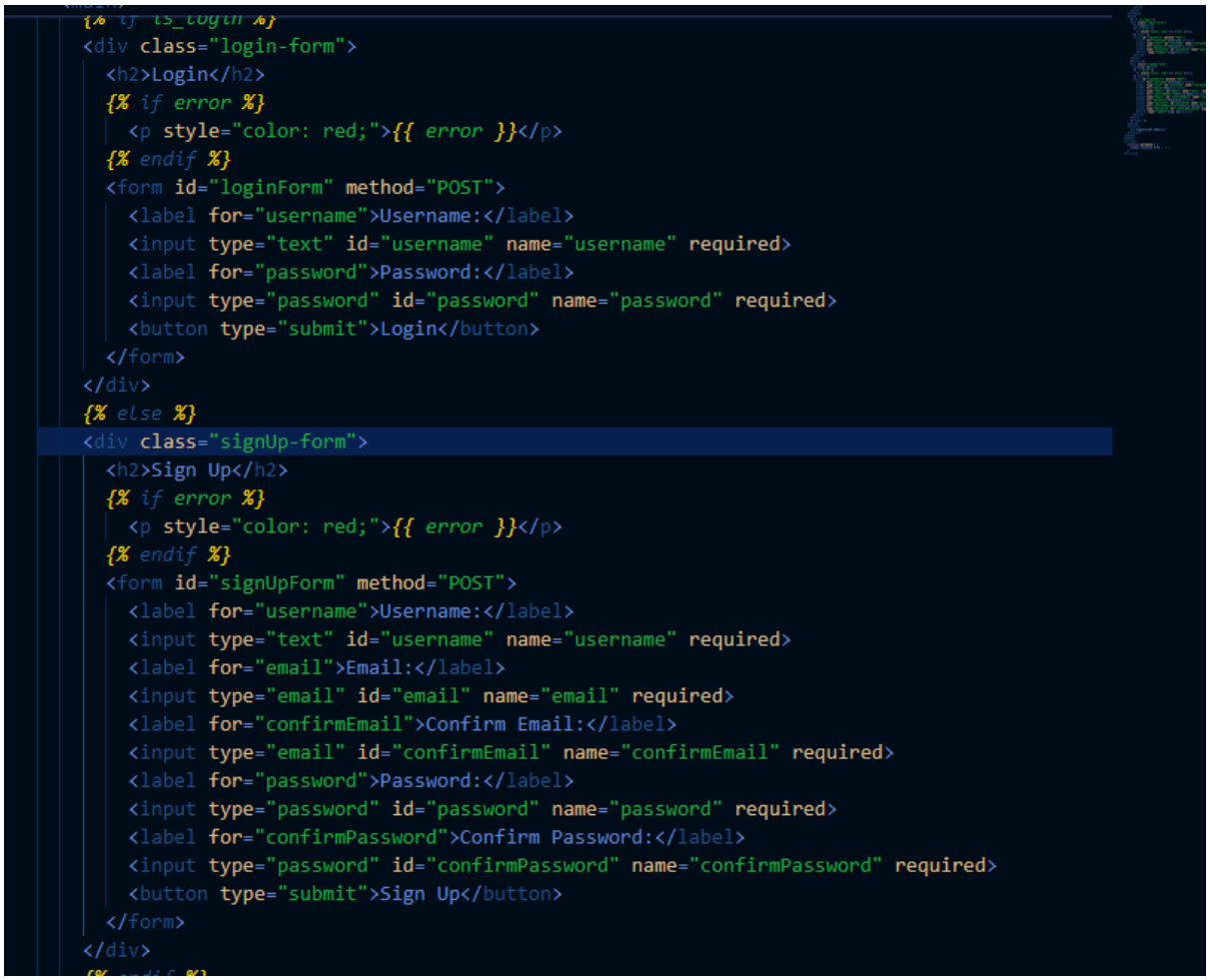
Database Integration: The use of the MySQL database allows CipherCraft to persistently store and retrieve user progress data. This database-driven approach enhances the scalability and reliability, ensuring that progress information remains accessible across sessions and devices. Within the Flask application, dedicated routes, handle interactions with the database, such as storing session data upon user progression or retrieving progress information for personalised content delivery.

The progress tracking functionality makes CipherCraft a learn at your pace environment, since the tracking is stored users are free to leave and come back at any stage and pick up right where they left off.

2.4.7 Authentication Functionality

CipherCraft preforms security around its authentication mechanisms, utilising authenticated tokens to determine whether a user is logged in or not, using Flask's sessions, to monitor login status for each user. Ensuring that only the authenticated users have access to the platform's resources. This approach allows us to keep track of active sessions and provide users with a secure experience.

Unified Login/Signup Interface: By utilizing a single HTML template, I dynamically adjust the content based on the user's access route. This approach displays a single-page application approach, which was a considered approach for the entire application, but seemed unjust based on the difficulties of managing the amount of code and the re-rendering of such a large document would slow down the responsive time of the application.



```
{% if is_login %}
<div class="login-form">
  <h2>Login</h2>
  {% if error %}
    <p style="color: red;">{{ error }}</p>
  {% endif %}
  <form id="loginForm" method="POST">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required>
    <button type="submit">Login</button>
  </form>
</div>
{% else %}
<div class="signUp-form">
  <h2>Sign Up</h2>
  {% if error %}
    <p style="color: red;">{{ error }}</p>
  {% endif %}
  <form id="signUpForm" method="POST">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required>
    <label for="confirmEmail">Confirm Email:</label>
    <input type="email" id="confirmEmail" name="confirmEmail" required>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required>
    <label for="confirmPassword">Confirm Password:</label>
    <input type="password" id="confirmPassword" name="confirmPassword" required>
    <button type="submit">Sign Up</button>
  </form>
</div>
```

Password Security: Users are required to enter their password twice upon signup making sure they entered the password intended and did not mistakenly hit another key. The password follows a strict policy, of twelve characters in length, mixture of upper and lower cases, symbols, and numbers. The password is then hashed before it is sent to the database, upon login the entered password of the user is hashed and compared with the hashed value, if the hash values are the same a user is granted access, otherwise they are redirected to the sign-up page.

```python
def passwordPolicy(password):
    if len(password) < 12:
        return "Password must be at least 12 characters long."

    if not any(char.isupper() for char in password) or not any(char.islower() for char in pa
        return "Password must contain a mixture of upper and lower case characters."

    if not any(char.isdigit() for char in password):
        return "Password must contain at least one digit."

    if not re.search(r"[!@#$%^&*(),.?\":{}|<>]", password):
        return "Password must contain at least one symbol (!@#$%^&*(),.?\":{}|<>)."

    return None
```

Encryption: To perform its encryption functionality, CipherCraft utilised the cryptography library, this library provided many different encryption algorithms and techniques. The encryption method used in CipherCraft is AES 256, Cipher Feedback (CFB) mode of operations. The reasoning behind using CFB is due to its data integrity operations, and how it acts as a stream-like encryption rather than a simple block-cipher. The encryption function is used on sensitive information gathered by the application such as emails. (IBM, 2021)

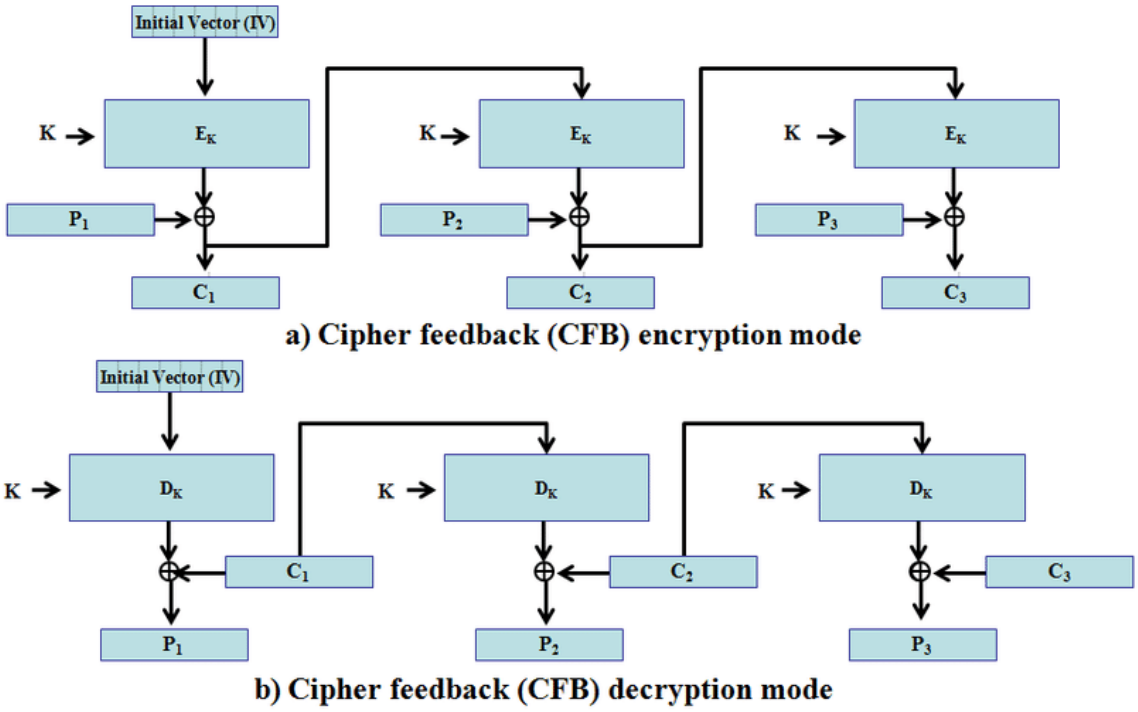


Figure 1(Maqableh, 2012)

The key used for encryption, is encrypted using a master key that is stored as an environment variable, storing the encrypted key in the database next to the encrypted field. This allows for a safe method of storing keys, for as long as the master key can remain unknown. At deployment stage, this method of encryption should be taken into consideration, as the use of a master key is a viable option, there are better forms of doing this such as vaults and hardware system modules.

2.5 Personal Achievements

2.5.1 Problem Solving

Dynamically Loading Content Using JavaScript and HTMX

Challenge: I encountered difficulties in dynamically loading content using JavaScript and HTMX, which posed a significant challenge with the performance and user experience of CipherCraft.

Solution: Known CipherCraft needs a more efficient method of displaying content over a static approach such as multiple HTML templates, or the use of a single-page application, the change to a backend content fetching functionality using Python came about, considering the two different approaches taken for frontend development of dynamically loading content created to many issues and troubles the only other option was to adapt to a backend version of the frontend implementation.

This functionality allowed for the retrieval of content down to the page number. Utilising sessions, which allowed for real-time updates, determining the page, topic, and module to retrieve the content, the ability of fetching content became much simpler. The solution did not just resolve the challenge but allowed for CipherCraft to become more scalable and flexible.

Personal Growth & Achievement: This challenge allowed for me to show off my level of resilience, as determination to create a dynamic method of displaying content, was ingrained in me. Being able to adapt to using a backend retrieval functionality when a frontend was the chosen method from the start, simply comes down the decision making made throughout the process of developing CipherCraft.

Dynamic Loading of Interactive Functions

Challenge: Implementing dynamic loading of interactive functions, content, images, and videos, presented another chance of discovering a solution.

Solution: The solution for this came with two choices, using a database to store the information, send a query to fetch the information whenever it is required, or using a JSON file with a hierarchical structure. Using a database, seemed like it would be a promising idea considering my familiarity with them from previous projects, but a with a deeper analysis, using databases would have limited the number of learning styles that could be used. As storing information related to interactive functions would have created a complex system of retrieving the required information for executing the function as well as the required input fields for them also. With that in mind, the only solution was to tackle a JSON file for the first time.

The setting up of a JSON file proved difficult, primarily due to the structure needed to store the data. As this was my first time ever using JSON, it took a little bit of time to get used to how these files should be structured. Once the file structure the rest became smooth sailing, as using keys to store the values of the content, images, videos, became a rinse and repeat matter. The issue lied in how I could get interactive functions to be truly dynamic and operable using this JSON.

Adding additional keys to determine if an interactive function is present or not, providing a key to store the function call, and then setting up keys that will store the required input fields needed for the function to execute was the solution that was used in the end. This method allowed for the dynamic loading of the interactive functions, and their required fields.

Personal Growth & Achievement: Facing this challenge I was able to develop my problem-solving skills, through my quick learning, such as using a JSON file for the time, especially one with such a large structure, and my creativity in using keys within the JSON file make the interactive function completely dynamic.

Tracking User Progress

Challenge: The need to track user progress within the platform posed another challenge, as reliable, real-time progress tracking and updates were required to enhance user experience.

Solution: The solution for this was using different sessions, to determine which module, topic, and page a user was currently on, then when a user clicked the next or previous button, these sessions would update one at a time, and as these sessions updated, so would the values within the database. Meaning whenever an update happened to the sessions their values were sent to the database and stored. This allowed for real-time updates on the progress as well as allowing for a learn at your own pace environment, as users could leave and come back to where they left off within the content.

Personal Growth & Achievement: This problem allowed for an analytic approach on discovering a solution, knowing that CipherCraft needed some method of progress tracking, to provide different benefits, such as a learn at your own pace environment.

These examples demonstrate the growth in my problem-solving skills, from my resilience in creating a viable to solution, to creative approaches, as before I would never come these conclusions, but the development of CipherCraft has tested my problem-solving skills at every turn, allowing me to grow this skill.

2.5.2 Project Management

Taking on the ambitious project of developing CipherCraft while having assignments from other modules, commitments to my work life and maintaining a social life, within the semester system for the first time has been a transformative experience for me. This challenge provided me with the prefect opportunity to build upon my project management skills. Due to the numerous tasks and deadlines, finding the perfect balance was difficult, learning to allocate time to resources and meeting the demands of each module, and deliverable proved difficult. As I worked through various parts of this project, from writing documents, making research posters, and giving presentations, I could see my organisation skills increasing. The pressure of the semester system left little room for procrastination, instead demanding urgency with every assignment. This journey not only showed my ability to work under pressure but also my ability to adapt in a dynamic environment.

3.1 Project Review

This section is provided a reflection on the development process of CipherCraft. Identifying areas for improvement trough a different approach if I had started all over again, any missing/incomplete features, and recommendations for anyone else who wishes to tackle a project like CipherCraft.

3.2 What I would do differently

Research more into other learning platforms:

If the opportunity presented itself to start this project over from scratch, what would change? The biggest change I would make, is during the research aspect of this project, instead of preforming my research with a narrow scope, homing in on application that only provide learning experience related to cryptography, I would expand this research to multiple learning platforms, such as NetaCad, HackTheBox, and any other learning platform available, as it would have presented insights into how those applications function and operate.

By gaining insights into more than just two different learning platforms, CrypTools and CryptoPals, CipherCraft's development stage could have been smoother, knowing how other applications also preform different tasks, how they lay out the content, and how they approach their assessment style, could have provided influence into the development of CipherCraft.

Scale back the number of modules:

Another aspect that would be put under review, is the number of modules that are included in CipherCraft, which is five. At first five modules does not seem like a major number. During the development process, my mindset swiftly changed. Having five modules, containing four topics, with ten pages of content for each topic, equating to two hundred pages of content, which is a lot for one person to develop, while doing their best to ensure that each page includes multiple avenues of learning.

Taking on a second iteration of this project I would remove the final two modules, hashing algorithms, and key management, which would have drastically changed the amount of time spent on researching, developing content, and the number of pages that requires different learning styles. Allowing for CipherCraft to potentially reach a higher level of completion, and the ability to include other features and functionality.

Systematic approach:

During the development of CipherCraft, the approach was to complete it section by section, i.e., complete all the content first, then complete the quiz, etc. This approach resulted in CipherCraft not getting to its highest level of completion as creating the content for the five modules took much longer than expected.

If the development process was to be redone, the approach I would take is to complete module one, then do the corresponding quiz, then complete the content for module two, and then its quiz. This approach would have allowed for CipherCraft to have reached a stronger level of completion as after each module is complete it would have a corresponding assessment.

Manage my time more efficiently:

From working on assignments for other modules, working part-time, and trying to hold a social file, while being my first time in a semester-based system, it almost felt as if time did not exist, due to the lack of it. From commuting every morning and evening to and from college, to working late hours on the weekend, it became a struggle to manage everything at once.

Had I known in advance the amount of time that is required to complete a project such as CipherCraft, I would have created a strict calendar, to follow, which would have accounted for every hour of every day, with a plan of action, to work around the busiest year of my life.

3.2 Missing/Incomplete features

Due to the high demand of other functions, such as the ability of making CipherCraft as dynamic as possible, some features/functions were not addressed, or not completed.

Content: Throughout the five modules, each page does contain some level of information, but not every module contains images, videos, or interactive functions. The only module that is fully complete is module one, as it contains images, videos, and interactive functions. The primary reason for the lack of learning style inclusion within the other modules is due to the lack of time, and the challenging process of making CipherCraft as dynamic as possible. As ensuring the dynamic aspects of CipherCraft really took up a lot of my time during the development process, as I had to change from JavaScript to HTMX, to Python. This took 2-3 weeks of development time resulting in less time to work on other sections and putting me behind schedule.

Quiz: As the demand for completing functionality, over content was as the forefront, this made it difficult to find time to complete questions for each module. Currently CipherCraft only has questions available for modules 1-3 each containing a pool of forty questions where a random fifteen is selected, all questions are multichoice, had more time been allocated, the quiz functionality and questions would have been designed differently. The quiz would have taken a more considerable approach by ensuring users are getting asked appropriate questions, based on previous results, and the questions would have taken on different forms, such as including images, true/false questions, complete the statement question etc. Due to the lack of time, the quiz only provides a basic approach in assessing the users.

Labs: CipherCraft intended on having labs for each module, providing a user with a step-by-step approach on how to complete/programme their own encryption algorithms, using different programming languages, with real world application. Unfortunately, due to the challenges that other sections of development brought, this feature never managed to reach the light of days, this made CipherCraft lack in providing users with the knowledge required for implementing cryptographic algorithms.

Non-Core Features: Other features not related to the core delivery of the project were also omitted due to, time constraints created by challenges, and other commitments, include a forgotten password/reset password functionality, from past experiences, I fully know the challenges and difficulties of getting functionality like this to be correctly implemented using emails. The lack of gamification within the application is also left to be desired, as focusing on getting the content to be dynamic was the primary focus. The ability to get user feedback is yet another non-core deliverable that was not meet due to time constraints. Other features such as a discussion board and extra educational resources were not implemented due to the lack of time, and the need to complete core deliverables.

3.3 Recommendations

This section of the document is to provide my personal recommendations for anyone that is planning to develop a project like CipherCraft, for the areas they should research, to the different insights I achieved during this project.

Further Research: During my research process, my primary focus was on how individuals best learn, to provide a strong learning experience for the users of CipherCraft to ensure they absorb the information from the application. This stopped me from focusing on other applications that preform the same tasks as CipherCraft, my recommendation would be to discover platforms that also take into consideration the abundance of learning styles and incorporate them into their application to this will provide extra insight into the development process and could significantly assist for the user interface on how to layout information.

Another section I felt lacked research for me, was the technologies. I recommend expanding upon that area, as I had never heard of HTMX, until it was recommended to me by a supervisor. I never knew of Jinja's full capabilities, even still I do not know all its functionality. There is likely an abundant of technology stacks out there that could have made the process of developing CipherCraft much easier, so recommendation is to further research the best technologies for developing a learning platform.

Security As I Go: Another area I would recommend is to preform security tasks with each step, if a new page or route is created, ensure that session management aspect is also implemented. If you are requesting user input, make sure to sanitise it. This will not only allow for the current aspect of your project to be secure, but it will also save time in the future from having to rush back and implement these all at once and provides the groundwork on how to implement them for each new section.

4.1 Relevance to Cyber Security

Cyber security is a difficult term to define due to how it has been used consistently within different definitions. Each time the term cyber security is used it surrounds the following, safeguarding cyberspace and the system connected through organizing, managing resources, processes, and structures. The main goal of cyber security is to protect cyberspace from incidents that break the legal and actual property rights. (Casey, 2024)

There are three main fundamental principles that are used within cyber security, confidentiality, integrity, and availability. These fundamental principles are typically extended to adding accountability and authenticity. These principles are used to prevent cyber-attacks such as data breaches and denial of services. (Casey, 2024)

As the purpose of cryptography is to provide confidentiality, integrity, and authenticity. The fundamental principles used throughout cyber security are confidentiality, integrity, availability, accounting, and authenticity. As cryptography provides three of the fives principles it is a clear indication that cryptography should be considered a pivotal topic used throughout cyber security. (Casey, 2024)

Cryptographic techniques are used in many different avenues such as, social media platforms, communications systems, finical transactions, healthcare systems, the list goes on. This demonstrates the demand of individuals who can perform different cryptographic techniques in the real world. (Casey, 2024)

As cryptography proves to be an effective method of protecting sensitive and valuable information from criminals, its usage has become more common in the field of information security. The need for cryptography in the modern world is evident through the different platforms such as WhatsApp, Facebook, Twitter, the online shopping market that all use the different cryptographic techniques to secure the sensitive information. (Casey, 2024)

Without cryptography sensitive information such as credit card details, passwords, and more would be visible for criminals to see and steal. When cryptography is used it turns the information into an unreadable format, this does not mean that the confidential information is secure. (Casey, 2024)

The level of security when cryptography is used, consists of how the information and techniques are handled within the process, for example how the key is created, stored, and used during the cryptographic method. The reason teaching cryptography is especially important, because understanding the process is what allows for correct handling of the information. (Casey, 2024)

CipherCraft aims to teach people the fundamentals of cryptography, starting from the very beginning such as the early encryption methods like the Caeser Cipher and bringing its users to a strong fundamental understanding. The content that CipherCraft provides ranges from the early encryption process to key management. Exploring different sections like symmetric vs asymmetric encryption, a deep dive into the Advanced Encryption Standard and the different modes of operations.

CipherCraft contains a structured curriculum, preventing its users from delving into advanced topics when they are unfamiliar with the basics. For a user to progress in CipherCraft they must pass a quiz to ensure an understanding of the concepts for the previous module, this will stop users from becoming overloaded with too much complex information at once, leading them to frustration.

The target audience of CipherCraft are students in third level education, primarily those doing computing courses such as, cyber security, software development, and IT management, as these courses are the ones that will likely be able to utilise the information provided from CipherCraft, within their project work, and future career. Providing them with the knowledge they need to understand the fundamentals of cryptography giving them a foot hold in the complex topic.

As CipherCraft utilises different learning styles, its aim is to provide users with their preferred style of learning. This will in exchange allow for them to obtain the information better leading to a stronger understanding of the cryptographic concepts used throughout the platform.

References

Adobe Experience Cloud Team. (2023, 19 7). *Single-page applications (SPAs) — what they are and how they work*. Retrieved from https://business.adobe.com/: https://business.adobe.com/blog/basics/learn-the-benefits-of-single-page-apps-spa

Barguzar, A. (2024, 2 8). *Python Flask versus FastAPI: Which Should You Choose?* Retrieved from https://www.netguru.com/: https://www.netguru.com/blog/python-flask-versus-fastapi#:~:text=Flask%20is%20a%20lightweight%20and,application%20up%20and%20running%20quickly.

Casey, J. (2024). *Research/Spec Document*.

Flask. (2010). *Templates*. Retrieved from https://flask.palletsprojects.com/: https://flask.palletsprojects.com/en/2.3.x/templating/

IBM. (2021, 06 25). *Cipher Feedback (CFB) Mode*. Retrieved from https://www.ibm.com/: https://www.ibm.com/docs/en/zos/2.4.0?topic=operation-cipher-feedback-cfb-mode

Maqableh, M. (2012). *Cipher feedback (CFB) mode*. Retrieved from https://www.researchgate.net: https://www.researchgate.net/figure/6-Cipher-feedback-CFB-mode_fig4_265234045