# **Project Title**

# **Creation Unblock: The integration of CGAN and NLP to provide a superior machine learning creativity and compositional tool for the music industry.**

Written by Joshua Nyarko

BSC Hons Computer Science

Table of Contents

[Project Title 1](#_Toc166450220)

[Creation Unblock: The integration of CGAN and NLP to provide a superior machine learning creativity and compositional tool for the music industry. 1](#_Toc166450221)

[Chapter 1: Introduction 4](#_Toc166450222)

[1.1 Background 4](#_Toc166450223)

[1.2 Aim of the project 6](#_Toc166450224)

[1.3 Project Scope 7](#_Toc166450225)

[**1.3.1 Objectives of the project** 7](#_Toc166450226)

[**1.3.2 Project deliverables** 8](#_Toc166450227)

[**1.4 Project Plan** 9](#_Toc166450228)

[Chapter 2: Literature Review 9](#_Toc166450229)

[2.1 GANs and CGANs 9](#_Toc166450230)

[2.2 The application of GANs and CGANs in music 11](#_Toc166450231)

[2.3 The application of NLPs and TensorFlow to music generation 13](#_Toc166450232)

[2.4 Research gap 15](#_Toc166450233)

[3.1 Types of research 16](#_Toc166450234)

[3.2 Research Plan 18](#_Toc166450235)

[3.3 Development Methodologies 18](#_Toc166450236)

[3.4 Domains 20](#_Toc166450237)

[**3.4.1 Machine Learning** 20](#_Toc166450238)

[**3.4.2 GANs and CGANs** 21](#_Toc166450239)

[**3.4.3 NLP** 21](#_Toc166450240)

[**3.4.4 TensorFlow** 22](#_Toc166450241)

[**3.4.5 Datasets for CGAN Training** 22](#_Toc166450242)

[3.5 Testing Strategies 22](#_Toc166450243)

[3.6 Analysis of Research 23](#_Toc166450244)

[**3.6.1 Analysis** 23](#_Toc166450245)

[**3.6.2 Choice of language** 23](#_Toc166450246)

[**3.6.3 Choice of Development Environment** 23](#_Toc166450247)

[Chapter 4: Design 24](#_Toc166450248)

[Chapter 5: Development 26](#_Toc166450249)

[Chapter 6: Conclusion and Future Works 33](#_Toc166450250)

[Running The Application 35](#_Toc166450251)

[Bibliography 36](#_Toc166450252)

# **Chapter 1: Introduction**

The Creation Unblock project focuses on the intersection of music, creativity and artificial intelligence. The concept behind Creation Unblock is Shazam, the music discovery application, that uses machine learning and other artificial intelligence to assist users in exploring music and finding similar songs and compositions. Creation Unblock aims to go beyond this process, developing a machine-learning creativity tool for music producers, artists, and listeners. As part of this process, the project aims to revolutionise the music creation process. Specifically, in a world saturated with digital content, the Creation Unblock program will seek to empower musicians and enthusiasts through the creation and introduction of a novel approach to music generation and exploration. The primary aim of this process is to develop a sophisticated machine-learning tool that can identify existing songs, lyrics and compositions, whilst also using these to facilitate the creative process for music producers and artists. This will be achieved by harnessing the power of Generative Adversarial Networks (GANs), Conditional GANs (CGANs), and TensorFlow, enabling users to seamlessly blend inspiration and innovation in their music composition journey. This project thus seeks to bridge the gap between recognition and creation, enabling new ways to discover and produce creative music compositions.

## **1.1 Background**

Whilst music has been studied and theorised extensively over the previous centuries, the concepts of musical imagination and creativity remain among the most abstract and complex elements of music and composition. Until recently, these concepts were rarely subject to extensive empirical enquiry, due to the challenges involved in this process. However, according to Hargreaves et al (2012) recent developments in music psychology and allied disciplines, including sociology, neuroscience, and education, have facilitated a general grasp of musical creation, to the point at which the study of musical creativity and imagination is now distinctly possible. At the same time, the growth, development and ubiquity of digital music technology have enabled both researchers and scholars to consider and examine how musical creativity can be enabled and encouraged through the use of said technology. This has previously focused on the use of these technologies as broad support for music learning and creativity, such as facilitating digital sampling and beat making (Kladder, 2016). However, as technology has continued to develop apace, so it has become possible to facilitate the generation of music through digital technology.

Unfortunately, the overall process of generating music remains complex, due to several notable differences with the other generative uses of technology, such as the generation of images and videos. According to Ali Elfa and Dawood (2023), this is due to three main issues. Firstly, music is an art of time and hence requires a temporal model to structure the creativity. Secondly, music involves the composition of multiple instrumental tracks, each with its temporal dynamics but also which must unfold over time interdependently to create a strong and coherent rhythm. Finally, musical notes are not generally produced individually but must be grouped into chords, arpeggios or melodies to support genuine polyphonic music, making it difficult to create a chronological ordering of notes. All of these elements create significant challenges to the use of generative AI, and its ability to support or enhance human creativity. As a result of this, developments in the field of music generation have generally looked for techniques and methods to attempt to overcome these issues, with varying levels of success. For example, Dong et al (2018) used the GAN framework to create three models of AI-enabled music composition under different assumptions and network architectures. These are referred to as the ‘jamming model’, the ‘composer model’ and the ‘hybrid model’, and can be used to generate different tracks and coherent music of up to four bars in length. However, they also rely heavily on human-AI cooperation to move beyond this limited scale.

This limitation is recognised more widely within the overall literature on generative AI and its use in music. In particular, the focus of research is presently on how AI can expand the creative process by enabling individuals to overcome the limitations of the human brain, thus inspiring new designers of creativity (Ali Elfa and Dawood, 2023). This hence highlights the potential for AI to be used to expand the creative process, whilst also noting the importance of human engagement and interaction to overcome the limitations of AI systems. Such arguments are supported by several existing projects and examples, which showcase the potential of combining machine learning with music composition, but with a stronger focus on melodies rather than lyrics. For example, Magenta Studio by Google explores the intersection of art and artificial intelligence, offering tools for musicians and artists to experiment with generative models in music creation; AIVA (Artificial Intelligence Virtual Artist) is an AI composer that generates classical music compositions; and IBM Watson Beat uses AI to compose music across different genres, highlighting the adaptability and versatility of generative models in music. Each of these systems highlights the capacity of AI to produce music that resonates with human emotions and preferences. However, they tend to focus on the musical melodies rather than the lyrics, thus showing the importance of a further focus on the application of AI to assist music producers, artists, and listeners in the lyrical creativity and compositional process, rather than just the creation of musical melodies.

## **1.2 Aim of the project**

The Creation Unblock project aims to apply emergent artificial intelligence technology to assist in the development of lyrical creativity and composition in the music industry. This is reflected in the core product, “Creation Unblock”, a machine-learning creativity tool that will aim to assist music producers, artists, and listeners through the application of AI technology to the music creation process. Such tools can draw on extensive secondary evidence, including from case studies of the music compositional process. For example, the work of Collins (2005) looked at various data sources including digital MIDI files, analogue audio files, semi-structured interview data, retrospective verbal accounts of the compositional process and various verification sessions in the compositional process. Such analyses have highlighted key moments of creative insight, facilitating the synthesis of differing creative process theories and a level of understanding of the compositional process. This hence provides the potential to apply such insights and understanding to the development of a novel approach to music generation and exploration, integrating GANs, CGANs and NLP systems to develop a new approach to music. In particular, the project will look to build on the example of Shazam and its music recognition algorithm, by integrating AI systems to enhance overall lyrical compositional outcomes (Jovanovic, 2023).

Within this process, the primary aim of Creation Unblock is to develop, test and refine a sophisticated machine-learning tool. This tool will not only identify existing songs, but will also look to facilitate the creative process for music producers and artists through the integration and testing of advanced AI software. Specifically, by harnessing the power of CGANs, TensorFlow, and NLP, Creation Unblock will aim to enable users to blend inspiration and innovation within their music composition journey, offering a seamless method to develop new lyrics. The project hence aims to bridge the gap between recognition and creation, using existing lyrical compositions for inspiration in a manner which can transform how music is discovered, recognised, experienced and produced. As part of this process, the project will be managed through a systematic approach, aiming to employ agile methodologies to ensure adaptability and efficiency throughout the development lifecycle. In addition to this, regular sprints and feedback will be integral to the project management strategy, aiming to foster a more dynamic and responsive development environment.

## **1.3 Project Scope**

### **1.3.1 Objectives of the project**

The project will look to leverage publicly available music databases coupled with Application Programme Interfaces (API) to compile a comprehensive and diverse dataset to develop the Creation Unblock program. The dataset will be curated to encompass a wide spectrum of musical genres, thus ensuring the model's ability to generate harmonies across different styles. The following objectives identify how the project will be processed, and how the success of the project will be measured. This will involve various quantitative and qualitative objectives and criteria that will enable the project to fulfil its aims whilst also ensuring the effectiveness, usability, and impact of the tool. The key objectives for the project are thus identified as follows:

1. To gather a diverse dataset of music lyrics that will be sufficient to train the machine learning models, whilst ensuring representation across genres and styles
2. To use TensorFlow to implement GANs, CGANs and NLP to create a robust model that is capable of generating harmonious and original music lyrics
3. To develop an intuitive and user-friendly interface which can seamlessly integrate Creation Unblock into the creative workflow of music producers and artists
4. To implement a recognition feature, based on that of Shazam but integrating machine learning and AI, to enable users to identify songs and instantly access related creative suggestions
5. To conduct rigorous testing and iterative development of the models and features, based on user feedback, to ensure continuous improvement
6. To achieve a minimum level of accuracy of 80%, in terms of stylistic coherence and artistic relevance across various genres, when generating music lyrics
7. To attain a user satisfaction rating for the tool of at least 4 stars out of 5, based on at least 200 items of user feedback
8. To maintain an average user engagement period of at least 5 minutes per session, demonstrating sustained interest and utilisation of the tool
9. To achieve an accuracy rate of 90% or higher for the recognition feature, in terms of the correct identification of songs
10. To demonstrate the ability of Creation Unblock to generate harmonious lyrics over a diverse range of music genres, based on qualitative user feedback
11. To ensure the seamless integration of Creation Unblock into existing music production workflows, as reflected by at least 50 positive user testimonials
12. To establish a reliable and valid feedback loop mechanism that can collect user suggestions and critiques, and incorporate at least 80% of addressed feedback into tangible improvements to the tool

### **1.3.2 Project deliverables**

The project will look to produce the following deliverables:

* A project report including a review of the literature and details of the chosen methodology
* A project plan
* Documentation showing the research and decisions made throughout the project
  + A list of requirements for the tool
  + Design documentation following a suitable design methodology
  + Conceptual diagrams of the system
  + Entity relationship diagram of data set being used for testing the neural network
  + Data features of input of algorithm
* The Creation Unblock tool
  + Details of the coding undertaken to deliver the tool
  + Test results and related documentation around the iterative design choices and processes
  + Details of the sprints and feedback process used in developing the tool
  + Documentation showing self-evaluation of the project decisions and outcomes

# **1.4 Project Plan**

A Gannt chart will be used throughout the project, and it will evolve as it goes along. In the Appendices variations of the Gannt charts will be included. (Appendix A)

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Activity** | **Start Month** | **End Month** | **Duration** |
| Research Analysis | December | January | 30 Days |
| Design | January | February | 30 Days |
| Mid-point review and Contingency | February | March | 30 Days |
| Implementation | March | April | 30 Days |
| Testing | April | May | 30 Days |
| Finishing / Hand In | May | May | 30 Days |

Table 1 provides a summary of the starting dates for each project phase.

## **Chapter 2: Literature Review**

## **2.1 GANs and CGANs**

Within the wider computer science literature, there is a significant focus on the potential role of GANs and CGANs as being a strong potential contemporary research focus for the development of artificial intelligence. Within this process, GANs are recognised as being networks which comprise both a generator element and a discriminator element, enabling them to generate new samples from the distribution of real data samples. These in turn can be trained under the adversarial learning idea to generate further new samples, and thus provide substantial algorithmic support for the development of parallel intelligence across multiple different fields of practice (Dash et al, 2023). This hence enables GANs to potentially provide a new method for computer systems to learn deep representations, without requiring the extensively annotated training data which is often required by other forms of AI. This can be achieved by using backpropagation signals, which are spread through a competitive process involving the use of a pair of networks. This in turn means that “by designing different neural network layers in the generator and discriminator, sequential data such as text and unstructured data such as graphs can all be ingested and learned by the network” (Wu et al, 2022, p2). Such variety makes GANs a potentially important and effective AI technique that is worthy of further study.

Beyond the simple GANs, the conditional version of generative adversarial nets has been developed recently. Such tools can be constructed by simply feeding the data set into both the generator and discriminator, enabling the model to condition itself. In theory, this model can generate new forms of insight conditioned on class labels, which can in theory be used to learn multi-modal models, and in turn to generate new descriptive tags for data sets which are not part of the initial training labels, supported by the adversarial principle underpinning the GAN learning process (Chakraborty et al, 2024). As such, CGANs have the potential to support the expansion of GANs beyond their simple initial focus, towards developing new data sets, conditions and tags. This has enabled GANs, as a form of a deep neural net, to succeed when applied to many long challenging computer vision problems. This includes semantic segmentation, which is required for autonomous driving systems, as well as supporting the main architectures of GAN that are used for segmentation or classification of complex medical image diagnostic applications (Jeong et al, 2022). In this regard, the general training of deep neural nets requires huge amounts of labelled data, which requires significant effort and resources to collect and annotate. As such, the use of CGANs to develop new conditions and tags for data sets can help drastically reduce the amount of time and effort which must be expended in these processes, and in turn, enhance the value of the resulting tools and outputs. In particular, the conditional generator can be used to transform synthetic features into real-life like features, with an associated discriminator to distinguish them, enabling the generator to produce features that fool the discriminator, and in turn enable it to learn and become more effective in future (Shahbazi et al, 2022).

This potential usage of CGAN has enabled the technique to be used as a deep learning technique that applies meta-learning techniques to overcome temporal issues in problem scenarios. For example, in the case of multi-period choice problems, Sun et al (2023) showed that CGAN can be used to generate classifications for the time series data provided, including its division into training and test sets. The CGAN model can then generate data, from the training sets, which can be used to train and develop the time series model further using the resulting expanded dataset. In theory, this enables CGAN algorithms to analyse time series data over a longitudinal time horizon, as well as immediate data sets. A similar value is highlighted for the classification problem, namely predicting which class an input data item belongs to. In general, this is achieved by using a machine learning algorithm applied to the given dataset. However, in practice, this requires the dataset to have a well-balanced class distribution to provide strong outcomes. As such, Kang et al (2023) argue that the CGAN model can be used to overcome an imbalanced class distribution when applied to the case of text-to-image synthesis, enabling GANs to benefit from the use of larger datasets without being restricted by class distributions. Such applications highlight the potential use and value of GANs and CGANs when used to address specific issues and problems that may arise with real-world data sets and their constraints.

As a result of these arguments, GANs and CGANs have been consistently highlighted as being among the most successful generative models in use. However, they do also present unique challenges and associated research opportunities, justifying their inclusion in high-level research and their application to complex problems (Shahbazi et al, 2022). In particular, GANs can be conceived as an artificial intelligence algorithm that is designed to solve a particular generative modelling problem. This in turn means that GANs have been developed and successfully applied to a wide variety of tasks, but also that they need to be applied to these particular tasks. For example, in recent years, impressive progress has been made regarding the development of implicit probabilistic models using GANs and CGANs, which have been shown to provide excellent solutions to image-processing applications, involving large and continuous output spaces (Ding et al, 2023). However, on the other hand, there is limited evidence of the effective application of these powerful tools to problems involving small dimensional output spaces. Beyond this, the value of GANs and CGANs is also limited by their core algorithms being based on game theory, as opposed to most other forms of generative modelling and algorithms which are instead based on optimisation (Shahbazi et al, 2022). This raises questions about the potential to apply GANs and CGANs to real-life issues such as music recognition, where optimisation may be needed for some cases.

## **2.2 The application of GANs and CGANs in music**

The application of artificial intelligence and generative technologies to music recognition and generation is recognised as being particularly important in the contemporary context. In particular, following the development of deep-learning frameworks for acoustic hologram generation, attention has shifted towards music compositions (Lee et al, 2022). This particularly includes the ability of said technologies to aid in the compositional process, without creating potential issues of copyright infringement or allegations of copying existing music. For example, Azahad and Hameeda (2023) note that GANs exploit available prior knowledge when generating new forms of realistic data, including within music composition, which in turn creates the potential for legal and ethical issues associated with their application. This is reflected in the use of chord sequences and priming methodologies as inputs into GANs requiring care to ensure that prior knowledge is not plagiarised. Despite these issues, the extensive development of deep learning and GANs within the field of music has continued, and GANs have been used in music generation to produce original compositions of music in various genres. In particular, “alternatives, such as WaveGANs and musical instrument digital interface (MIDI) GANs, have demonstrated encouraging results in producing realistic and varied musical compositions” (Wijanto et al, 2024, p30980). As such, GANs and CGANs are now increasingly seen as being a key technology in the development of new music compositions, enabling the creation of a range of new melodies and harmonies, which can then be reviewed and improved by human artists and producers to create new music forms.

The performance of GANs and CGANs in this area is particularly impressive given the identified challenges regarding the generation of music by AI, when compared to the generation of images and videos. In particular, music composition requires a combination of a temporal model with lexical and acoustic inputs to develop a realistic matching of melodies, lyrics and beats (Vidal and Busso, 2023). In this regard, the framework of GANs can play a strong role in enabling consideration of differing underlying assumptions and network architectures, suiting a wide range of compositional cases. At the same time, Adhikari (2023) found that generating coherent raw audio waveforms using GANs is challenging, and requires additional architectural changes and the use of further training techniques. This includes the development of broader frameworks for harmonisation using GANs to ensure that the generated compositions carry some musical weight and value, ensuring the qualitative quality of the resulting music outputs.

Beyond this application, further research on automatic music generation has supported the use of CGANs as part of the development of deep neural networks applied to the generation of music. However, evidence from this research indicates that, despite the developments in this area, GANs still face challenges in areas such as mini-batch discrimination and batch normalization, potentially resulting in model collapse and gradient problems with the volume and variety of data (Zhong, 2023). This can create problems with complex multi-instrument compositions, which require the use of recurrent convolutional generative models to ensure the harmonic features of the associated arrangements and different genres. As such, efforts have been made to support the use of deep learning techniques in a more structured manner, using multiple GAN models and associated post-processing techniques to generate diverse samples and support more effective compositional outcomes. For example, Bastianello (2022, p1) proposes a pipeline comprising “SpecGAN, employed to generate one-second inharmonic samples, UNAGAN, used to generate variable-length harmonic samples [and] post-processing, applied to minimize the amount of noise in the generated samples. This approach is shown to support the generation of variable-length compositions, including both harmonic and inharmonic melodies, with quality comparable to that of real samples. However, the length of these offerings remains somewhat limited, as does their value and use in the wider music context.

## **2.3 The application of NLPs and TensorFlow to music generation**

As can be seen from the literature above, the majority of the focus of the use of GANs and CGANs to date, in both the general and the music case, has focused on nonverbal elements, such as the generation of images and chord sets. This reflects the challenges involved in the use of AI to develop verbal data, particularly the importance of the natural language processing (NLP) function. In general, there have been efforts to achieve this, including the development of a proposed method “for combining the transformer deep-learning model with generative adversarial networks (GANs) to explore a more competitive music generation algorithm”, which incorporated NLP to address the challenge of how to develop viable lyrics (Min et al, 2022, p2515). Within this element of the research process, different deep learning models have been used to generate music, often using NLP metrics to attempt to evaluate models and to validate them against a baseline, as well as against a population of human listeners (Ferreira et al, 2023). However, within this process, the concept of using NLP for genuine text generation, in the form of music lyrics, is still largely theoretical, with NLP generally only widely used for reference. As a result of this, most of the resulting models lack the necessary integration to deliver results and can struggle to process and generate the forms of long sequence music that the Shazam algorithm is intended to handle (Min et al, 2022). This in turn indicates that, to achieve the genuine generation of music lyrics through GANs and CGANs, it will be important to integrate genuine NLP capabilities into a new and superior machine learning creativity and compositional tool.

Within this process, it will be important to draw on existing developments in deep learning and wider artificial intelligence, particularly employing deep neural network architectures that can support natural language processing and other associated domains such as speech recognition. In this regard, one of the most critical developments in recent years occurred in November 2015, when Google released TensorFlow as an open-source deep learning software library that can be used to develop, define, train and deploy new forms of machine learning models (TensorFlow, 2024). This has been further refined with the development of TensorFlow 2, supporting the creation of new forms of generative AI using Python coding to create images, text, and music. TensorFlow has thus become established as a key driver of modern deep learning concepts and software, with the potential to support NLP and to enable the creation of sophisticated algorithms which harness the capabilities of neural networks (Kulkarni et al, 2023). The key properties of TensorFlow in this regard is the ability of the machine learning system to represent computation, shared state, and the operations that mutate that state through appropriate dataflow graphs, making it “the most common and successfully used library that provides various tools for machine learning applications” (Vinnay et al, 2023, p8). This in turn provides significant support for training and inference developments on deep neural networks, and the integration of NLP into this context.

Further to this, recent advances in associated machine-learning tools have also served to empower a wide range of application developers in various creative contexts. This is reflected in the growing use of TensorFlow as a method to use AI technologies without demanding domain expertise, and to pretrain models related to various music tasks. In particular, Guo et al (2023) note that TensorFlow can be used to generate multiple melodies and train them to fit in line with the auditory habits of listeners, creating good results in the testing process. This highlights the potential value of TensorFlow in a generative context, using the NLP systems that can classify music by voice to develop further insights into the wider compositional process, and in turn to develop new forms of lyrical arrangements. Indeed, there is already evidence of the development of potential datasets that can be used for music generation, incorporating learning from NLP and Python coding. For example, Deepaisarn et al (2023) highlight the value of NLP in supporting the use of classification machine learning algorithms to compare performances and classify musical compositions. The results of this study indicated that classification performance was sensitive to the feature extraction methods used to support the NLP process, requiring a significant focus on musical words and sub-word vectors to support effective lyrical recognition. However, this study focuses on classification processes, thus in turn leaving a gap in the literature regarding the development and application of NLP and wider music composition AI systems in a cross-genre context.

More recent research in the field has continued to highlight the extent to which AI generation of various musical elements remains a highly challenging problem that is yet to be solved. For example, Aditya et al (2023) look at how music-driven choreography systems can be developed using RNNs and LSTMs. This study highlights how AI can be trained against previously successful music to generate new and inspired music, and that this enables professional musicians to create better music faster, whilst at the same time recognising the importance of originality. Such an element reflects the process of building a model that can compose new music using data sets that contain music from different artists, with the RNN looking to learn the patterns of music in the datasets to generate new music. However, this music must be carefully checked and assessed to ensure it is original, with the RNNs potentially having recurrent or looped connections that enable it to learn sequence data such as music, but also may cause it to repeat. Such evidence shows the ongoing challenges of music composition, in the form of other elements which must be considered outside the core issues of chords and lyrics. Finally, the literature also notes that music generation, as a creativity problem, continues to attract attention from artificial intelligence researchers and remains a subject worthy of consistent and detailed study. In this regard, the concept of lyrics-conditional melody generation has been highlighted as a way to leverage NLP and GAN to generate music from texts, as a promising unsupervised solution. In this regard, once again, the adversarial training of two agents, in the form of the generator and discriminator, is shown to allow GANs to achieve a better generation performance, particularly when applied in the form of CGANs (Sun et al, 2022). However, this case is focused on the use of lyrics to generate music and thus leaves open the issue of whether and how a machine learning tool could be used to create and compose lyrics from a data set of music and other lyrics.

## **2.4 Research gap**

This literature review has critically evaluated the extant literature in the research area. In general, GANs, CGANs and NLP have all been highlighted as strong tools that can be used in the development of generative AI systems. They have also all been applied, to some degree, within the music industry as creativity and compositional tools. However, these tools have been primarily applied to the generation of music, in the form of chords and melodies, as well as to other elements such as choreography and creative aspects. As such, they have not been applied in any significant detail or with any degree or rigour to the creation, composition and generation of lyrics. This leaves a clear gap in the literature which the present research aims to address through the development of Creation Unblock as a superior machine learning creativity and compositional tool used to assist in the searching, identification, classification and generation of lyrics for the music industry.

**Chapter 3: Research**

## **3.1 Types of research**

Research methods generally fall into two categories, namely primary research and secondary research. Of the primary research is information gathered through self-conducted research methods, and thus represents data which has not previously been collected or processed. Examples of self-conducted research include surveys, interviews, focus groups and direct observations. In contrast, secondary research is data and information gathered from previously conducted studies or activities and often processed and published by the authors of said studies or activities. Examples of secondary research thus include published research articles, white papers, media and published data sets. Some secondary data may be public, whilst other data may be private, such as from the internal documents or files of businesses, and thus this may require consent in its use and care regarding its disclosure (Robson and McCartan, 2015).

In general, primary research is primarily required for research issues which have not been specifically investigated to any great degree in the past, and hence issues for which there is relatively limited relevant secondary data available. In such a case, the research can only achieve true insight through the collection and analysis of primary research data, which is tailored and relevant to the specific needs of the present research. However, to collect primary data, there is often a significant burden placed on the research, as said research must be carried out carefully and deliberately which ensures that the research does not become skewed or biased. In contrast, secondary data is often used to analyse and report on existing issues from a new angle or perspective. As such, this form of research can make use of the secondary data which is already available on a given topic, such as from reports, analysis or press comments, enabling additional insights into the theoretical and empirical situation around a given research topic (Saunders et al, 2019). In the present context, there is significant secondary data available regarding music composition, including various music databases and also the different Python and TensorFlow datasets which can be used to train the music system. As such, this provides a strong set of options and indicates that secondary research will be the main method applied by this project. However, there are also important considerations regarding the specific issues of song lyrics and the response of individuals to the created algorithm, and thus some level of primary research in the form of user feedback and reviews will also be required.

The other main typology of research is regarding quantitative data and qualitative data. In this regard, quantitative research is numerical data which can be expressed in numbers and displayed using graphs and charts. This form of data can be used to test or confirm theories and assumptions, as well as hypotheses. As such, this type of data is often used in deductive research, to test hypotheses and in turn to establish generalisable facts about a topic or to test models and theories to explain said topic. In contrast, qualitative research is generally all data which cannot be expressed numerically and instead is usually expressed in words. It is used to understand concepts, thoughts, or experiences, and in turn, enables researchers to gather insight into topics that are not well understood. As such, one of the most important factors that need to be considered when constructing a data collection instrument and designing the wider research instruments is whether the research should primarily use qualitative data or quantitative data.

Of these choices, the qualitative data collection strategy is generally seen as positive in cases where it is important to consider the words and meanings within the data, rather than the quantification of facts and figures. This in turn enables research to focus on the specific meanings and insights from the data collected, rather than simply looking to quantify complex research issues. This hence leads to the argument that qualitative data facilitates a richer understanding of research topics, particularly those characterised by the diverse views and beliefs of respondents, whilst also enabling insight into various contextual factors and motivations which can be influencing their responses. However, this increased richness can come at the expense of insights and objectivity. In particular, quantitative research enables a researcher to assess a great volume of data and develop clear connections between data points and insights, without considering wider issues around meaning and interpretation. This in turn makes quantitative data a more effective choice than a qualitative strategy when the research aims to answer specific questions with definitive and objective answers or to develop clear theories and frameworks. Quantitative research is thus most useful when looking to analyse broad and general data which needs to be processed and analysed numerically. This makes quantitative research most valuable when looking to develop and test algorithms and systems to create strong insights and understanding of their effectiveness (Holton and Walsh, 2017). As such, the present research looks to use quantitative data where possible.

## **3.2 Research Plan**

This project will primarily require secondary research since the domain is already well-understood and broadly researched. The secondary research content for the project will thus include the following core elements:

* Development Methodologies
* Domains
  + Machine Learning
  + GANs
  + CGANs
  + NLP
  + TensorFlow
  + Datasets for CGAN Training
* Testing Strategies
* Analysis of Research
  + Analysis
  + Requirements
  + Choice of language
  + Choice of Development Environment

## **3.3 Development Methodologies**

The following section discusses a variety of popular software development methodologies which can be applied to manage the software development life cycle of a system. This section considers each methodology, provides a comparison of each methodology, details the methodology chosen for the project, and offers an outline of how the chosen methodology will be implemented. In general, it is important to select a suitable methodology and to adhere to it throughout the project, to ensure a greater chance of project success (Kumar, 2019). In this particular case, an appropriate software development methodology can be used to ensure a stronger level of delivery performance, keep a project on time, prioritise tasks, provide better estimates and deliver a more stable final system.

The first type of development methodology is the waterfall methodology. This methodology involves breaking down each stage of development into phases, where the deliverables of each phase are used as the input for each successive phase. The development process thus does not have an iterative cycle, and instead requires all tasks for each phase to be completed before moving onto the successive phase. In contrast, the Spiral methodology is an iterative software development methodology looking broadly to follow the phases of the Waterfall methodology but adapted to manage the software development lifecycle of large government projects. As such, the Spiral methodology involves a spiral process of four steps, involving identifying objectives; performing risk analysis; development and testing; and finally reviewing and planning for the next phase of the spiral. This hence follows the Waterfall methodology in successive phases, but with each phase having its own specific set of steps which must be completed before the next phase is engaged.

The next option for a methodology is the Agile Scrum methodology. This is a software development model which is designed to develop, deliver, and sustain complex projects within a team. The name of the methodology is often referred to as ‘Agile Scrum’, reflecting the rugby scrum as a team pack, where everyone in the team acts together and provides all their effort towards a shared goal, pushing together to deliver a project within time and with minimal cost (Mahalakshmi and Sundararajan, 2013). Under the Agile Scrum concept, ‘User Stories’ are used to describe functionality and outcomes, and thus as a core point of focus. Whilst these do not replace project requirements, they do point to critical tasks that are necessary to complete requirements and also outline key design documents and requirement documents. These User Stories are stored and placed on the ‘Sprint Board’ when they are planned to be completed in the current sprint, thus offering a highly focused methodology aiming to address current goals. Finally, the Extreme Programming methodology is an agile software development framework that aims to produce high-quality software where requirements may be subject to change. This is hence intended to be used to develop software in the face of vague or rapidly changing requirements, such as those which may be faced by a small project in a turbulent environment (Denscombe, 2010).

In light of these options, the present project will be managed through a systematic approach, employing Agile Scrum methodologies to ensure adaptability and efficiency throughout the development lifecycle. Regular sprints and feedback will be integral to the project management strategy, fostering a dynamic and responsive development environment. This will support the use of a modified Agile Scrum methodology due to its time-boxing, task prioritisation methods, low documentation, and ability to adapt to changing or additional requirements. This methodology will thus enable the major concepts of the Agile Scrum methodology, within the scope of a project with just a single developer, and thus optimise expected delivery outcomes.

## **3.4 Domains**

### **3.4.1 Machine Learning**

Machine learning is an increasingly common concept in modern everyday life. Its presence can be found in various digital forms, such as social media recommendation algorithms and digital ‘chatbots’ used to substitute for human customer service advisors. In most cases, the systems are designed to respond to the specific needs and requirements of each user, but without being explicitly programmed for each user. This is made possible using machine learning algorithms that give computers the ability to learn and display a form of specific artificial intelligence in their given context, without being explicitly programmed with responses and inputs (Parvat and Dev, 2017). In this regard, there are three main methods of machine learning, namely supervised, unsupervised and reinforcement learning. In the present context, the project aims to develop a system which applies machine learning to the creation of song lyrics. This indicates that unsupervised learning methods are likely to be the best-suited learning methods for the development of the specific tools in this project, given the need for the algorithm to function independently without supervision when used by producers and other music industry individuals.

### **3.4.2 GANs and CGANs**

GANs can be conceived as a form of generative modelling which uses deep learning methods to discover and learn the various regularities and patterns in input data, enabling the model to generate new forms of output and examples that could plausibly have been a part of the original dataset. In this regard, GANs train the generative model by framing the problem as a form of internal supervised learning applying two sub-models. These consist of the generator model which is trained to generate new examples, and the discriminator model which is trained to classify examples as being either real data or fake generated data. These two models are trained together in an unsupervised manner, repeatedly generating and classifying examples, until the generator can fool the discriminator around half the time, thus showing the generation of plausible examples as the discriminator is effectively ‘guessing’ at whether examples are real or fake. The conditional GAN, or CGAN, represents an advancement of this process, enabling the user to condition the network with additional information, particularly class labels. As such, during the training, this involves providing the network with labelled data to learn the difference between classifications, and in turn to generate data for individual classes. In the present case, the CGAN will be used to develop lyrical compositions that can be applied to particular genres of music, thus offering a general lyric generation system.

### **3.4.3 NLP**

NLP refers to the ability of artificial intelligence programs to understand natural human language, specifically language in the form that it is spoken and written by real humans. Whilst NLP has existed for several decades now, and has been applied in various fields such as medical research, business intelligence and search engines, it is often limited in its ability to understand natural language with heavily unstructured roots, such as dialects. This in turn creates challenges for the use of NLP in music, where sentence structures are often irregular and linguistic laws and rules may be bent or broken. As such, it will be important to develop and condition NLP within the CGAN model to enable it to identify, classify and generate lyrical compositions which are sufficiently natural and real as to be of use to music producers and creators.

### **3.4.4 TensorFlow**

TensorFlow, and the updated TensorFlow 2, is an open-source deep learning software library that can be used to develop, define, train and deploy new forms of machine learning models (Babcock and Balie, 2021). This library supports the training and creation of new forms of generative AI using Python coding to create images, text, and music. TensorFlow has thus become established as a key driver of modern deep learning concepts and software, with the potential to support NLP and to expand the capabilities of various AI and machine learning models and algorithms (Goldsborough 2016). The key properties of TensorFlow in this regard is the ability of the machine learning system to represent computation, shared state, and the operations that mutate that state through appropriate dataflow graphs, providing high levels of flexibility to the application developer (Abadi et al, 2016). This in turn provides significant support for training and inference developments on deep neural networks, and the integration of NLP into this context. As such, TensorFlow will be key to the integration of NLP into the CGAN model used to develop lyrical compositions and support creativity.

### **3.4.5 Datasets for CGAN Training**

There are several databases of music lyrics available for training and research purposes. Of these, the following will be used to test the CGAN model:

* MLDb
* Open Lyrics Database
* Lyrics.com
* MetroLyrics.com

## **3.5 Testing Strategies**

The core testing strategies will involve first training the CGAN model with the data from the identified datasets. Following this, the CGAN will be allowed to engage in unsupervised learning, generating various conditional lyric forms to be discriminated and refined over time. Multiple CGAN algorithms will be tested and assessed based on the following criteria:

* Accuracy in generating music lyrics that are stylistically coherent and artistically relevant across various genres
* User satisfaction ratings based on user feedback and surveys

## **3.6 Analysis of Research**

### **3.6.1 Analysis**

A modified Agile Scrum methodology was developed in this chapter, involving a high focus on agility and the strong use of regular sprints as part of the methodology. This differs from classical Agile Scrum as it does not require the ‘scrum’ of collaboration between members of the development team, but instead focuses on a single specific developer. At the same time, the majority of the Agile method was retained. The research process involves the generation, training and testing of CGAN models to provide sufficient insights regarding the performance of these models and their abilities in terms of the generation of music lyrics that are stylistically coherent and artistically relevant across various genres, and also achieve high user satisfaction ratings.

### **3.6.2 Choice of language**

The language chosen to develop the application for this project is Python. Python has been chosen due to the high level of development of systems and databases which already use this language. In particular, TensorFlow 2 is designed specifically to support the creation of new forms of generative AI using Python coding to create images, text, and music. Further to this, Dong et al (2020) developed a proprietary MusPy database as an open-source Python library for symbolic music generation. This highlights the value of Python as a language for coding the CGAN and NLP systems that will be at the heart of Creation Unblock. Python is also recognised as one of the most commonly used and widely recognised coding languages, thus meaning it provides higher levels of access to various tutorials and sources of support when developing, coding and adapting the tool to enhance its performance (Stack Overflow, 2022).

### **3.6.3 Choice of Development Environment**

The chosen development environment for this project is PyCharm. PyCharm has been selected as it represents an effective integrated development environment, created specifically for programming in the Python programming language. The value of the environment is reflected in the fact that PyCharm is widely used by people learning to code, showing its performance as a learning tool (Stack Overflow, 2022). PyCharm also has useful features for Python development including coding assistance, project and code navigation, code refactoring, an integrated Python debugger and integrated unit testing. In addition to this, Streamlit will be used to create the UI for CreationUnblock, as well as the Gemini API. Streamlit is a framework for building interactive web applications with Python and Gemini is for interacting with Google's AI capabilities. They’re both simple and easy to understand, integrate well with machine-learning libraries and are deployable on many different platforms.

## **Chapter 4: Design**

For the suggested system, this chapter will include basic designs and product needs.

**4.1 Requirements**

The MoSCoW approach will be used in this project to rank requirements. MoSCoW is a method of prioritisation that guarantees essential requirements are addressed first and ensures that the project’s needs are met efficiently within the time constraints.

Must Haves:

* Ability to detect BPM, Tempo and Genre of a song
* Ability to generate authentic song lyrics based on genre
* Have a user-friendly interface

Should haves:

* A diverse dataset so a wide variety of lyrics can be available for use, whether that be for song generation or song identification.
* Shazam like API integration so users can play songs and detect the BPM, Tempo and Genre
* Rigorous testing of the model and features based on user feedback.

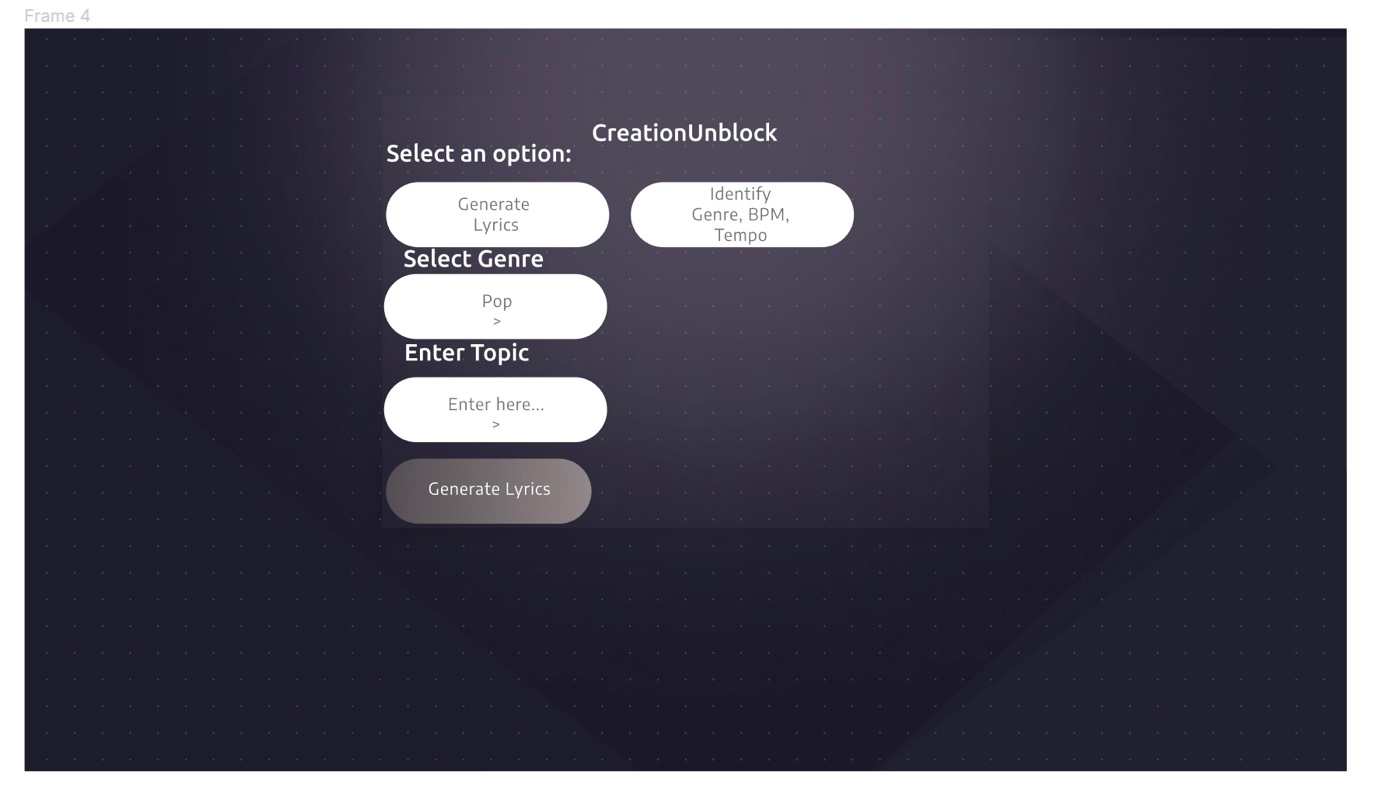
Could haves:

* Additional features such as lyric editing tools, collaborative composition features, or lyric sentiment analysis.
* Integration of other AI techniques or algorithms to enhance lyric generation or recognition capabilities.

Won’t Have:

* Integration with advanced music production software suites or hardware devices.

**4.2 Graphical User interface Design**

 Figure 1 GUI Design

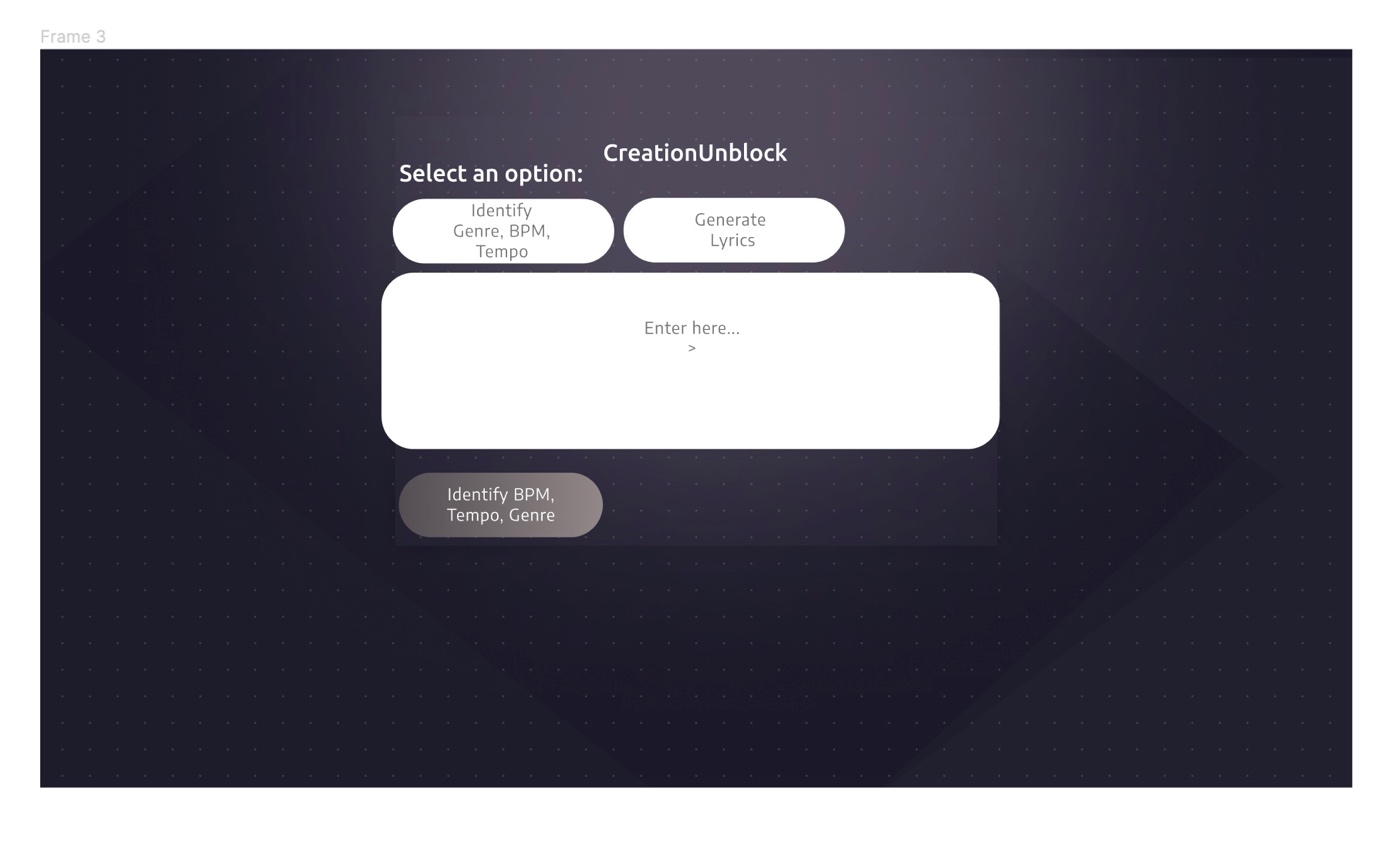


Figure 2 GUI Design.

## **Chapter 5: Development**

The project's development phase is covered in this chapter.

**5.1 Sprint 1**

In this sprint, the environment will be developed by installing the necessary libraries required for the project. This includes installing Streamlit, Langchain, Langchain Google GenAI, and Google Generative AI libraries. Once the libraries are installed, I’ll ensure that the environment is configured properly for code execution.

1. Installation of Required Libraries:

- Install Streamlit version 1.31.1: `pip install streamlit==1.31.1`

- Install Langchain version 0.1.13: `pip install langchain==0.1.13`

- Install Google Generative AI version 0.4.1: `pip install google-generativeai==0.4.1`

- Install Langchain Google GenAI library: `pip install langchain\_google\_genai`

With the environment set up, moving forward with coding and implementing the core functionality of the application is possible.

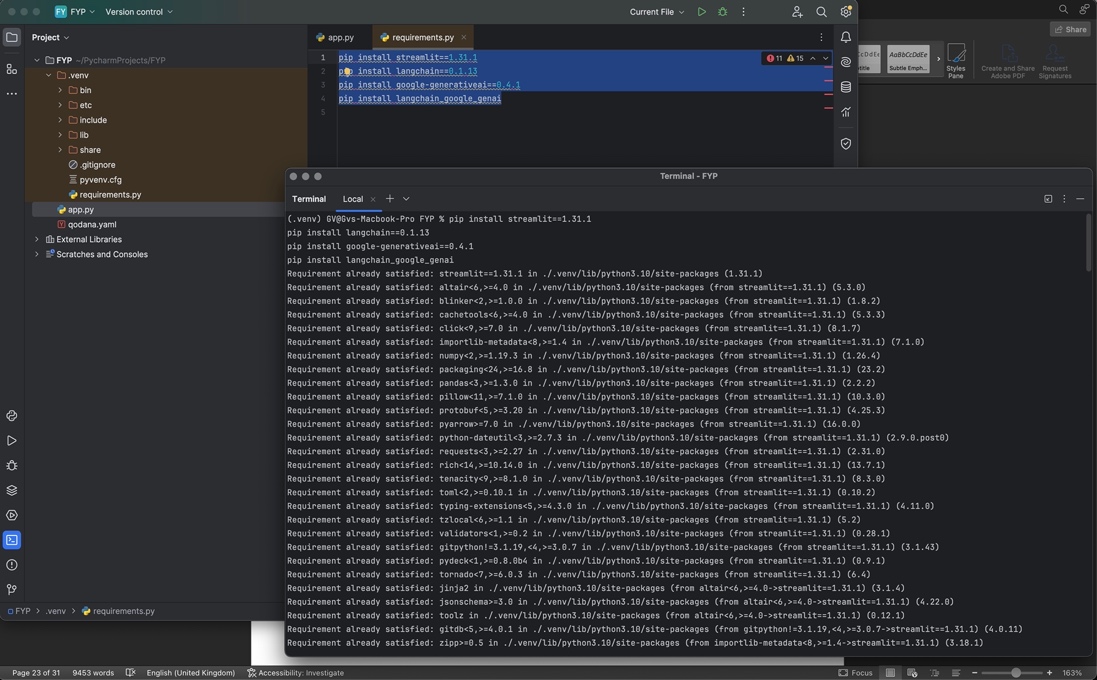


Figure 3 Library Installation

**5.2 Sprint 2**

During this sprint, the focus will be on writing the code to implement the core functionality of the application. Functions will be defined for identifying genre, BPM, and tempo, as well as for generating lyrics. Additionally, I’ll design the Streamlit user interface to provide a seamless experience for users. The code will handle user interactions, such as selecting options and inputting data, and display the output accordingly.

1. Library Imports:

- Import required libraries including Streamlit, Langchain, Langchain Google GenAI, and Google Generative AI.

2. Function Definitions:

- Define functions `identify\_genre\_bpm\_tempo` and `generate\_lyrics` for identifying genre, BPM, and tempo, and generating lyrics respectively.

3. Streamlit UI Definition:

- Define the Streamlit UI with radio button options for selecting actions and input fields for lyrics and genre/topic.

4. Option Handling:

- Based on the selected option, either identify genre/BPM/tempo or generate lyrics.

5. Function Calls:

- Call the appropriate function (`identify\_genre\_bpm\_tempo` or `generate\_lyrics`) based on the user's selection.

6. Display Output:

- Display the predicted genre, BPM, and tempo, or the generated lyrics on the Streamlit UI.

With the code execution completed, proceeding to the testing phase to ensure that the application functions as expected is possible.

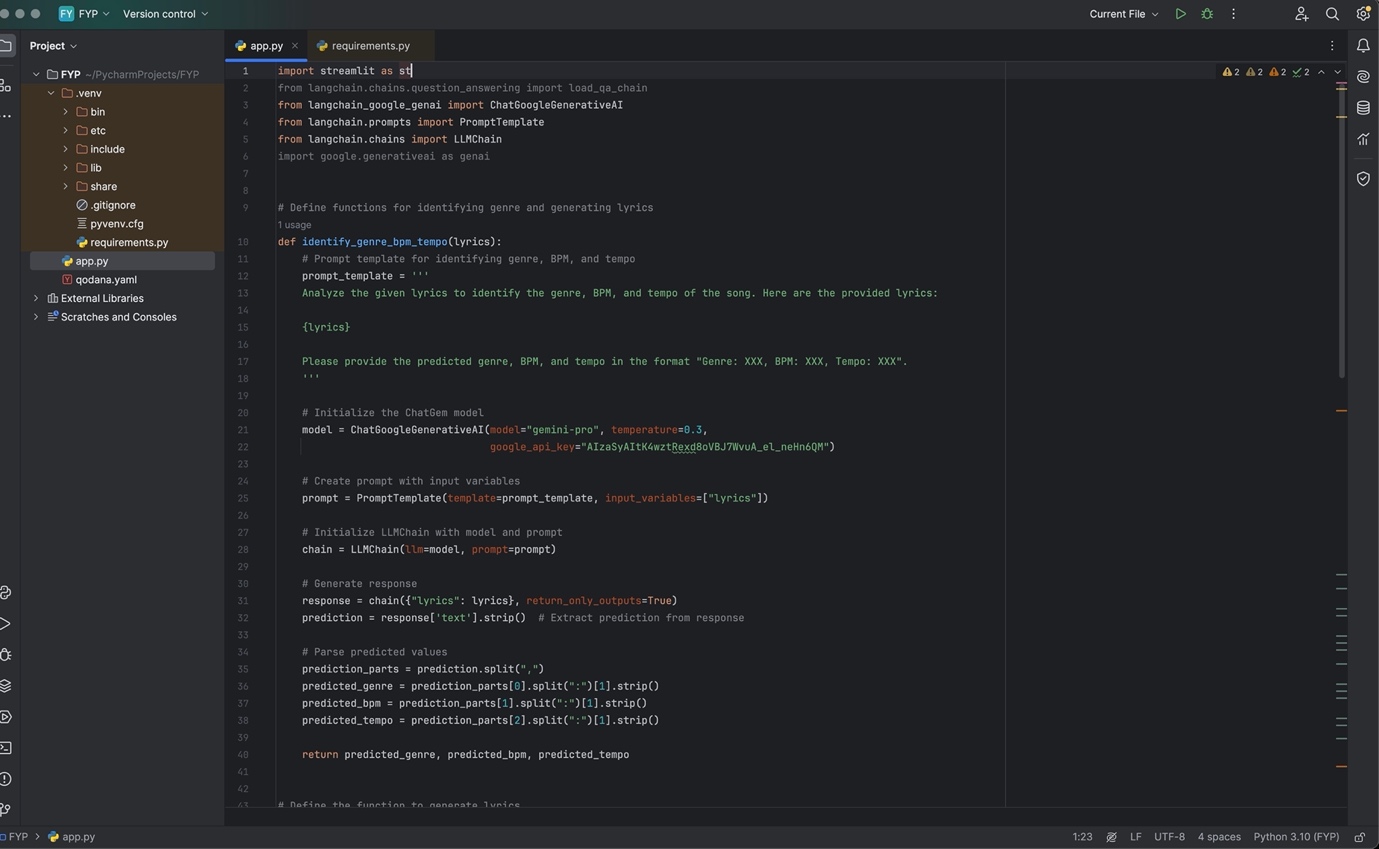


Figure 4 Code

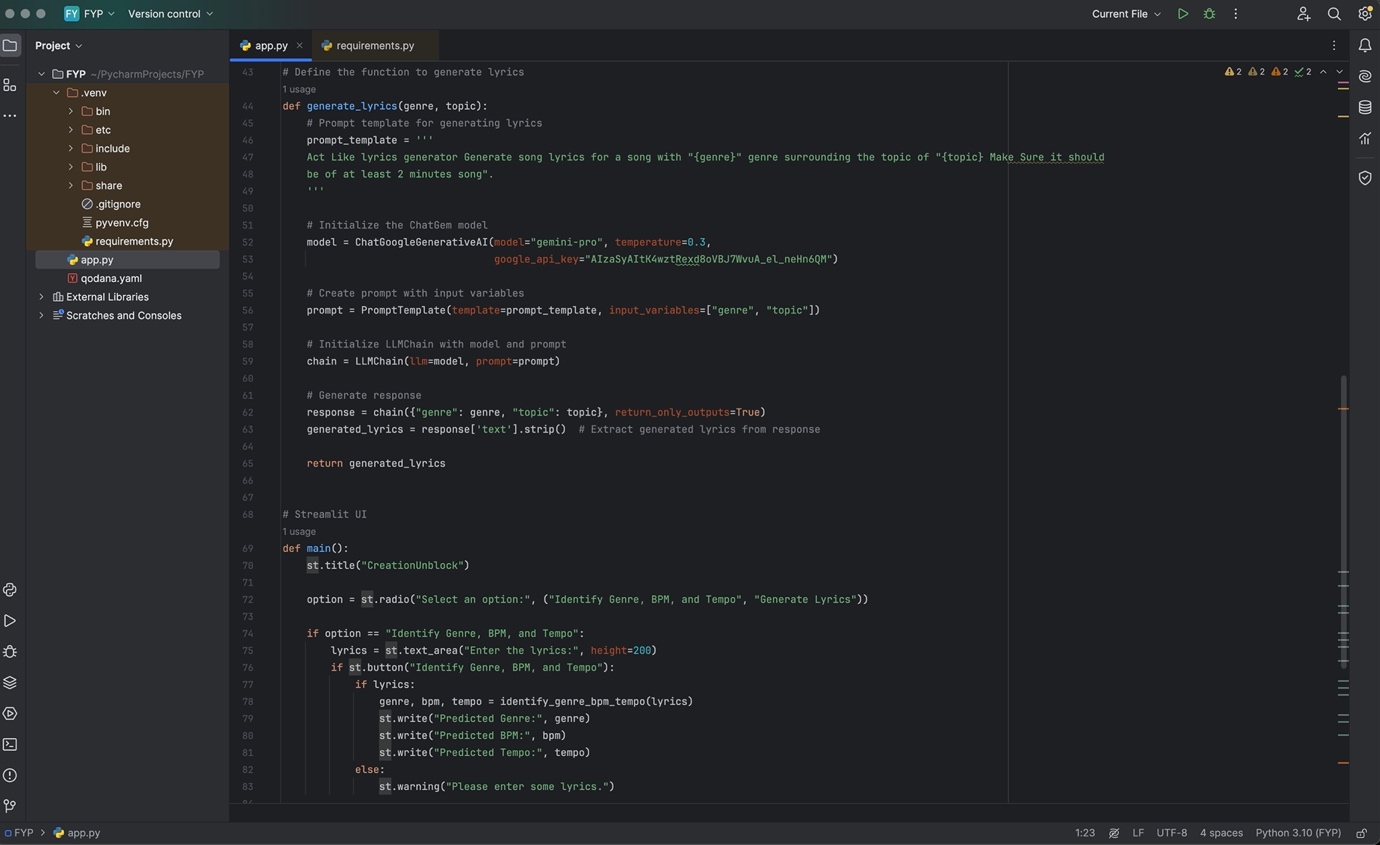


Figure 5 Code

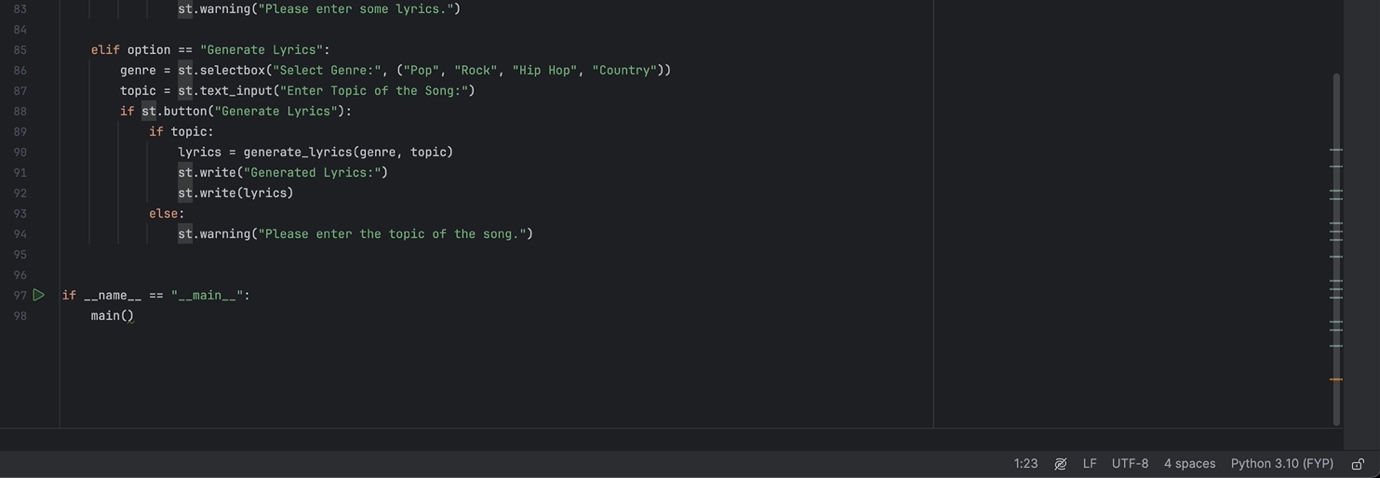


Figure 6 Code

**5.3 Sprint 3**

In this sprint, I’ll thoroughly test the application to ensure that it functions as expected. I’ll input test data, interact with the user interface, and validate the output against expected results. Any errors or unexpected behaviours encountered during testing will be addressed through debugging and code adjustments. I will also gather feedback from users to identify areas for improvement and iterate on the code accordingly.

1. Run the Application:

- Execute the Python script containing the code.

2. Test Functionality:

- Test the functionality of the application by inputting lyrics, selecting options, and observing the generated output.

3. Debugging:

- Address any errors or unexpected behaviours encountered during testing.

4. User Feedback:

- Gather feedback regarding the usability and effectiveness of the application.

5. Iterate Based on Feedback:

- Make necessary adjustments or enhancements to the code based on user feedback to improve the application's performance and user experience.

With testing and iteration completed, moving forward with documentation and deployment preparations is possible.

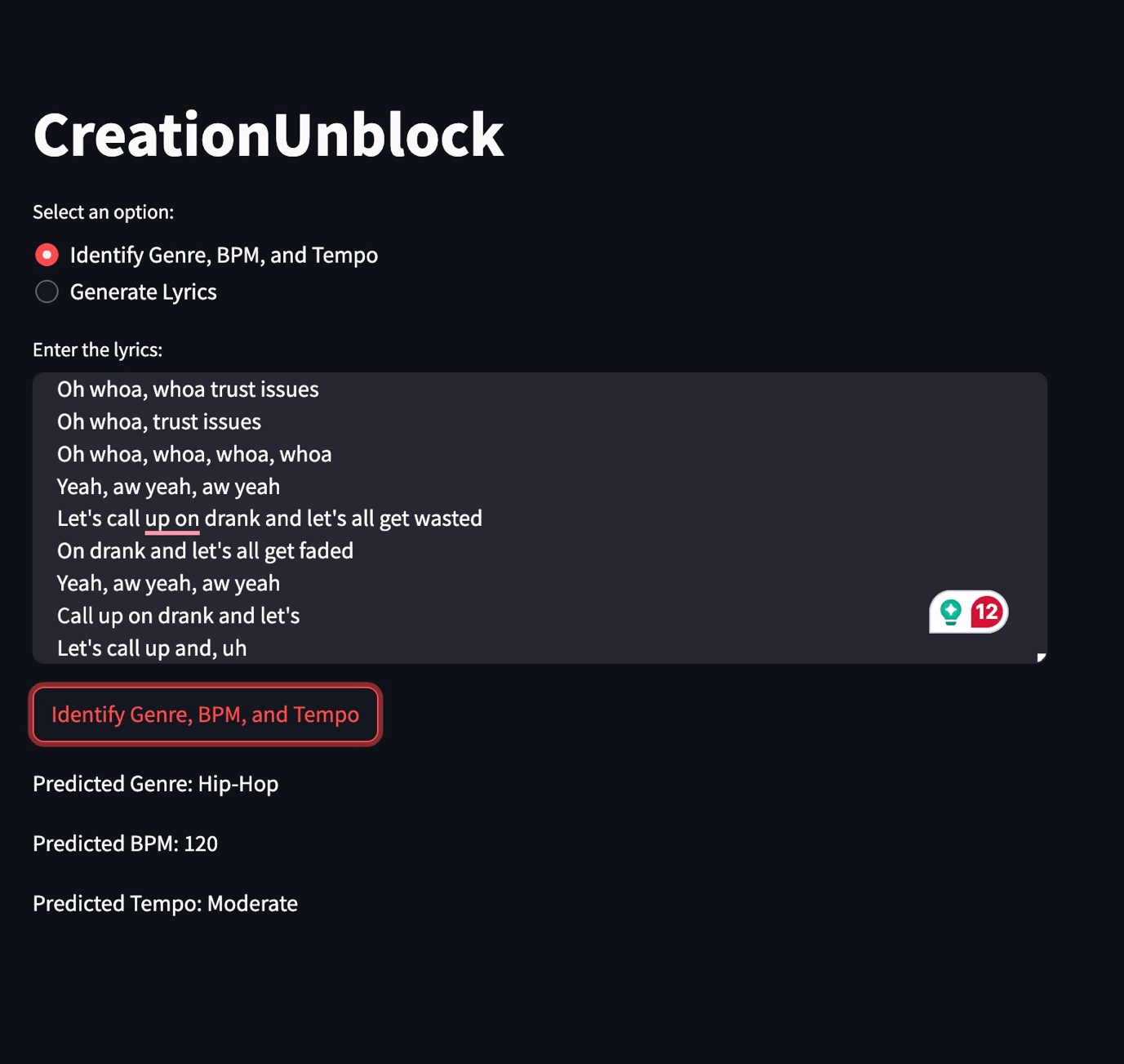


Figure 7 Deployment

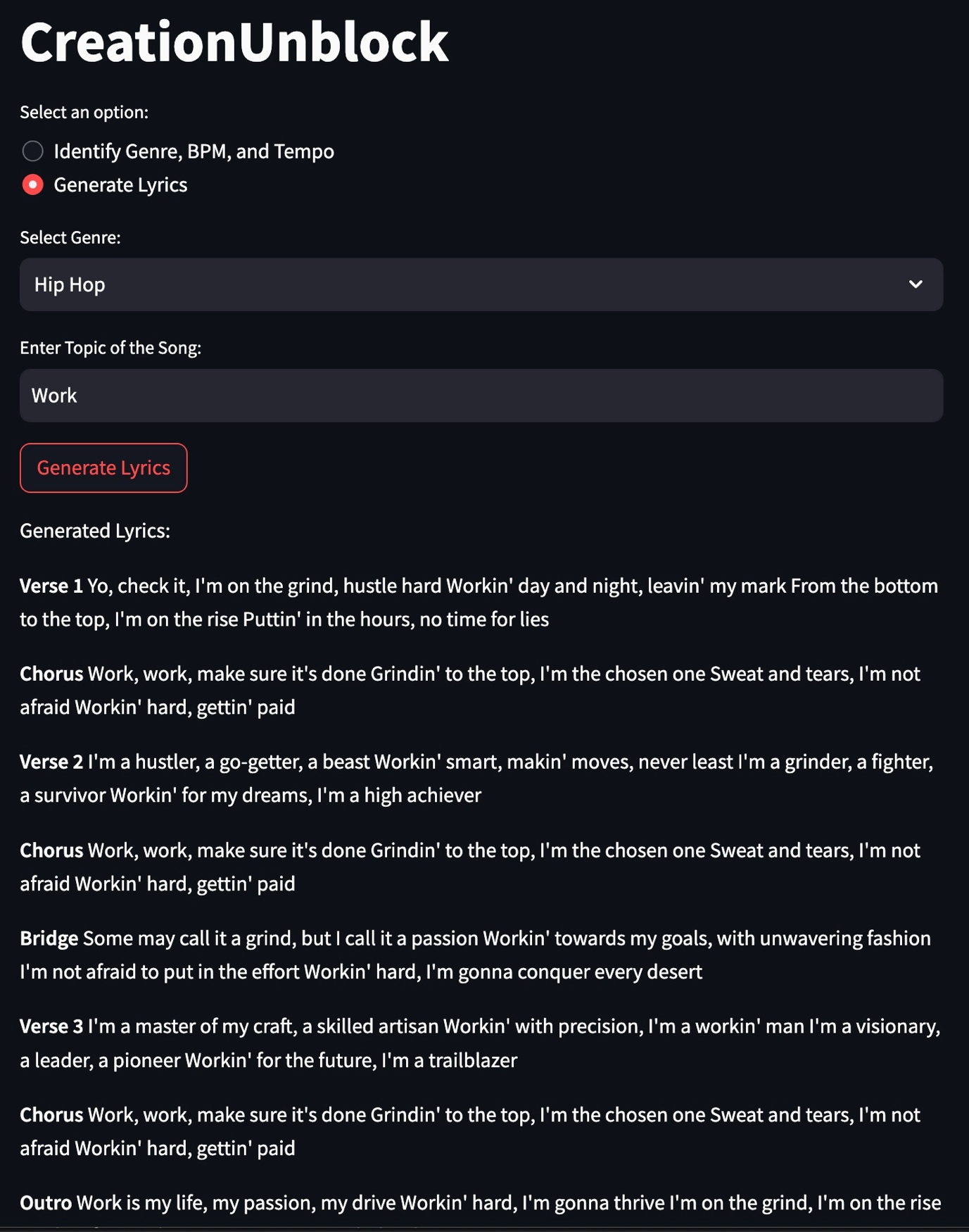


Figure 8 Deployment

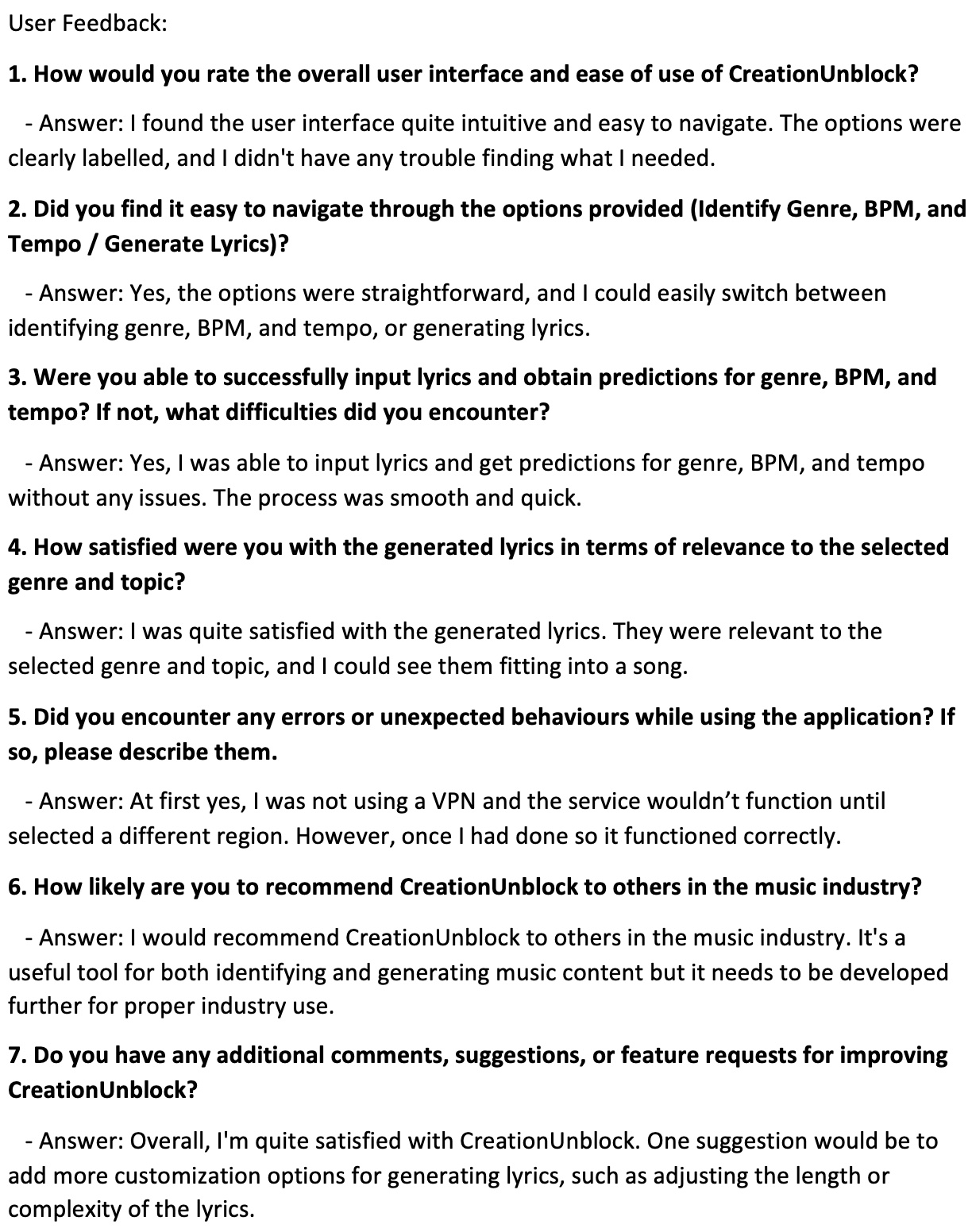


Figure 9 User Feedback

**5.4 Sprint 4**

During the final sprint, I’ll focus on documentation and deployment preparations. Additionally, I’ll prepare the application for deployment to a production environment, ensuring that all dependencies are correctly installed and configured.

1. Documentation:

- Document the installation process, code structure, and functionality for future reference.

2. Deployment Preparation:

- Prepare the application for deployment to a production environment, ensuring all dependencies are correctly installed and configured.

With documentation and deployment preparations completed, the project is now ready for deployment.

**5.5 Implementation Review**

The Appendices include a video that shows the entire implementation in action. (Appendix B) This checks every GUI feature. The MoSCoW criteria' Must was fulfilled in the final application.

Every feature of the GUI functions flawlessly. The application presents a text field for text entry and a functional button to initiate the command when you access the identify feature. When using the application's lyric generation feature, dropdown menus are displayed appropriately, along with a text field where users can enter a topic and a button that initiates the command. Not long after the lyrics are accurately generated.

## **Chapter 6: Conclusion and Future Works**

This chapter will provide a summary of the project, covering project management, upcoming tasks, and potential improvements to the project's state if it were to be replicated.

**6.1 Project Management**

The project was managed using Gantt charts, which are located in the Appendices.

Using Gantt charts was highly beneficial in managing this project as the charts offer a visual representation of tasks and timelines, providing clarity on the sequence of activities and the overall project schedule. Gantt charts ensure that activities are executed in the correct order, facilitating efficient resource allocation and minimizing conflicts. Moreover, they allowed for real-time progress tracking, allowing easy identification of delays early on and taking corrective actions.

Sprints were also helpful to the project as they provided an organised, iterative method of development that allowed for regular input and adjustment. Sprints allowed for a concentration on delivering incremental value within time constraints by segmenting the project into smaller, more manageable work packages. This cycle of iteration allows for ongoing enhancement and guarantees that the project stays adaptable to evolving needs derived from feedback.

**6.2 Future Works**

For future iterations of this project, several areas could be explored to enhance its functionality, usability, and impact. Additional features could be implemented to further enrich the user experience and cater to diverse user needs. For example, integrating more advanced AI capabilities, such as sentiment analysis or personalized recommendations, could offer users deeper insights and customization options. Moreover, expanding the dataset used for training the machine learning models could improve the accuracy and diversity of generated content, ensuring better representation across various music genres and styles. Additionally, refining the user interface design based on user feedback and usability testing could streamline workflows and make the application more intuitive to use.

**6.3 Repeating of the Project**

It would be advantageous to do a more in-depth first planning stage. This could aid in defining more precise project objectives and a more organised implementation schedule. To guarantee alignment goals. More focus might also be placed on early user interaction and feedback. Regular user testing sessions would be built into the development calendar. Lastly, putting in place reliable systems for tracking and assessing project development, like frequent progress reviews, may aid in spotting possible problems early on and allow for prompt modifications to project priorities and objectives.

# **Running The Application**

To run the application:

1. Enable a VPN in one of the regions working with the Gemini API (United States)

2. Ensure you have installed the required libraries by running the following commands:

pip install streamlit==1.31.1

pip install langchain==0.1.13

pip install google-generativeai==0.4.1

pip install langchain\_google\_genai

3. Once the libraries are installed, execute the code provided in your preferred Python environment.

4. After executing the script, Streamlit will launch a local web server and open the application in your default web browser automatically.

**Appendices**

Appendix A – Gantt Chart = Each iteration of the chart as the project progressed

Appendix B – Video of a working implementation

# **Bibliography**

1. Adhikari, T. (2023) MeloHarmony: Exploring Emotion in Crafting AI-Generated Music with Generative Adversarial Network Powered Harmony. *SSRN Electronic Journal*. doi.org/10.2139/ssrn.4575257
2. Aditya, H. Dev, J. Das, S. and Yadav, A. (2023) Music generation using RNNs and LSTMs. In *Artificial Intelligence, Blockchain, Computing and Security*. 1st Edition. New York: CRC Press
3. Ali Elfa, M. and Dawood, M. (2023) Using Artificial Intelligence for Enhancing Human Creativity. *Journal of Art, Design and Music*. 2(2) doi.org/10.55554/2785-9649.1017
4. Azahad, S. and Hameeda, S. (2023) A Deep Exposition of GAN and its applications. *International Journal of Engineering Technology and Management Sciences*. 7(2) doi.org/10.46647/ijetms.2023.v07i02.005
5. Bastianello, E. (2023) *Sound Generation using GAN Models*. Padua: University of Padua
6. Bell, J. (2014) *Doing Your Research Project: A Guide for First-Time Researchers*. London: McGraw-Hill Education
7. Chakraborty, T. Ujjwal Reddy, K. Naik, S. Panja, M. and Manvitha, M. (2024) Ten years of generative adversarial nets (GANs): a survey of the state-of-the-art. *Machine Learning: Science and Technology*. 5(1) e011001 DOI:10.1088/2632-2153/ad1f77
8. Collins, D. (2005) A synthesis process model of creative thinking in music composition. *Psychology of Music*. 33(2) p193-216 doi.org/10.1177/0305735605050651
9. Dash, A. Ye, J. and Wang, G. (2024) A Review of Generative Adversarial Networks (GANs) and Its Applications in a Wide Variety of Disciplines: From Medical to Remote Sensing. *IEEE Access*. 12, p18330-18357 doi:10.1109/ACCESS.2023.3346273
10. Data Scientist (2023) What is a Conditional Generative Adversarial Network (cGAN)? https://datascientest.com/en/what-is-a-conditional-generative-adversarial-network-cgan Accessed 24th November 2023
11. Deepaisarn, S. Chokphantavee, S. and Chokphantavee, S. (2023) NLP-based music processing for composer classification. *Sci Rep*. 13, e13228 doi.org/10.1038/s41598-023-40332-0
12. Denscombe, M. (2010) The good research guide: for small-scale social research projects. 4th Edition. Maidenhead: McGraw-Hill
13. Ding, X. Wang, Y. Wang, J. and Welch, W. (2023) Efficient subsampling of realistic images from GANs conditional on a class or a continuous variable. *Neurocomputing*. 517, p188-200 doi.org/10.1016/j.neucom.2022.10.070
14. Dong, H. Hsiao, W. Yang, L. and Yang, Y. (2018) MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. *Proceedings of the AAAI Conference on Artificial Intelligence*. 32(1) doi.org/10.1609/aaai.v32i1.11312
15. Ferreira, P. Limongi, R. and Favero, L. (2023) Generating Music with Data: Application of Deep Learning Models for Symbolic Music Composition. *Appl. Sci*. 13(7) p4543 doi.org/10.3390/app13074543
16. Guo, Y. Liu, Y. Zhou, T. Xu, L. and Zhang, Q. (2023) An automatic music generation and evaluation method based on transfer learning. *Plos ONE*. https://doi.org/10.1371/journal.pone.0283103
17. Hargreaves, D. Miell, D. and MacDonald, R. (2012) *Perception*. Oxford: OUP
18. Holton, J. and Walsh, I. (2017) Classic grounded theory: Applications with Qualitative and Quantitative Data. Thousand Oaks, CA: Sage
19. Imasato, N. Miyazawa, K. Duncan, C. and Nagai, T. (2023) Using a Language Model to Generate Music in Its Symbolic Domain While Controlling Its Perceived Emotion. *IEEE Access*. 11, p52412-52428 doi:10.1109/ACCESS.2023.3280603
20. Jeong, J. Tariq, A. and Adejumo, T. (2022) Systematic Review of Generative Adversarial Networks (GANs) for Medical Image Classification and Segmentation. *J Digit Imaging*. 35, p137-152 doi.org/10.1007/s10278-021-00556-w
21. Jovanovic, J. (2023) How does Shazam work? Music Recognition Algorithms, Fingerprinting, and Processing. https://www.toptal.com/algorithms/shazam-it-music-processing-fingerprinting-and-recognition#:~:text=Analog%20to%20Digital%20%2D%20Sampling%20a,in%20various%20forms%20back%20then. Accessed 25th November 2023
22. Kamberaj, V. Kadriu, A. and Besimi, N. (2023) Instruments Music Composition in Different Genres and Techniques Using AI: A Review. In: Bexheti, A. Abazi-Alili, H. Dana, LP. Ramadani, V. and Caputo, A. (Eds) *Economic Recovery, Consolidation, and Sustainable Growth*. Cham: Springer doi.org/10.1007/978-3-031-42511-0\_48
23. Kang, M. Zhu, J. Zhang, R. Park, J. Shechtman, E. Paris, S. and Park, T. (2023) Scaling Up GANs for Text-to-Image Synthesis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, p10124-10134
24. Kladder, J. (2016) Maschine-activity: How I found my creativity using a digital sampling device to compose music. *Journal of Music, Technology & Education*. 9(3) p289-313doi.org/10.1386/jmte.9.3.289\_1
25. Kulkarni, A. Mahajan, P. Nimbokar, G. and Raut, N. (2023) AI Based Song Recommendations System. *Conference: IDES and Association of Computer Electrical Electronics and Communication Engineers (ACEECom)*. June 2023
26. Kumar, R., (2019) Research methodology: A step-by-step guide for beginners. London: Sage Publications Limited
27. Lee, M. Lew, H. Youn, S. Kim, T. and Hwang, J. (2022) Deep Learning-Based Framework for Fast and Accurate Acoustic Hologram Generation. *IEEE Trans Ultrason Ferroelectr Freq Control*. 69(12) p3353-3366 doi:10.1109/TUFFC.2022.3219401
28. Min, J. Liu, Z. Wang, L. Li, D. Zhang, M. and Huang, Y. (2022) Music Generation System for Adversarial Training Based on Deep Learning. *Processes*. 10(12) p2515 doi.org/10.3390/pr10122515
29. Peppers, N. Tsakanikas, P. Daskalakis, E. Alexakis, T. Adamopoulou, E. and Demestichas, K. (2023) FoGGAN: Generating Realistic Parkinson’s Disease Freezing of Gait Data Using GANs. *Sensors*. 23(19) p8158 doi.org/10.3390/s23198158
30. Robson, C. and McCartan, K. (2015) Real World Research. 4th Edition. London: Wiley
31. Saunders, M., Lewis, P., & Thornhill, A. (2019) *Research Methods for Business Students*. 8th Edition. Harlow: Pearson Education Limited
32. Shahbazi, M. Danelljan, M. Paudel, D. and Van Gool, L. (2022) Collapse by Conditioning: Training Class-conditional GANs with Limited Data. *Computer Vision and Pattern Recognition*. doi.org/10.48550/arXiv.2201.06578
33. Stack Overflow (2022) 2022 Developer Survey. https://survey.stackoverflow.co/2022/#most-popular-technologies-new-collab-tools-learn Accessed 1st February 2024
34. Sun, F. Tao, Q. Yan, J. Hu, J. and Yang, Z. (2022) MRGAN: Multi-Criteria Relational GAN for Lyrics-Conditional Melody Generation. *2022 International Joint Conference on Neural Networks (IJCNN)*. 18-23 July 2022. DOI:10.1109/IJCNN55064.2022.9892702
35. Sun, H. Deng, Z. Chen, H. and Parkes, D. (2023) Decision-Aware Conditional GANs for Time Series Data. *ICAIF '23: Proceedings of the Fourth ACM International Conference on AI in Finance*. November 2023 p36-45 doi.org/10.1145/3604237.3626855
36. Tensor Flow (2024) Generate music with an RNN. https://www.tensorflow.org/tutorials/audio/music\_generation Accessed 6th March 2024
37. Vidal, A. and Busso, C. (2023) Multimodal attention for lip synthesis using conditional generative adversarial networks. *Speech Communication*. 153, e102959 doi.org/10.1016/j.specom.2023.102959
38. Vinnay, M. Pujitha, K. Faisal, A. Ganesh, D. and Manjunath, G. (2022) Web-Based Music Genre Classification For Timeline Song Visualization And Analysis. *IEEE Access*. 9, p18801-18816 doi:10.1109/ACCESS.2021.3053864
39. Wijanto, E. Cheng, H. Liao, B. Huang, C. and Chang, Y. (2024) Research on Dispersion Compensation of FD-OCT System via Pix2Pix GAN Technique. *IEEE Access*. 12, p30976-30988 doi:10.1109/ACCESS.2024.3368051
40. Wu, A. Stouff, R. and Biljecki, F. (2022) Generative Adversarial Networks in the built environment: A comprehensive review of the application of GANs across data types and scales. *Building and Environment*. 223, e109477 doi.org/10.1016/j.buildenv.2022.109477
41. Zhong, H. (2023) Challenges, Corresponding Solutions, and Applications of Generative Adversarial Networks. *International Conference On Artificial Intelligence, Database And Machine Learning (AIDML 2023)*. 57, DOI:https://doi.org/10.54097/hset.v57i.9991