DEPARTMENT OF INFORMATION AND ELECTRONIC ENGINEERING

POSTGRADUATE PROGRAMME
MASTER OF SCIENCE (MSC) IN WEB INTELLIGENCE

Gutenberg Analytics

MASTER'S THESIS

of

NIKOLAOS POPIS
&
GEORGE PARDIS

Supervisor: Michail Salampasis
Professor, International Hellenic University (IHU)

Thessaloniki, June 2023

DEPARTMENT OF INFORMATION AND ELECTRONIC ENGINEERING
POSTGRADUATE PROGRAMME
MASTER OF SCIENCE (MSC) IN WEB INTELLIGENCE

Gutenberg Analytics

MASTER'S THESIS

of

NIKOLAOS POPIS
&
GEORGE PARDIS

Supervisor: Michail Salampasis
Professor, IHU

Approved by the three-member examination committee on June 30th, 2023.

(Signature)	*(Signature)*	*(Signature)*
.....	.....	.....
, IHU	, IHU	, IHU

Thessaloniki, June 2023

(Signature)

.....

Nikolaos Popis

Computer engineer

(Signature)

.....

George Pardis

Computer engineer

© 2023 All rights reserved

Περίληψη

Η παρούσα διατριβή παρουσιάζει μια νέα διαδικτυακή εφαρμογή που αναπτύχθηκε για την εξαγωγή μετα-δεδομένων και την ανάλυση έργων που προέρχονται από το Project Gutenberg. Η εφαρμογή αυτή προσφέρει ένα ευρύ φάσμα λειτουργιών, όπως το να επιτρέπει στους χρήστες να αναζητούν και να επιλέγουν θεατρικά έργα από την εκτεταμένη συλλογή του Project Gutenberg και να τα κάνουν προεπισκόπηση σε μορφή ακατέργαστου κειμένου ή σε μορφή HTML. Παρέχει επίσης τη δυνατότητα δημιουργίας και εφαρμογής προσαρμοσμένων κανόνων CSS ή επιλογής από μια δυναμικά παραγόμενη λίστα επιλογών CSS για την προσαρμογή της οπτικής παρουσίασης. Η εφαρμογή παράγει μια τελική προβολή πίνακα που εμφανίζει την πράξη, τον χαρακτήρα, το περιεχόμενο διαλόγου και την ανάλυση συναισθήματος για κάθε καταχώρηση. Οι χρήστες μπορούν να φιλτράρουν τα εμφανιζόμενα δεδομένα με βάση τις προτιμήσεις τους, να τροποποιούν το συναίσθημα, να επεξεργάζονται το περιεχόμενο κειμένου και να αποθηκεύουν το αναλυμένο κείμενο στη βάση δεδομένων. Επιπλέον, τα προηγουμένως αναλυμένα έργα μπορούν να ανακτηθούν εύκολα για μελλοντική αναφορά, εξασφαλίζοντας με αυτό τον τρόπο την πρόσβαση σε προηγούμενες αναλύσεις και διευκολύνοντας τη συνέχεια της διαδικασίας ανάλυσης. Η εφαρμογή προσφέρει επίσης αναλυτικά στοιχεία για το επιλεγμένο έργο, συμπεριλαμβανομένων των ρόλων, του αριθμού των διαλόγων και του αριθμού των λέξεων ανά εγγραφή. Η ανάλυση συναισθήματος πραγματοποιείται σε επίπεδο διαλόγου, πράξης, χαρακτήρα και της εξέλιξης του θεατρικού στον χρόνο, επιτρέποντας στους χρήστες να παρακολουθούν τις αλλαγές στο συναίσθημα καθ' όλη τη διάρκεια του έργου. Επιπλέον, οι χρήστες μπορούν να δημιουργούν και να διαχειρίζονται εξατομικευμένες λίστες ανάγνωσης, παρέχοντας ευελιξία στην οργάνωση των επιλογών τους. Αυτή η διαδικτυακή εφαρμογή συμβάλλει στον τομέα της ανάλυσης θεατρικών έργων παρέχοντας ολοκληρωμένα εργαλεία και αναλυτικά στοιχεία, δίνοντας στους χρήστες τη δυνατότητα να κατανοήσουν βαθύτερα τα επιλεγμένα έργα και να ενισχύσουν τη δέσμευσή τους.

Λέξεις Κλειδιά: web intelligence, Project Gutenberg, metadata extraction, sentiment analysis.

Abstract

This thesis presents a novel approach to extract metadata and analyze plays sourced from Project Gutenberg using a web application. The developed web application offers a wide range of functionalities and analytics, allowing users to search and select plays from the extensive Project Gutenberg collection and preview them in raw text or rendered HTML formats. It also gives the ability to create and apply custom CSS rules or select from a dynamically generated list of CSS selectors to customize the visual presentation. The application generates a final table view displaying the act, role, dialog content and sentiment analysis for each entry. Users can filter the displayed data based on their preferences, modify sentiment, edit textual content and store the analyzed text in the database. Moreover, previously analyzed plays can be conveniently retrieved for future reference, ensuring seamless access to previous analyses and facilitating continuity in the analysis process. The app also offers analytics for the selected play, including roles, the number of dialogs and the number of words per entry. Sentiment analysis is conducted at the dialog, act, character and timeline levels, enabling users to track changes in sentiment throughout the play. Additionally, users can create and manage personalized reading lists, providing flexibility in organizing their selections. This web application contributes to the field of literature analysis by providing comprehensive tools and analytics, empowering users to gain a deeper understanding of the selected plays and enhancing their engagement.

Keywords: web intelligence, Project Gutenberg, metadata extraction, sentiment analysis.

Table of contents

1 *Introduction* *1*

1.1 Background 1

1.2 Objectives 1

1.3 Literature Analytics 1

1.4 Thesis structure 2

2 *Project Gutenberg: Purpose and Significance* *3*

2.1 History and Evolution of Project Gutenberg 3

2.2 Significance of Project Gutenberg in Literature Preservation 4

2.3 Open Access and Public Domain Literature 5

2.4 Literature Analytics 6

3 *System Design and Implementation* *7*

3.1 Architecture Overview 7

3.1.1 Introduction 7

3.1.2 System Components 7

3.1.3 Client-Server Architecture 8

3.1.4 Technologies and Tools 8

3.1.5 Communication and Data Flow 9

3.1.6 Deployment Considerations 10

3.2 Data Collection and Preprocessing 10

3.2.1 Introduction 10

3.2.2 Data Collection 10

3.2.3 Data Preprocessing 11

3.3 Application Development 12

3.3.1 Introduction 12

3.3.2 Architectural Decisions 12

3.3.3 User Interface Design 15

3.4 Technologies and Tools Used 16

3.4.1 Introduction 16

3.4.2 Backend Technologies and Additional Libraries 16

3.5 Features and Functionalities of the App 17

3.5.1 Login and Logout 18

3.5.2 Search Anything 19

3.5.3 Book List Management 20

3.5.4 Raw Text and Rendered HTML Preview 24

3.5.5 CSS Rule Creation 26

3.5.6 Table View and Interactive Modifications 28

3.5.7 Database Integration 31

3.5.8 Sentiment Analysis and Analytics 33

3.5.9 Analytics Charts 34

3.5.10 Sentiment Timeline 38

3.5.11 Chart Comparison 41

3.6 Methodology 44

3.6.1 Development Approach 44

3.6.2 Requirements Gathering 45

3.6.3 Design and Prototyping 45

3.6.4 Development Process 45

3.6.5 Data Collection and Preprocessing 45

3.6.6 User Testing 45

3.6.7 Ethical Considerations 46

3.7 Challenges and Limitations 46

4 Evaluation 48

4.1 Introduction 48

4.2 Purpose and Objectives of Evaluation 48

4.3 Designing the Evaluation Survey 48

4.4 Administering the Evaluation Survey 49

4.5 Data Collection and Analysis 49

4.6 Interpretation and Application of Evaluation Findings 49

4.7 Survey Results 50

4.7.1 Was it easy to navigate and use the web application? 50

4.7.2 Did you find the search feature helpful? 50

4.7.3 Did the sentiment analysis feature provide valuable insights into the plays? 50

4.7.4 Did the provided analytics enhance your understanding of the plays? 50
4.7.5 Were you able to effectively organize and manage your selections using the personalized reading lists? 50
4.7.6 Did you find the web application useful for conducting literature analytics? 51
4.8 Conclusion 51
5 Future Work and Improvements 52
5.1 Potential enhancements to Metadata extraction 52
5.2 Scalability and Performance Optimization 53
5.3 Additional Improvements 54
6 Conclusion 56
6.1 Summary of Achievements 56
6.2 Contributions to the Field 56
6.3 Reflection on the Research Process 58
6.4 Implications and Future Directions 58
7 References 59

Table of figures

FIGURE 1 - PROJECT GUTENBERG WEBSITE 4
FIGURE 2 - APPLICATION LOGIN PAGE 19
FIGURE 3 - INVALID/EMPTY LOGIN CREDENTIALS 19
FIGURE 4 - APPLICATION SEARCH FUNCTIONALITY 20
FIGURE 5 - APPLICATION BOOK LIST MENU 21
FIGURE 6 - BOOK LIST CREATION 22
FIGURE 7 - BOOK LIST VIEW 22
FIGURE 8 - BOOK LIST DELETE MODAL 23
FIGURE 9 - ADD BOOK TO LIST MENU 23
FIGURE 10 - BOOK LIST ITEM VIEW 24
FIGURE 11 - BOOK HTML VIEW 25
FIGURE 12 - BOOK RAW HTML VIEW 25
FIGURE 13 - CREATE RULES DEFAULT OPTIONS 27
FIGURE 14 - CREATE RULES CUSTOM RULE 28
FIGURE 15 - RULES IN HTML VIEW 28
FIGURE 16 - DIALOGUES TABLE 29
FIGURE 17 - ACT FILTERING 29
FIGURE 18 - CHARACTER FILTERING 30
FIGURE 19 - TEXT FILTERING 30
FIGURE 20 - TEXT UPDATE 31
FIGURE 21 - CODE FOR GENERATING ANALYTICS (PART ONE) 36
FIGURE 22 - CODE FOR GENERATING ANALYTICS (PART TWO) 37
FIGURE 23 - ANALYTICS CHARTS (TOP) 37
FIGURE 24 - ANALYTICS CHARTS (DOWN) 38
FIGURE 25 - CODE FOR GENERATING SENTIMENT TIMELINE 40
FIGURE 26 - SENTIMENT TIMELINE CHART 41
FIGURE 27 - SENTIMENT TIMELINE CHART WITH CUSTOM STEP SIZE 41
FIGURE 28 - ANALYTICS CHARTS COMPARISON DROPDOWN MENU 43
FIGURE 29 - ANALYTICS CHARTS WITH COMPARISON (TOP) 43
FIGURE 30ANALYTICS CHARTS WITH COMPARISON (DOWN) 44
FIGURE 31 - SENTIMENT TIMELINE WITH COMPARISON 44

1 Introduction

1.1 Background

In today's digital age, the great amount of textual resources available on the web presents new opportunities and challenges for researchers, educators and enthusiasts. Project Gutenberg offers an extensive collection of such resources and the analysis of theatrical plays and books is essential in understanding the tones of theatrical plays and exploring their themes, character dynamics and emotional elements. This thesis introduces a web application designed to act as a useful tool to explore these opportunities by providing a range of functionalities for extracting metadata and conducting analysis on Project Gutenberg plays.

1.2 Objectives

The primary objective of our thesis is to develop a web application that enhances the analysis of plays sourced from Project Gutenberg. Our goal is to provide users with a range of functionalities and analytics that will lead to deeper insights on each play. By incorporating features such as text analytics and sentiment analysis, customizable CSS rules and the personalized reading lists, we aim to facilitate easier access, analysis and exploration to empower users with tools that enable them to delve into the textual content from multiple perspectives.

1.3 Literature Analytics

The development of this web application holds significant implications for researchers, educators and literature enthusiasts, offering a powerful tool for conducting comprehensive literature analytics. By automating the extraction of metadata and providing sophisticated analysis tools, we streamline the process of accessing and comprehending plays from Project

Gutenberg. Researchers can benefit from the application's functionalities for in-depth analysis, identifying trends and exploring the emotional nuances. Educators can use the application to enhance teaching and learning experiences while literature enthusiasts can gain new perspectives and insights into their favorite plays. The application's user-friendly interface and comprehensive features contribute to the democratization of knowledge and the advancement of literary research in the digital era.

1.4 Thesis structure

The remaining sections of this thesis is structured as follows: Chapter 3 delves into the purpose and significance of Project Gutenberg, discussing the history, evolution and role in literature preservation. Chapter 4 presents the system design and implementation, providing an overview of the application's architecture, data collection and preprocessing methods, application development process, technologies and tools used, as well as the features and functionalities offered. The chapter also discusses the methodology employed, challenges faced and limitations encountered. Chapter 5 explores future work and improvements, including potential enhancements to metadata extraction, scalability and performance optimization and additional improvements to enhance functionality. Finally, Chapter 6 concludes the thesis by summarizing the achievements, contributions and implications of the research, while providing reflections on the research process and discussing future directions in the field.

2 *Project Gutenberg: Purpose and Significance*

2.1 History and Evolution of Project Gutenberg

This section provides a comprehensive analysis of the history and evolution of Project Gutenberg, a pioneering effort aimed at the digital preservation and dissemination of literature. The project was founded in 1971 by Michael S. Hart, who recognized the potential of computers to make literature widely accessible by making electronic versions freely available to the public. The journey began with the digitalization of the United States Declaration of Independence, marking a significant milestone in the project's inception.

As technology advanced, Project Gutenberg evolved to embrace the digital revolution. To overcome initial challenges related to limited resources and technological limitations, it adopted new methodologies for digitalizing books and created a growing collection of electronic texts. This was achieved by evolving from plain-text files to more sophisticated eBook formats like HTML, EPUB and MOBI. This transition facilitated improved accessibility and compatibility across various devices, catering to the needs and preferences of a diverse readership.

Over time, Project Gutenberg expanded its scope beyond individual works and aimed to digitize entire libraries and genres. Today, the project has an extensive collection of literary works, promotes a culture of knowledge sharing and ensures widespread access to literary treasures. The invaluable contributions of volunteers and the continuous growth of the collection have made Project Gutenberg a cornerstone of digital libraries and a valuable resource for researchers, students and literature enthusiasts.

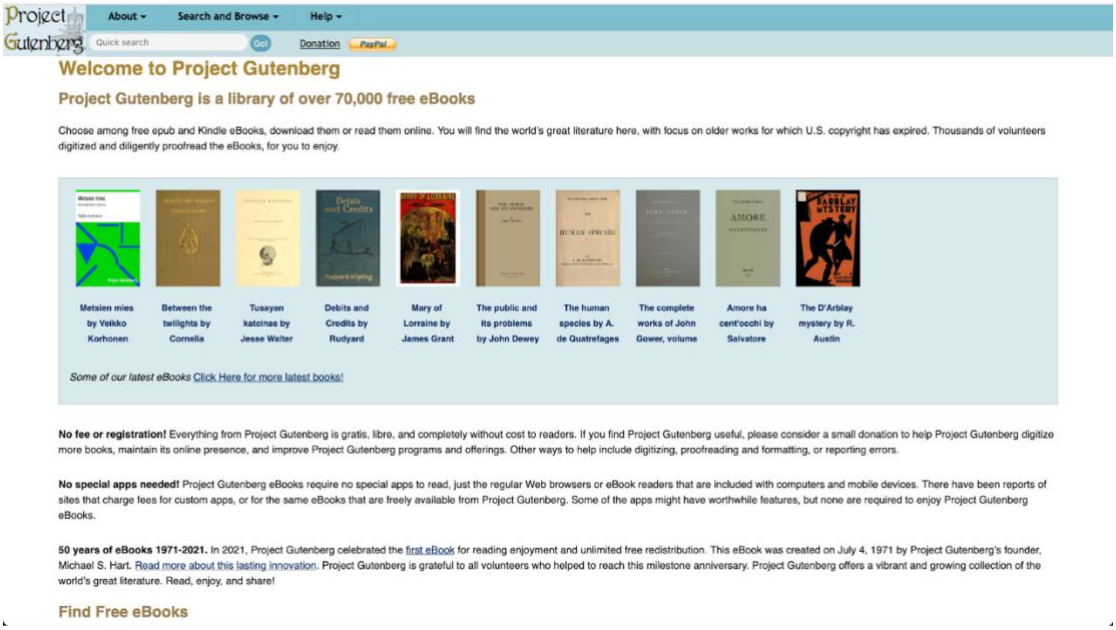

Figure 1 - Project Gutenberg Website

2.2 Significance of Project Gutenberg in Literature Preservation

Project Gutenberg plays a center role in the preservation of cultural heritage. By digitally archiving and disseminating an extensive collection of books, the project ensures that these literary treasures are not lost to time and are always available for free. Through its digitization efforts, it ensures the longevity and accessibility that might otherwise be limited to physical copies vulnerable to deterioration, loss, or restricted availability.

Digital preservation offered by Project Gutenberg addresses the challenges of preserving literature in the digital age. By creating a reliable and sustainable repository of digital books, Project Gutenberg safeguards literary works from the risks of physical degradation and loss. This preservation effort ensures that future generations can explore and appreciate the richness of literary achievements that span across cultures, time periods and genres.

Furthermore, Project Gutenberg's significance in literature preservation extends beyond the preservation of individual works [1]. By curating a diverse collection of books from diverse cultures and eras, it promotes cross-cultural understanding, intellectual exploration and the exchange of ideas. The accessibility and availability of these works contribute to a vibrant and inclusive literary landscape, enabling readers and researchers to delve into a vast array of knowledge and artistic expressions.

Project Gutenberg has been instrumental in quantifying statistical properties of natural language in numerous cases. The field of complex systems and quantitative linguistics heavily relies on

Project Gutenberg's extensive collection of books for statistical analysis. Pioneering works by Ebeling et al and Schurman and Grassberger in the 1990s utilized up to 100 books from Project Gutenberg to study long-range correlations, while Baayen investigated vocabulary growth curves [2]. Since then, Project Gutenberg has become an indispensable resource for quantitatively analyzing language, exploring universal properties like correlations, the scale-free nature of word-frequency distribution and various aspects related to genres and emotions.

While Project Gutenberg has undeniably proven its value to the research community, it has often been used in a careless and unsystematic manner. Firstly, most studies only examine a small subset, typically fewer than 20 books, out of the vast collection of over 50,000 books available in Project Gutenberg. Furthermore, these subsets often consist of the same manually selected books, such as the ubiquitous "Moby Dick," resulting in potentially biased and correlated datasets. Although a well-curated subset may suffice for specific linguistic inquiries, the limitation lies in the laborious process of manually downloading and processing individual Project Gutenberg books, hindering the use of more books. Particularly when studying linguistic laws and associated exponents, the intricate variability they exhibit can be challenging to capture with restricted Project Gutenberg samples. Another compelling example is the observation of double-power laws in Zipf's law, which requires extensive samples and has been suggested to be an artifact of text mixing.

Also, different studies employ diverse filtering techniques to extract, parse, select, tokenize and clean the data, or sometimes fail to provide sufficient methodological details. Consequently, even when supposedly using the same Project Gutenberg data, studies may end up with slightly different datasets. These limitations collectively raise concerns about the replicability and generalizability of previous and future research findings. While efforts have been made to achieve this for various textual datasets in machine learning, such as the UCI machine learning repository and diachronic corpora for studying language change, like The Corpus of Contemporary American English, similar initiatives have been lacking for Project Gutenberg data.

By addressing these concerns and establishing standardized protocols for utilizing Project Gutenberg data, we can enhance the reliability and applicability of research in the field, benefiting both current and future studies in quantitative linguistics [3].

2.3 Open Access and Public Domain Literature

Another of the key principles underlying Project Gutenberg is open access to literature. Open access refers to the unrestricted availability of scholarly and creative works, enabling anyone to access, use and build upon them without legal or financial barriers. This is achieved by focusing on books that have entered the public domain, where copyright restrictions no longer

apply. This commitment to open access ensures that literary works are not only preserved but also accessible to a global audience, encouraging a culture of knowledge sharing and intellectual growth.

Furthermore, by making public domain works freely available, Project Gutenberg allows for a broader and more inclusive literary landscape. It empowers individuals from diverse backgrounds, regardless of socioeconomic status or geographic location, to explore and enjoy plays that may have otherwise been restricted to the privileged.

Also, a broader implication of open access is that it enhances creativity, inspiration and the development of new ideas as users are free to adapt, mix and build upon existing works.

In conclusion, Project Gutenberg's commitment to open access principles ensures the accessibility, preservation and dissemination of cultural heritage. By making public domain works freely available, it contributes to the democratization of knowledge, the creativity growth and the advancement of research and education.

2.4 Literature Analytics

In recent years, the field of literature has witnessed a transformative shift with the integration of analytics, which involves the systematic analysis and interpretation of literary texts using data-driven approaches. This chapter delves into the significance of analytics in literature work, highlighting its various applications and benefits that have emerged because of advancements in digital technologies and computational methods. By exploring the role of analytics in literature, we gain a deeper understanding of its potential and the opportunities it presents for researchers and scholars.

Literature analytics is a multidisciplinary field that employs analytical techniques and tools to extract insights from literary texts. It encompasses a broad range of methodologies and applications, allowing researchers to uncover patterns, themes and meanings that might otherwise remain hidden. It is important to differentiate between traditional literary analysis, which relies primarily on subjective interpretations and literature analytics, which utilizes objective data-driven approaches. Finally, the convergence of digital technologies and literary studies has given rise to the field of digital humanities, which has played a pivotal role in the development of literature analytics. The availability of vast digital archives and computational tools has revolutionized the way literary works are analyzed, offering new perspectives and methodologies for researchers. Notable digital humanities projects focused on literature analytics have paved the way for further advancements in the field.

3 System Design and Implementation

3.1 Architecture Overview

3.1.1 Introduction

The Architecture Overview section aims to provide a comprehensive and detailed explanation of the system architecture for the application developed in this master thesis. This section presents an in-depth analysis of the key components and their interactions, emphasizing the client-server architecture employed. Furthermore, the section discusses the technologies and tools chosen for the development process, providing justifications for their selection.

3.1.2 System Components

The application comprises three main components: the frontend, the backend and the database. The frontend component is responsible for rendering the user interface and facilitating user interactions. It encompasses the visual elements and user experience, including the display of books, creation of favorite lists and the selection of individual books. The backend component handles the business logic and serves as the intermediary between the frontend and the database. It encompasses the server-side code responsible for handling API requests, executing operations on the data and generating responses. The database component stores the book metadata and user-related information, providing efficient data storage and retrieval capabilities.

3.1.2.1 Frontend Component

The frontend component is developed using Angular, a popular frontend framework. Angular allows for the creation of dynamic and responsive user interfaces with its component-based architecture. It provides extensive UI capabilities, including customizable book listings, interactive forms for creating favorite lists and seamless navigation. Additionally, Angular's

data binding features enable real-time updates of the user interface based on the underlying data changes.

3.1.2.2 Backend Component

The backend component is developed using Python with the FastAPI framework. Python was chosen as the primary programming language due to its versatility, readability and extensive library support. FastAPI, a modern web framework, was selected for its high performance and asynchronous capabilities. FastAPI leverages Python's asynchronous programming features, allowing for efficient handling of multiple client requests concurrently. It also provides automatic API documentation generation, simplifying the documentation process and facilitating collaboration.

3.1.2.3 Database Component

The database component utilizes PostgreSQL, a robust and reliable database management system. PostgreSQL was chosen for its scalability, support for complex queries and ACID-compliant transactions. It allows for efficient storage and retrieval of book metadata and user-related information. The database schema is designed to optimize queries for retrieving books, creating and managing favorite lists and handling user authentication and authorization.

3.1.3 Client-Server Architecture

The system follows a client-server architecture, which enables a clear separation between the presentation layer (frontend) and the application logic layer (backend). The client, implemented using Angular, runs in the user's browser and communicates with the server through HTTP(S) protocols. The server, developed using Python with the FastAPI framework, receives the requests from the client, processes them and sends back the necessary data. This architecture promotes scalability, maintainability and modularity. It allows for the separation of concerns, enabling independent development and updates of the frontend and backend components. The client-server architecture also facilitates seamless integration with additional client applications, such as mobile applications or third-party services, in the future.

3.1.4 Technologies and Tools

This subsection provides an in-depth explanation of each chosen technology and tool used in the development process. It highlights the advantages and justifications for selecting these technologies

3.1.4.1 Python

Python was chosen as the primary programming language for the backend due to its versatility, extensive libraries and large developer community. Python provides a robust ecosystem that enables rapid development, efficient code organization and easy integration with various frameworks and libraries.

3.1.4.2 FastAPI

FastAPI, a modern web framework for building APIs with Python, was selected for its high performance and asynchronous capabilities. FastAPI leverages Python's asynchronous programming features, allowing for efficient handling of multiple concurrent requests. It offers automatic API documentation generation based on standardized OpenAPI specifications, reducing the effort required for API documentation and enabling better collaboration between frontend and backend developers.

3.1.4.3 PostgreSQL

PostgreSQL was chosen as the database management system for its reliability, scalability and support for complex queries. It offers ACID-compliant transactions, ensuring data integrity and consistency. PostgreSQL's extensibility allows for the implementation of custom functions and data types, facilitating the storage and retrieval of book metadata and user-related information.

3.1.4.4 Angular

Angular, a popular frontend framework developed by Google, was chosen for its robust features and extensive ecosystem. Angular provides a component-based architecture, allowing for the development of reusable UI components. Its data binding capabilities enable real-time updates of the user interface, enhancing the user experience. Angular's extensive tooling and testing support simplify development and maintenance tasks.

3.1.5 Communication and Data Flow

This subsection focuses on explaining the communication and data flow within the application. The frontend communicates with the backend through RESTful APIs. The frontend sends HTTP requests to the backend endpoints, which are defined in the FastAPI application. These endpoints handle the requests by processing the data, executing business logic, retrieving information from the database if necessary and returning the appropriate responses to the frontend. The data flow follows a request-response pattern, allowing for efficient and controlled communication between the client and the server. The use of RESTful APIs promotes interoperability, scalability and decoupling between frontend and backend components.

3.1.6 Deployment Considerations

This subsection discusses the deployment considerations for the application. It covers the hosting options for the frontend and backend components, ensuring optimal performance and availability. The frontend component can be hosted using a web server such as Nginx or Apache, which provides static file hosting and configuration options. The backend, developed with FastAPI, can be deployed using ASGI servers like Uvicorn or Gunicorn, ensuring efficient handling of concurrent requests. The database component, utilizing PostgreSQL, can be deployed separately on a dedicated server or cloud-based environment. Additionally, security measures such as HTTPS encryption, user authentication and authorization mechanisms should be implemented to protect sensitive user data and ensure secure communication between the components.

This chapter provided a comprehensive overview of the system architecture for the developed application. It covers the frontend, backend and database components, explaining their roles and interactions. The client-server architecture and the technologies and tools chosen for the development process are thoroughly discussed, highlighting their benefits and justifications. The communication and data flow within the application are explained, emphasizing the use of RESTful APIs. Deployment considerations, including hosting options and security measures, are also discussed, ensuring a robust and secure system implementation. This chapter serves as a solid foundation for understanding the overall system design and implementation, providing valuable insights into the application's structure and functionality.

3.2 Data Collection and Preprocessing

3.2.1 Introduction

The Data Collection and Preprocessing section focuses on the process of gathering and preparing the data required for the application developed in this master thesis. This section explains the data collection methods employed to obtain the book metadata from Project Gutenberg and the preprocessing techniques applied to ensure data quality and usability.

3.2.2 Data Collection

The data collection process in the application involves obtaining book metadata from Project Gutenberg, a prominent digital library that provides free access to an extensive collection of literary works. Project Gutenberg offers different methods for accessing and downloading book metadata, including web scraping and official APIs. The chosen method for collecting book metadata will be described in detail, outlining the steps involved in retrieving relevant

information such as book titles, authors, publication dates and genres. Additionally, any limitations or challenges encountered during the data collection process will be discussed.

The chosen method for collecting book metadata from Project Gutenberg is web scraping, which involves programmatically extracting data from web pages by parsing the HTML structure of the website. The application utilizes libraries like BeautifulSoup or Gutenbergpy to retrieve the desired book metadata. The process begins by retrieving the URLs of the books either through website exploration or targeted search queries. Once the book URLs are obtained, the application accesses the corresponding web pages using libraries for parsing and navigating HTML. By inspecting specific HTML elements, the application extracts and stores relevant metadata such as book title, author, publication date and genre for further processing or storage.

During the data collection process for Project Gutenberg, various limitations and challenges can arise. Changes in the website structure, such as updates to HTML elements, class names, or page layouts, may require adjustments to the scraping code to maintain the extraction of book metadata. Additionally, rate limiting measures or restrictions on the number of requests per unit of time imposed by Project Gutenberg should be respected to ensure fair usage and maintain a positive relationship with the website. Furthermore, the quality and completeness of the collected data may be affected by occasional errors or missing information. To address this, error handling and data validation mechanisms should be implemented in the application to ensure the reliability and usability of the collected metadata.

By leveraging web scraping techniques and libraries like BeautifulSoup or Gutenbergpy, the application collects book metadata from Project Gutenberg. The metadata, including book titles, authors, publication dates and genres, is extracted by parsing the HTML structure of the web pages. The data collection process is subject to potential limitations and challenges such as website structure changes, rate limiting, data quality issues and legal considerations. Addressing these challenges and ensuring compliance with ethical guidelines allows for the reliable acquisition of book metadata for further processing within the application.

3.2.3 Data Preprocessing

Data preprocessing plays a crucial role in ensuring the quality and usability of the collected book metadata. Several steps are involved in this process, starting with data cleaning. Data cleaning focuses on removing irrelevant or duplicate entries, handling missing values and addressing inconsistencies within the dataset. This step ensures that the collected book metadata is accurate and complete. Techniques commonly used for data cleaning include removing duplicate records, filling in missing values using statistical measures and correcting inconsistent data formats.

The next step in data preprocessing is data transformation. This involves converting the collected book metadata into a suitable format for further processing and analysis. Transformation may include structuring the data into formats like JSON or CSV, or integrating it into a database management system such as PostgreSQL. Additionally, data normalization techniques may be applied to ensure uniformity and comparability across different attributes, enabling effective analysis and interpretation.

Data integration is another important preprocessing step, involving the combination of collected book metadata with existing data sources or external datasets, if applicable. Integration enhances the functionality and value of the application by enabling enriched analysis. This process often involves matching and merging records based on common attributes or utilizing data linking techniques to establish relationships between different datasets, resulting in a more comprehensive and interconnected dataset.

To maintain data integrity and reliability, data validation and quality assurance measures are applied. Data validation checks verify the correctness of book titles, authors and other attributes against authoritative sources. Statistical analysis techniques may be used to identify potential outliers or anomalies in the dataset. Quality assurance techniques such as data profiling and data auditing assess the overall quality and integrity of the collected book metadata, ensuring its suitability for subsequent analysis and usage.

3.3 Application Development

3.3.1 Introduction

The Application Development section focuses on the implementation of the application developed in this master thesis. This section provides an overview of the development process, including the architectural decisions, user interface design and the implementation of key features. It also discusses the testing and debugging strategies employed to ensure the functionality and stability of the application.

3.3.2 Architectural Decisions

During the development of the application, several architectural decisions were made to ensure a well-structured and maintainable codebase that meets the requirements of the application. These decisions encompassed the selection of appropriate software architecture patterns, the application of design principles and the consideration of scalability and maintainability.

The chosen software architecture pattern for the application is the Model-View-Controller (MVC) pattern. The Model-View-Controller (MVC) pattern is widely used in software development to achieve a clear separation of concerns and enhance code organization. In this

pattern, the model, view and controller are distinct components that play specific roles in the application.

The model component represents the data and business logic of the application. It encapsulates the data structures, algorithms and rules required to manipulate and manage the application's data. In the backend, the model layer includes various components responsible for tasks such as data validation, interacting with the database and implementing business logic. By isolating these functionalities in the model, the application can maintain a coherent and consistent representation of the data, ensuring its integrity and reliability.

The view component is responsible for presenting the user interface to the application's users. It encompasses the visual elements and user interface logic required to render and display information. In the context of web development, the view layer focuses on generating the user interface and delivering it to the client. It may involve templates, HTML, CSS and JavaScript to create an interactive and visually appealing experience for the users. The view is often designed to be modular and reusable, allowing developers to separate the presentation logic from the underlying data and business logic.

The controller acts as an intermediary between the model and the view. Its primary role is to handle user input, process it and update the model and view accordingly. In the backend, the controller layer is responsible for receiving incoming requests, extracting relevant data and invoking the appropriate model operations to perform the necessary actions. It then determines the appropriate view to render as a response, ensuring that the user interface reflects the updated state of the application. By separating the responsibilities of handling user input, modifying the model and updating the view, the controller promotes maintainability, reusability and testability of the application.

In summary, the MVC pattern divides an application into three key components: the model, responsible for data and business logic; the view, responsible for the user interface; and the controller, acting as an intermediary between the model and the view. This separation of concerns allows for better code organization, promotes code reusability and enables easier maintenance and testing of the application.

During the development of the application, several design principles were followed to ensure a modular, flexible and readable codebase. These principles include SOLID (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation and Dependency Inversion) and DRY (Don't Repeat Yourself).

- The Single Responsibility Principle (SRP) states that each class or module should have a single responsibility. This helps in organizing the code into smaller, focused components that are easier to understand and maintain.

- The Open-Closed Principle (OCP) suggests that software entities should be open for extension but closed for modification. This means that new functionality can be added by extending existing components rather than modifying them, reducing the risk of introducing bugs.
- The Liskov Substitution Principle (LSP) states that subtypes should be substitutable for their base types without altering the correctness of the program. Following LSP ensures that derived classes or modules can be used interchangeably with their base classes, promoting code reusability.
- The Interface Segregation Principle (ISP) advises creating small, specific interfaces instead of large, general-purpose interfaces. This prevents clients from depending on unnecessary interfaces and allows for better decoupling between components.
- The Dependency Inversion Principle (DIP) suggests that high-level modules should not depend on low-level modules, but both should depend on abstractions. This promotes the use of interfaces and abstractions to facilitate easier integration, testing and maintenance.

By applying these design principles, the application's codebase becomes more flexible, extensible and easier to understand. It reduces code duplication, improves code maintainability and makes it easier to implement changes or add new features.

Throughout the development process, scalability and maintainability were also considered. The chosen architecture and design patterns provide a solid foundation for these aspects, but additional measures may be taken, depending on the specific requirements of the application.To ensure scalability, the backend is built using asynchronous programming techniques, leveraging libraries like FastAPI and AnyIO. Asynchronous programming allows the application to handle concurrent requests efficiently, improving performance and responsiveness.

Regarding maintainability, the use of the MVC pattern, along with design principles like SOLID, promotes code organization and modularity. Additionally, the separation of concerns provided by the MVC pattern makes it easier to identify and isolate bugs, perform targeted modifications and conduct thorough testing.

Regular code reviews, automated testing and continuous integration practices can further enhance maintainability and contribute to the overall quality of the codebase. These practices help identify potential issues early on and ensure that the code remains reliable, efficient and maintainable throughout the application's lifecycle.

The architectural decisions made during the development of the application, such as the selection of the MVC pattern, the application of design principles and the consideration of scalability and maintainability, contribute to a well-structured, modular and scalable codebase.

These decisions aim to ensure that the application meets the requirements, is easy to maintain and can accommodate future changes and growth.

3.3.3 User Interface Design

The user interface (UI) design of the application focuses on creating an intuitive and visually appealing experience for the users. The design process involves various stages, including wireframing, prototyping and iterative refinement. Usability and accessibility principles play a crucial role in ensuring that the UI is user-friendly and inclusive.

The design process typically starts with wireframing, which involves creating low-fidelity sketches or digital representations of the UI layout. Wireframes help in visualizing the structure and placement of different elements, such as navigation menus, forms, buttons and content sections. They serve as a blueprint for the overall UI design and provide a starting point for discussion and feedback.

Once the wireframes are finalized, prototyping comes into play. Prototypes are interactive representations of the UI that allow users or stakeholders to experience the flow and functionality of the application. Feedback gathered during the prototyping phase helps in identifying usability issues, refining the UI layout and improving the overall user experience.

Iterative refinement involves incorporating feedback from users into the UI design, making necessary adjustments to enhance usability and address any identified issues. This iterative process continues until the UI design aligns with the desired user experience and meets the application's goals.

Throughout the UI design process, usability and accessibility principles are given due consideration. Usability focuses on creating interfaces that are easy to use, intuitive and efficient. Key usability principles include simplicity, consistency, feedback and clear navigation. The UI design strives to minimize cognitive load, provide clear visual cues and ensure that user interactions are predictable and responsive.

Accessibility is another crucial aspect of UI design, aiming to make the application usable by individuals with disabilities or impairments. The UI design adheres to accessibility guidelines, such as the Web Content Accessibility Guidelines (WCAG), to ensure that users with visual, auditory, or motor disabilities can access and interact with the application. Accessibility considerations include providing alternative text for images, ensuring proper color contrast, implementing keyboard navigation support and using semantic HTML markup.

The chosen design framework or library, such as Angular Material or Bootstrap, also plays a significant role in shaping the UI design. These frameworks provide a set of pre-designed UI components, styles and patterns that facilitate the development of a consistent and visually appealing interface.

Angular Material, for example, offers a comprehensive collection of ready-to-use components following Material Design guidelines. It provides a cohesive design language, responsive layouts and accessibility features out of the box. Customization and theming options allow the UI to be aligned with the application's goals and branding. Developers can leverage the flexibility of Angular Material or similar frameworks to create a visually consistent and engaging UI while maintaining usability and accessibility standards.

3.4 Technologies and Tools Used

3.4.1 Introduction

This section provides an overview of the various technologies, frameworks and tools employed in the development of the application. This section discusses the rationale behind the selection of each technology and its role in the overall system architecture. It also highlights the benefits and advantages offered by these technologies in terms of efficiency, performance and ease of development.

3.4.2 Backend Technologies and Additional Libraries

The backend of the application is developed using Python with the FastAPI framework, which provides a powerful and efficient foundation for building web APIs. To further enhance the functionality and improve the performance of the backend, several additional libraries are utilized. These libraries extend the capabilities of the backend, enabling seamless integration with databases, efficient handling of asynchronous programming, secure communication with external services and streamlined data manipulation. Below we mention some of the most important libraries that have an important role for the application:

- FastAPI (version 0.88.0): FastAPI is a high-performance web framework for building APIs with Python. It combines the simplicity of Python with the speed and scalability of asynchronous programming.
- FastAPI JWT Auth (version 0.5.0): FastAPI JWT Auth is a library that provides JWT (JSON Web Token) authentication support for FastAPI applications. It simplifies the implementation of authentication and authorization mechanisms.
- FastAPI Pagination (version 0.11.0): FastAPI Pagination is a library that simplifies the implementation of paginated responses in FastAPI applications. It provides easy-to-use pagination utilities for handling large datasets.
- Gutenbergpy (version 0.3.4): Gutenbergpy is a Python wrapper for accessing the Project Gutenberg API. It simplifies the retrieval of book metadata from Project Gutenberg.

- Numpy (version 1.23.5): Numpy is a fundamental package for scientific computing in Python. It provides support for efficient numerical operations on multidimensional arrays and matrices.
- Pandas (version 1.5.2): Pandas is a powerful data manipulation and analysis library for Python. It provides high-performance data structures and data analysis tools, making it suitable for handling structured data.
- SQLAlchemy (version 1.4.41): SQLAlchemy is a popular SQL toolkit and Object-Relational Mapping (ORM) library for Python. It simplifies database access and provides a high-level API for interacting with databases.
- Uvicorn (version 0.20.0): Uvicorn is a lightning-fast ASGI server for Python. It enables the deployment of FastAPI applications with excellent performance and scalability.
- Beautifulsoup4 (version 4.11.2): Beautifulsoup4 is a library for parsing HTML and XML documents. It simplifies the extraction and manipulation of data from HTML and XML content.
- Roman (version 4.1): Roman is a library for working with Roman numerals in Python. It provides functions for converting Roman numerals to integers and vice versa.
- TextBlob (version 0.17.1) [4]: TextBlob is a library for processing textual data in Python. It provides a simple API for tasks such as sentiment analysis, part-of-speech tagging and noun phrase extraction.
- Chart.js: Chart.js is a JavaScript library that enables the creation of interactive and visually appealing charts and graphs on web pages.

These additional libraries enhance the functionality and capabilities of the backend application, allowing for efficient data processing, database interaction, asynchronous programming, template rendering and external API integration, among other tasks.

3.5 Features and Functionalities of the App

Our web application is designed to provide users with an immersive and comprehensive experience, offering an extensive array of features and functionalities that empower them to explore and analyze plays sourced from Project Gutenberg. By harnessing the power of technology, the application enhances the user's understanding and appreciation of literary works, unlocking valuable insights and enriching their overall experience.

One of the key features of the application is its vast collection of plays sourced from Project Gutenberg. Users can access a wide range of plays from different eras, genres and playwrights, allowing them to explore and delve into the world of theater. From the timeless classics of

William Shakespeare to the contemporary works of modern playwrights, the application offers a diverse selection that caters to various interests and preferences.

To facilitate seamless navigation and exploration, the application incorporates an intuitive and user-friendly interface. Users can effortlessly browse through the extensive play collection, filter and search for specific works and access detailed information about each play, including its synopsis, characters and themes.

3.5.1 Login and Logout

The application includes secure and user-centric login and logout functionality to ensure that only authorized individuals can access its features and content. To gain access to the application, users must provide their email or username for authentication purposes. Both fields are mandatory and the application prompts users to enter these details if they are left blank, ensuring that the necessary information is provided for a successful login.

For users who already have an account, the login feature allows them to enter their credentials and gain personalized access to the application's features and functionality. By logging in, users can unlock a range of personalized settings, preferences and content tailored to their specific needs and interests. This personalized experience enhances the overall usability and relevance of the application, ensuring that users can access their customized settings and enjoy a seamless experience each time they log in.

Logout, on the other hand, allows users to securely exit the application, ending their session and ensuring that their account remains protected. By logging out, users can prevent unauthorized access to their account and maintain the privacy and security of their personal information. This feature is particularly valuable when users access the application from shared devices or public spaces, as it ensures that their account is properly signed out and inaccessible to others.

Login and logout functionality within the application ensures a controlled and personalized user experience. By requiring authentication via email or username, the application creates a secure barrier that allows only authorized individuals to access its features. Whether it is accessing personalized content, managing preferences or maintaining account security, the login and logout features contribute to a secure and tailored experience for users, ensuring that their interactions with the application are both convenient and secure.

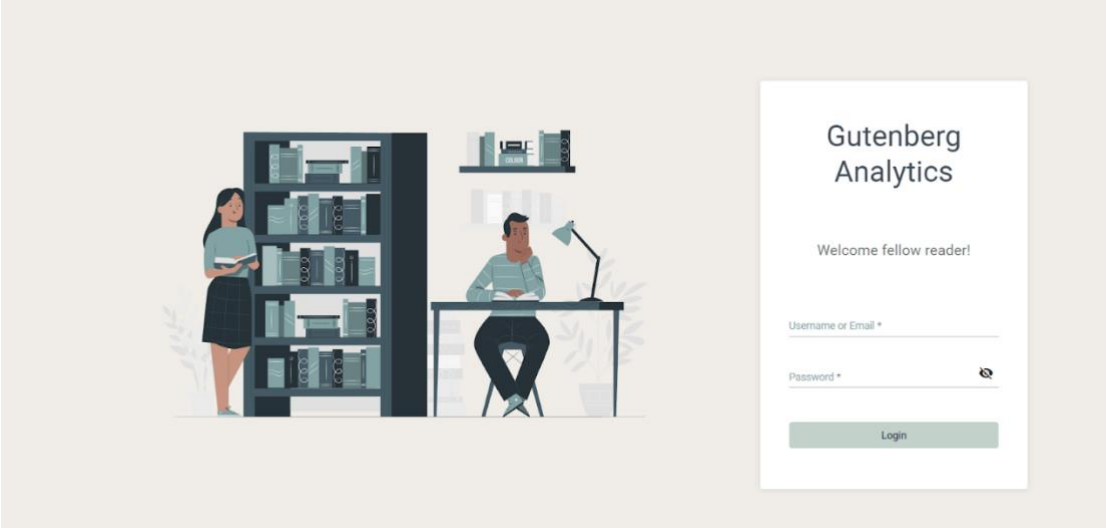

Figure 2 - Application Login Page

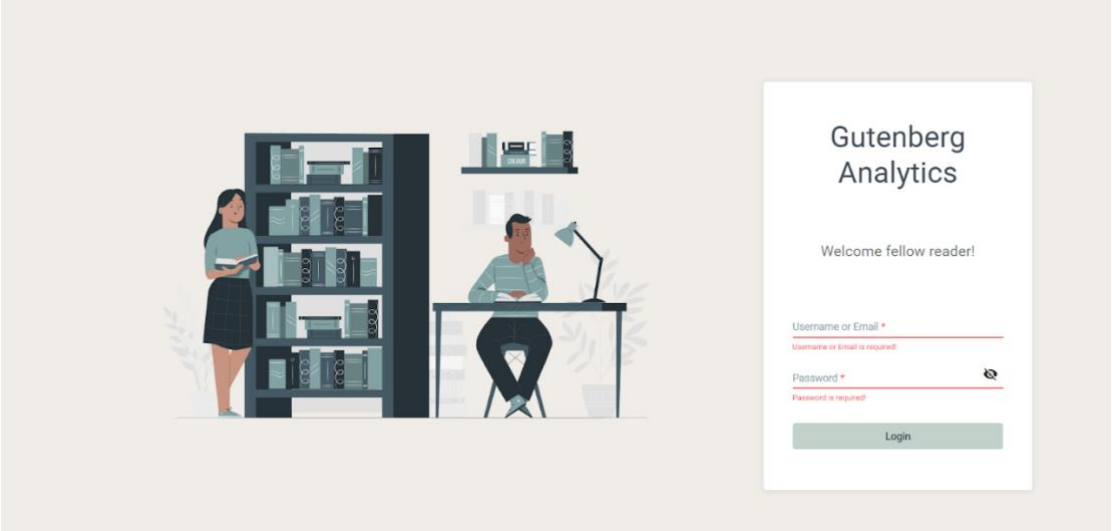

Figure 3 - Invalid/Empty login credentials

3.5.2 Search Anything

The application includes a powerful and comprehensive search function that allows users to navigate Project Gutenberg's vast collection with utmost ease. This feature allows users to search by a variety of parameters, including Gutenberg ID, play name, or author, providing a convenient and efficient way to locate and select content.

The Gutenberg ID search option allows users to directly enter the unique identifier associated with a particular play or literary work. This streamlined approach allows users to quickly retrieve the exact content they are looking for, bypassing any potential ambiguity. In addition, the search functionality allows users to search by play name. Users can enter the title of a play they are interested in and use the application's search capabilities to quickly find the content they want. Whether users have a specific play in mind or are looking to discover new works,

the play name search option provides a convenient way to navigate Project Gutenberg's extensive collection.

Users can also conduct searches based on the author's name. By entering the name of a specific author, users can retrieve a comprehensive list of plays attributed to that author. This search criterion proves particularly valuable for users who are fond of a particular playwright and wish to explore their complete body of work. It offers a convenient way to browse through the plays authored by their favorite writers, ensuring an enriched and tailored reading experience.

The search functionality within the application provides users with a robust and versatile tool for exploring Project Gutenberg's extensive collection. By offering search options based on the Gutenberg ID, play names and author names, users can easily locate and select desired content. Whether users have specific works in mind or are looking to discover new plays or authors, the comprehensive search functionality ensures a seamless and convenient experience, facilitating efficient access to an array of literary treasures.

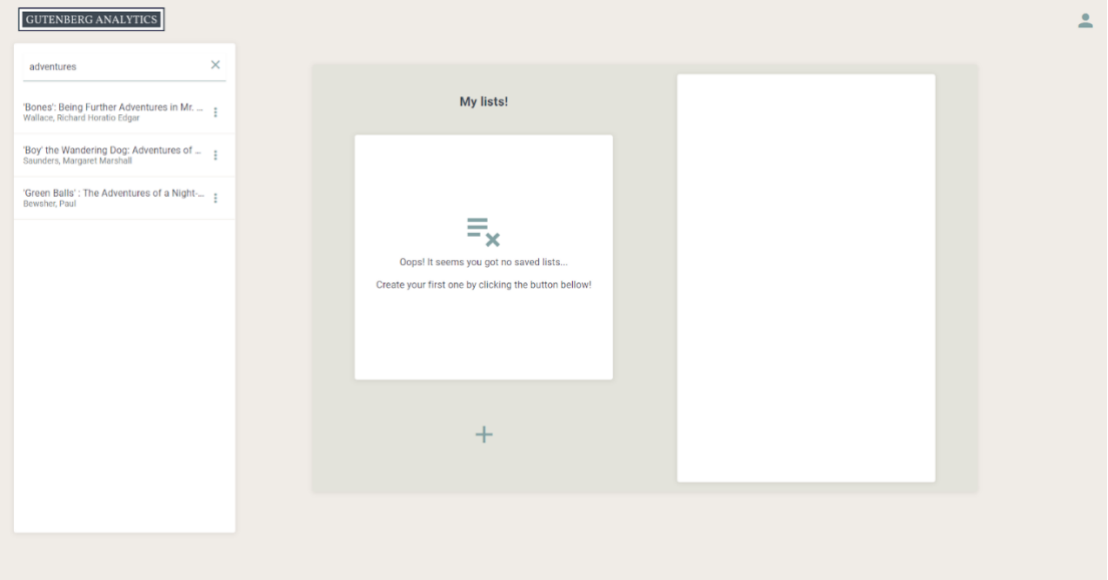

Figure 4 - Application Search Functionality

3.5.3 Book List Management

Book List Management offers users a versatile and personalized approach to organizing their literary collections. With this feature, users gain the ability to create and delete lists of books, as well as add or remove books from any list according to their preferences and needs.

The creation of book lists allows users to categorize their books based on various criteria such as genre, author, or reading priority. By organizing their collection into distinct lists, users can easily locate specific books and maintain a systematic overview of their reading materials. Whether it is a list of favorite novels, a collection of books to read for a specific purpose, or a

compilation of recommendations, the flexibility of book list creation ensures that users can tailor their collections to their individual interests and requirements.

In addition to creating lists, users can also delete lists that are no longer relevant or necessary. This feature enables users to keep their book management system streamlined and clutter-free by removing outdated or unused lists. By eliminating unnecessary lists, users can maintain a concise and efficient library organization, ensuring that their focus remains on the books that matter most to them.

Users can also add or remove books from any list at any time using the Book List Management feature. This level of flexibility allows users to adapt their collections as their reading preferences evolve. Whether they have finished a book and want to remove it from their "To Read" list, or whether they have discovered a new literary gem to add to their "Must Read" list, users have complete control over the content of their book lists. This dynamic feature ensures that users can continually curate and refine their collections to reflect their evolving literary tastes and interests.

By offering the ability to create and delete lists, add, or remove books, Book List Management provides users with a comprehensive and customizable solution for organizing and managing their book collections. This feature fosters a sense of ownership and personalization, enabling users to create a library system that aligns with their unique reading habits and preferences. Whether users are avid readers, students, or book enthusiasts, this flexible management tool enhances their overall reading experience and ensures that their literary collections are well-structured and easily accessible.

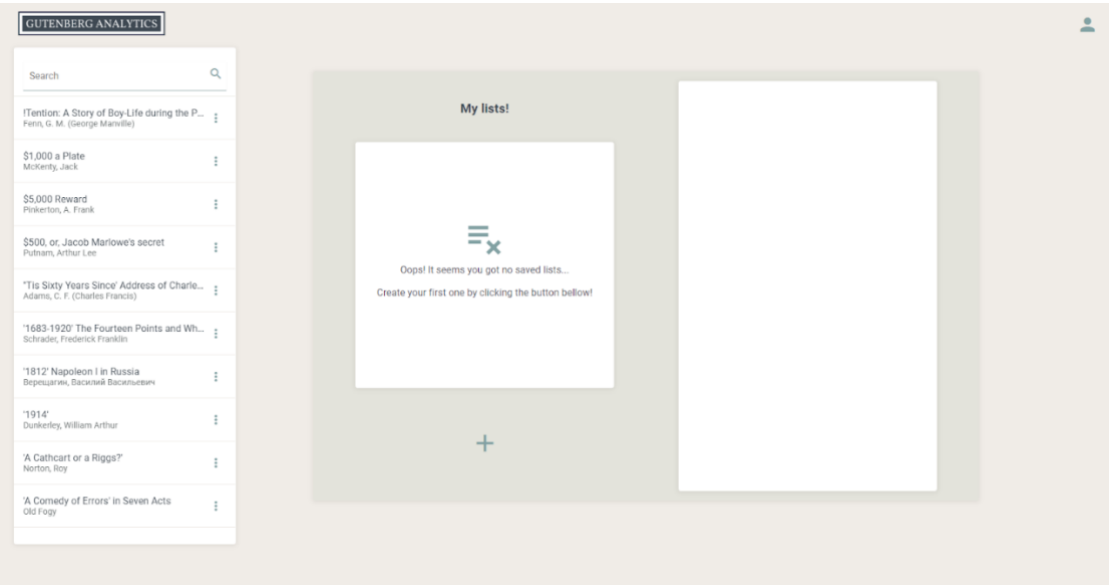

Figure 5 - Application Book List Menu

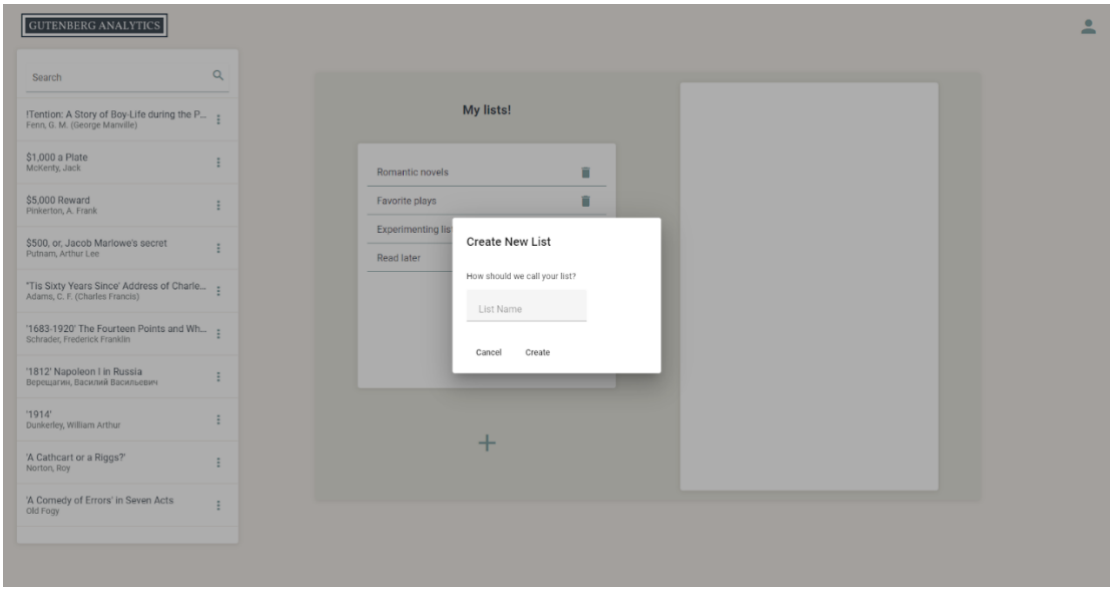

Figure 6 - Book List Creation

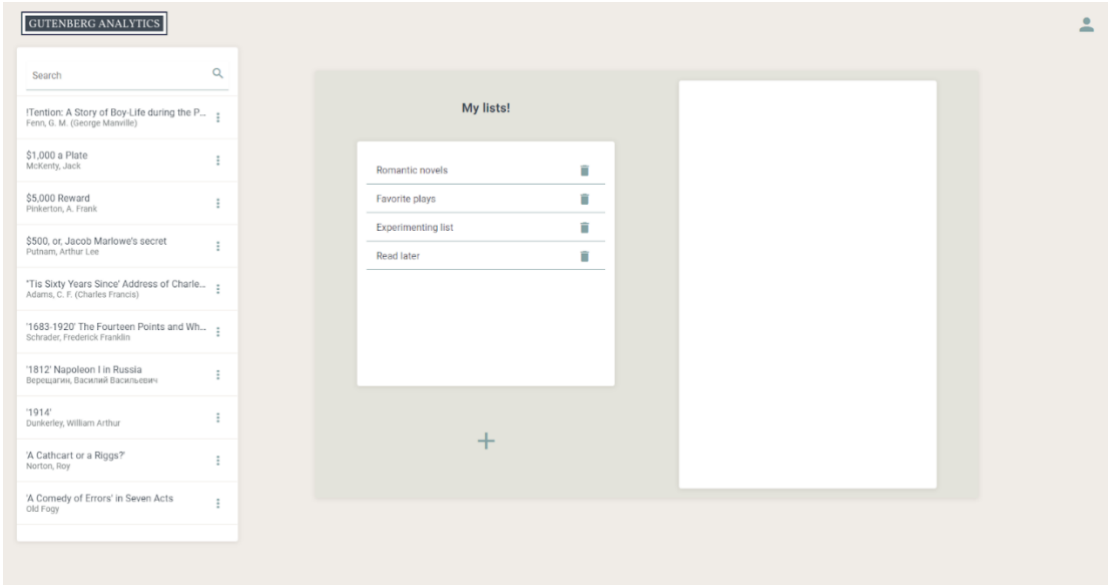

Figure 7 - Book List View

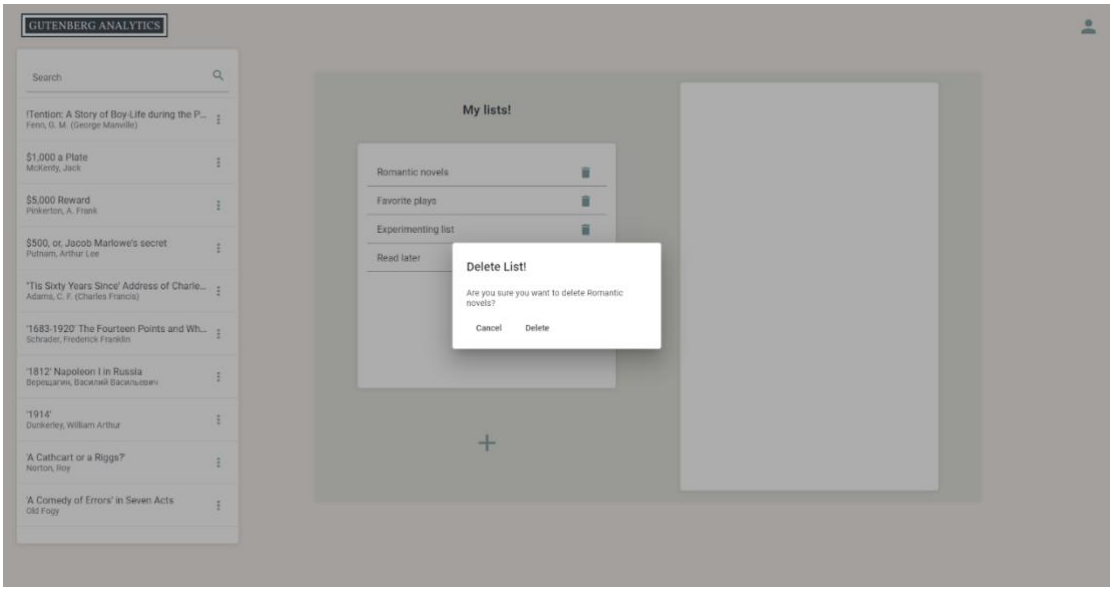

Figure 8 - Book List Delete Modal

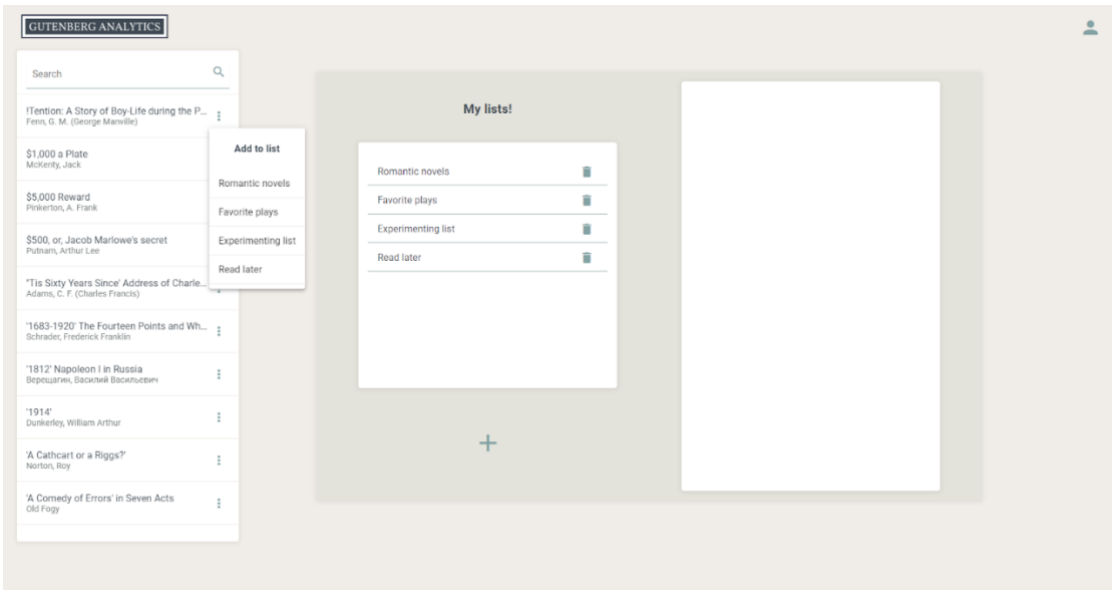

Figure 9 - Add Book to List Menu

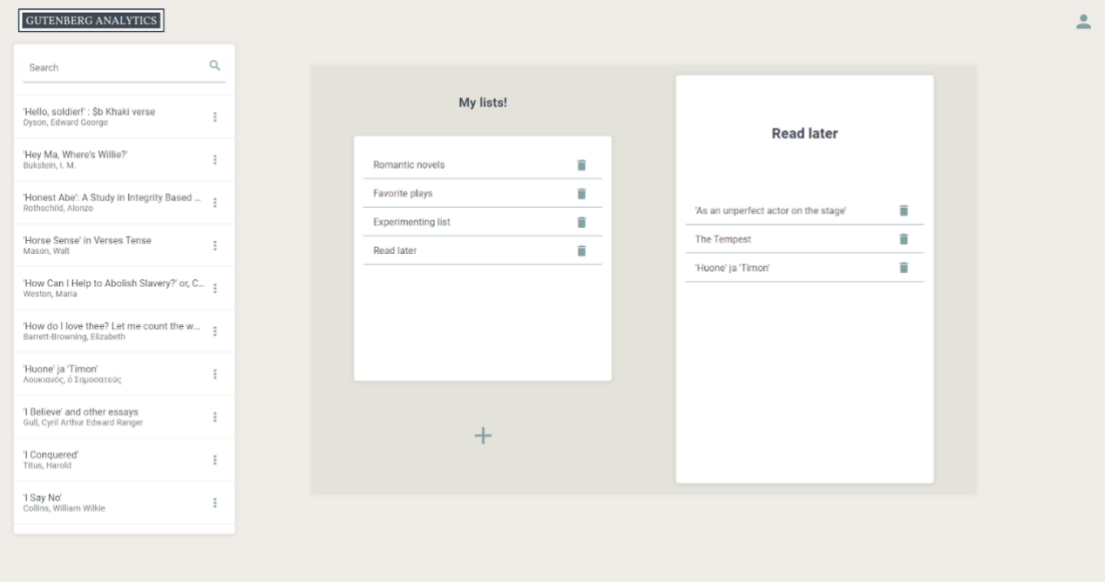

Figure 10 - Book List Item View

3.5.4 Raw Text and Rendered HTML Preview

Users are provided with the convenience of previewing plays in two distinct formats: raw text and rendered HTML. These two options cater to different preferences and offer unique advantages when it comes to exploring and evaluating the content of a play.

The raw text preview presents the play exactly as it is, in its unadulterated, original form. This format offers a straightforward and unembellished representation of the play's content, presenting the script in a plain and simple manner. By displaying the play in raw text, users can focus solely on the words and dialogue without any distractions. This minimalistic approach allows for a comprehensive examination of the play's structure, character interactions and plot progression.

On the other hand, the rendered HTML preview takes the play to a whole new level of visual presentation. By leveraging the capabilities of HTML, the preview transforms the play into a visually enhanced reading experience. The rendered HTML format replicates the original formatting, bringing to life the intricate details of the play's layout, such as stage directions, character names and scene descriptions. This visual rendition provides users with a more immersive and engaging encounter, enabling them to better comprehend the play's intended visual cues and theatrical elements.

The use of rendered HTML also opens possibilities for additional interactive features. Hypertext links can be embedded within the play, allowing users to navigate easily between different scenes, acts, or even related external resources. This dynamic nature of rendered HTML brings a new level of interactivity and exploration to the play, enhancing the overall user experience.

Overall, the availability of raw text and rendered HTML previews for plays provides a flexible and comprehensive approach to accessing and evaluating content. While the raw text format provides a stripped-down, uninterrupted view of the play's script, the rendered HTML format enhances the reading experience by replicating the original formatting and introducing interactive elements. Whether users prefer a minimalist examination or a visually immersive encounter, these preview options cater to their diverse needs and ensure a well-rounded exploration of the play's artistic and narrative qualities.

Figure 11 - Book Html View

Figure 12 - Book Raw Html View

3.5.5 CSS Rule Creation

This chapter delves into the powerful CSS (Cascading Style Sheets) rule creation functionality within the application. Recognizing the importance of visual presentation and customization in enhancing the reading experience, the application empowers users to create CSS rules and apply them to the HTML document of the play. This chapter explores the intricate details of this feature, emphasizing its versatility and impact on the visual aesthetics of the text.

A key aspect of our CSS rule creation functionality is the dynamic creation of an HTML tag list. Users are presented with a range of HTML tags to choose from, encompassing elements such as headings, paragraphs, quotes and more. This dynamic list ensures that users have access to a comprehensive set of tags, enabling them to precisely target specific elements within the

By selecting HTML tags from the dynamically generated list, users can establish the foundation for their CSS rules. This feature simplifies the rule creation process, as users can easily identify and designate the specific elements they wish to modify visually. The inclusion of a wide variety of HTML tags facilitates fine-grained customization, allowing users to tailor the appearance of the play according to their preferences.

A key component of CSS rule creation is the ability to assign desired colors to the selected HTML tags. The application provides users with an intuitive color selection interface, enabling them to choose from a spectrum of colors, predefined color palettes, or even input custom color values. This flexibility allows users to personalize the visual aspects of the play and create a customized color scheme that aligns with their aesthetic sensibilities.

By assigning colors to the selected HTML tags, users can effortlessly modify the appearance of different elements within the play. For instance, they can assign a vibrant and attention-grabbing color to headings to make them stand out or opt for a soothing and subtle color palette for the paragraphs to create a calming reading experience. This granular control over color selection empowers users to create a visually cohesive and pleasing presentation that resonates with their individual preferences.

In addition to selecting predefined HTML tags and assigning colors, the application takes customization a step further by allowing users to apply their own custom CSS rules to the HTML document. This advanced functionality grants users complete freedom to define and implement their desired styles, transcending the limitations of predefined tag-color combinations.

By leveraging custom CSS rules, users can unleash their creativity and apply unique visual styles to the play. They can manipulate various CSS properties such as font styles, background colors, margins and more to create a truly bespoke reading experience. This flexibility enables users to tailor the visual presentation to suit their specific requirements, whether it be

emphasizing certain elements, experimenting with typography, or crafting a visually striking and immersive environment for the reader.

The CSS rule creation feature offers users a multitude of benefits, enhancing the visual aesthetics and customization options within the application. By dynamically selecting HTML tags and assigning colors, users can effortlessly modify the appearance of different elements within the play. This visual customization enriches the reading experience, making the text more visually appealing and engaging.

The inclusion of custom CSS rule application takes customization to a whole new level. By enabling users to define their own styles and apply them to the HTML document, the application empowers them to create a unique reading environment that aligns with their personal preferences. This level of customization fosters a sense of ownership and creativity, allowing users to craft a visual presentation that enhances their immersion in the play and facilitates a more enjoyable and personalized reading experience.

Accessibility and readability are further enhanced by the ability to create CSS rules and customize the visual aspects of the game. Users can adjust font styles, color contrasts and other visual parameters to optimize readability, ensuring that text is comfortably readable for people with different preferences and visual needs.

The CSS rule creation functionality within the application offers users a powerful tool for visual customization and personalization. By dynamically selecting HTML tags, assigning colors and applying custom CSS rules, users can transform the visual presentation of the play to suit their preferences. This feature enhances the reading experience, fosters creativity and promotes accessibility, ultimately empowering users to engage with the text in a visually appealing and immersive manner.

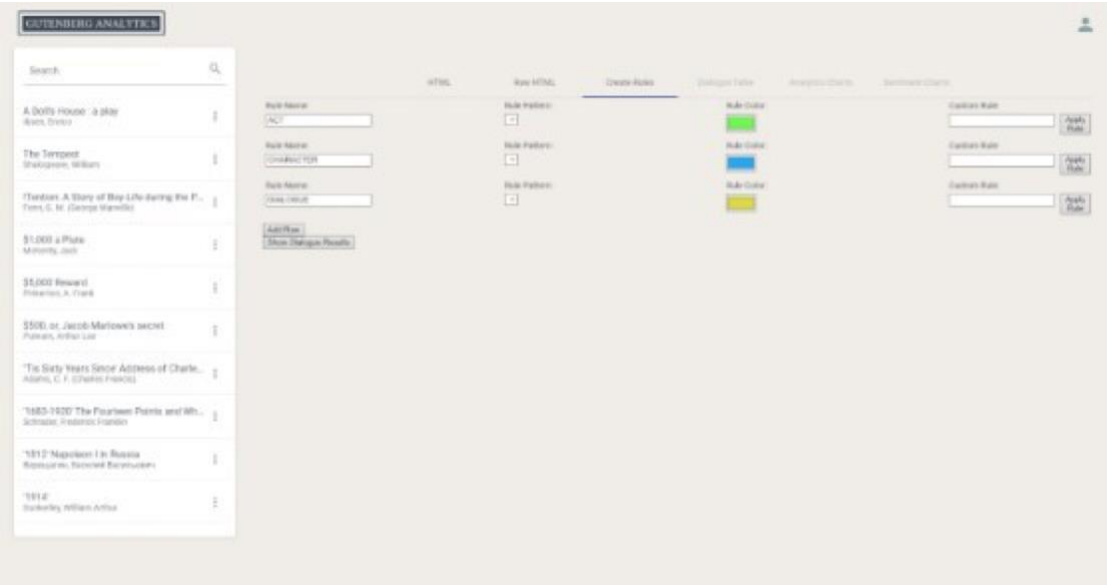

Figure 13 - Create Rules Default Options

Figure 14 - Create Rules Custom Rule

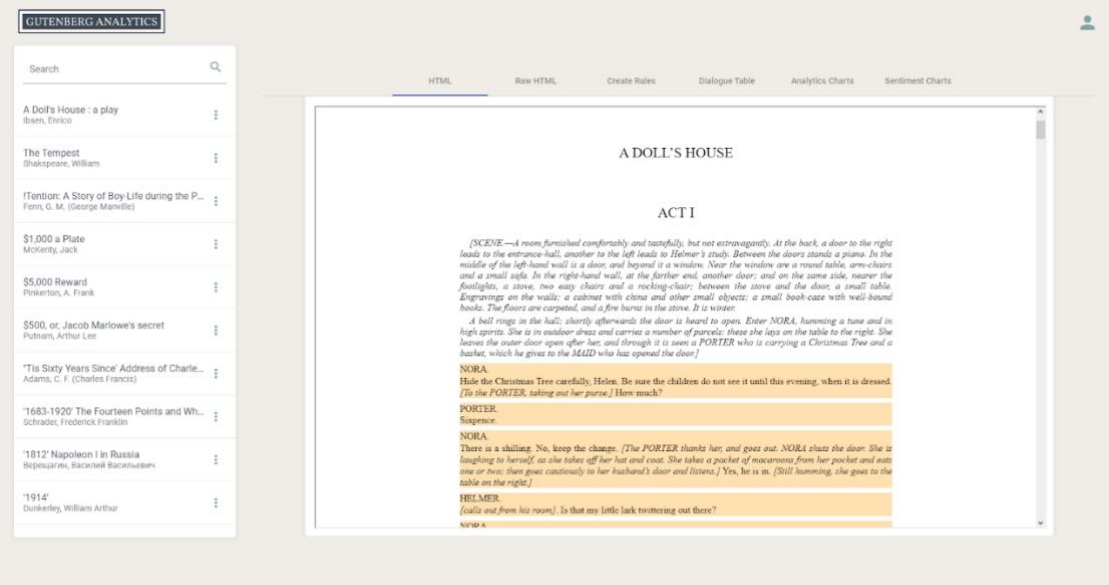

Figure 15 - Rules In Html View

3.5.6 Table View and Interactive Modifications

When users apply the CSS rules, the results are applied at the “Dialog Table” tab. This table-view feature offers a comprehensive representation of the play, presenting data in a structured and organized manner. Users can easily navigate through the play, view dialogs, acts and characters and gain insights into the overall structure and composition. Additionally, the application facilitates interactive modifications, allowing users to make changes to dialogs, acts, or characters within the play. This functionality empowers users to perform targeted analysis, apply filters and dive deeper into specific aspects of the play. With the combination of the Table View and Interactive Modifications, users can navigate, analyze and interact with

plays in a seamless and engaging manner, unlocking new perspectives and enhancing their understanding of the literary.

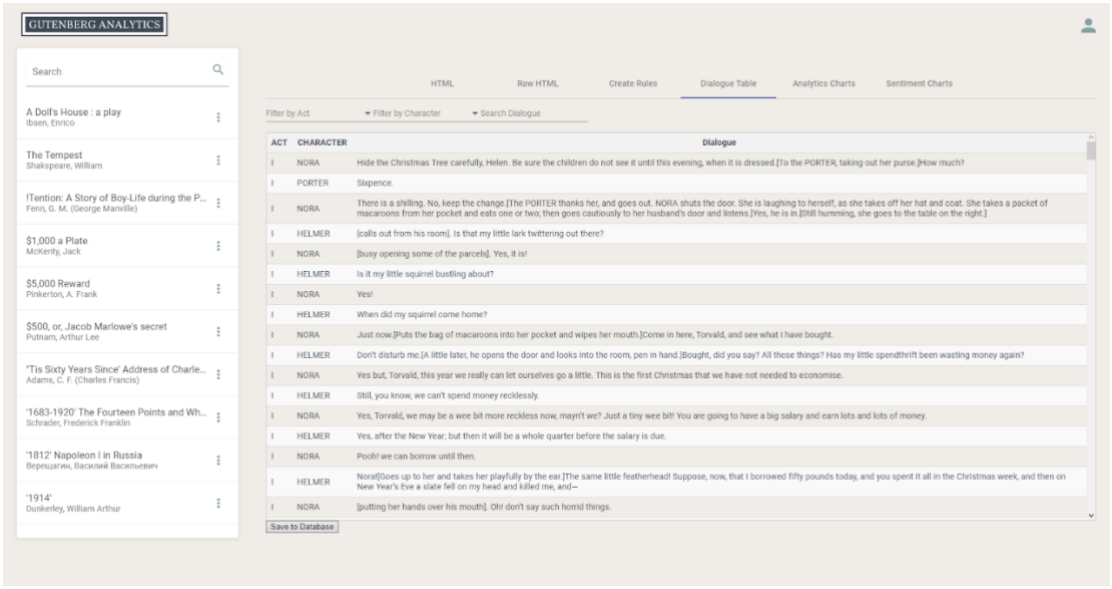

Figure 16 - Dialogues Table

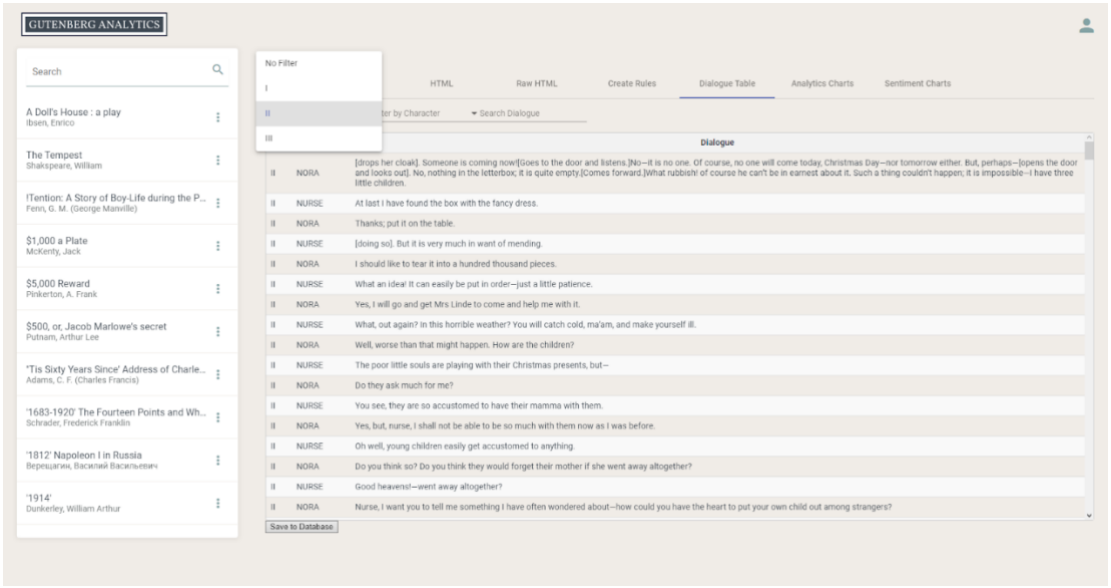

Figure 17 - Act Filtering

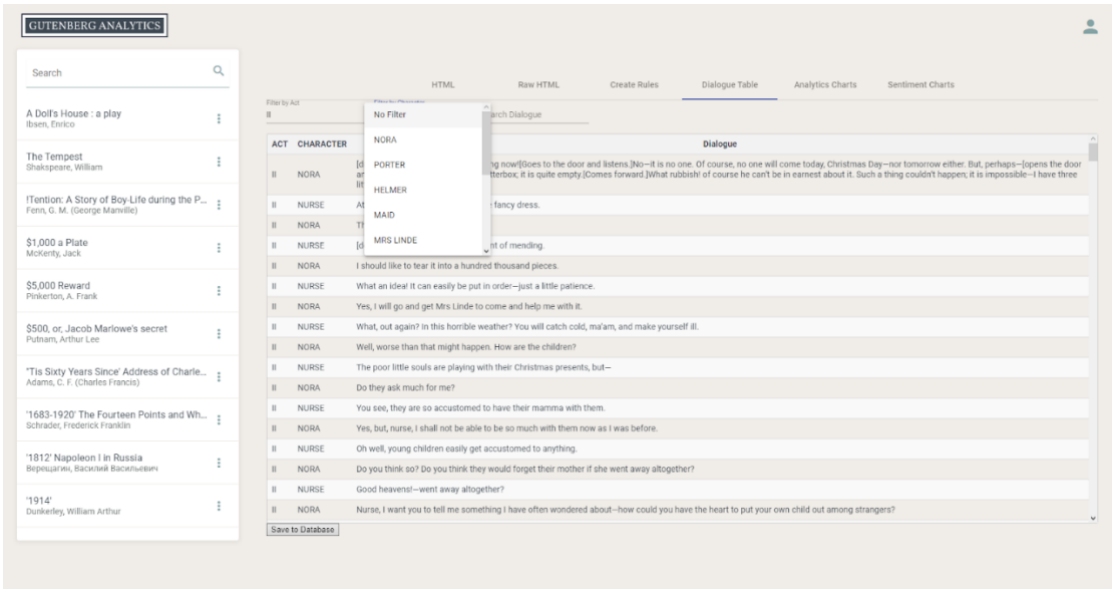

Figure 18 - Character Filtering

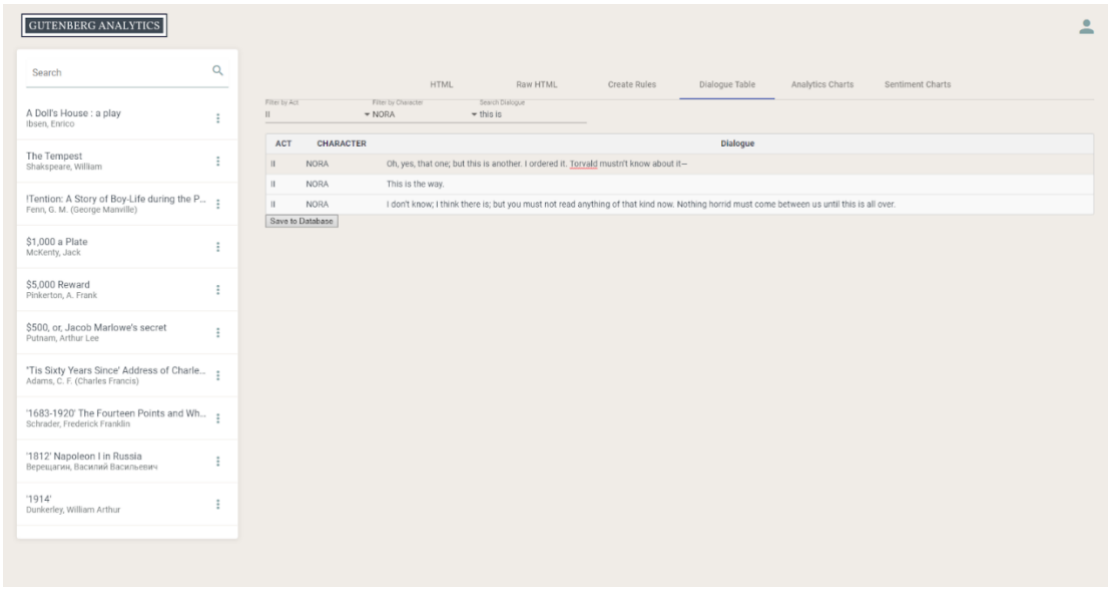

Figure 19 - Text Filtering

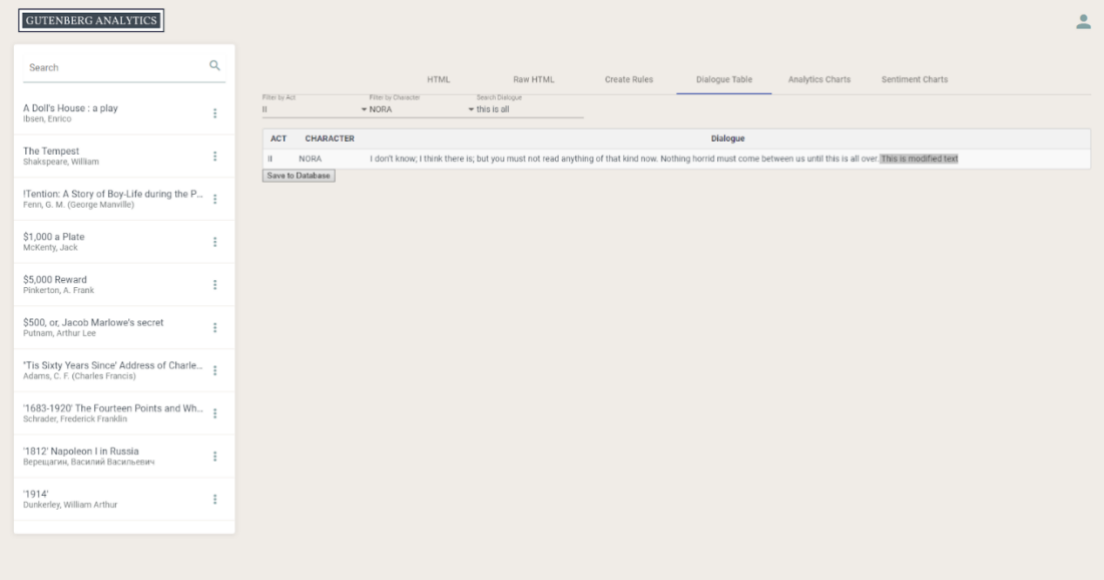

Figure 20 - Text Update

3.5.7 Database Integration

In this chapter, we examine the crucial aspect of database integration within the application. Recognizing the significance of storing and retrieving processed data for future reference, the application offers users the capability to save analyzed text in a database. This feature ensures the seamless accessibility of previously analyzed plays, allowing users to retrieve and review the processed data at their convenience. This chapter provides an in-depth exploration of the database integration functionality, emphasizing its value in facilitating data management and analysis within the application.

One of the key functionalities of the application is the ability to store analyzed text in a robust and efficient database system. When users analyze a play, the processed data, including character breakdowns, dialog details, word counts, sentiment analysis results and other relevant information, can be seamlessly stored in the database. By doing so, users can ensure that their hard work and analytical insights are preserved for future reference.

The database serves as a repository for the analyzed text, allowing users to organize and manage their data effectively. The stored information retains its structural integrity, ensuring that the various components of the analysis remain interconnected and accessible. This ensures that users can retrieve and review the analyzed plays comprehensively, enabling them to revisit their findings, make comparisons and derive new insights based on their prior analysis.

The database integration feature goes beyond just storing the analyzed text. It enables users to retrieve the processed data effortlessly whenever needed. By accessing the database, users can retrieve previously analyzed plays, facilitating continuity in their analysis and research endeavors.

When users initiate the retrieval process, the database retrieves the stored data associated with the desired play. This includes all the relevant information generated during the analysis, such as character breakdowns, dialog details, word counts and sentiment analysis results. By seamlessly retrieving this data, users can access a comprehensive overview of the analyzed play, enabling them to delve into the intricacies of their previous analysis.

The retrieval process ensures that users can effortlessly review their past work, compare different plays and draw meaningful connections and conclusions based on their accumulated knowledge. This enhances the overall analytical experience, as users can build upon their previous insights, refine their interpretations and contribute to a more comprehensive understanding of the plays they analyze.

The integration of a database within the application offers numerous benefits and holds significant importance for users. Firstly, the ability to store analyzed text in a database ensures data preservation and prevents the loss of valuable analytical work. By securely storing the processed data, users can rest assured that their efforts and insights are safeguarded and readily available for future use.

Secondly, the seamless retrieval of analyzed plays from the database enhances the efficiency and continuity of users' analysis. Rather than starting from nothing with each new analysis, users can conveniently access their previously analyzed plays, allowing for a more informed and comprehensive approach to their research. The ability to review and compare multiple plays fosters a deeper understanding of patterns, themes and stylistic elements across different works, enriching the overall analytical experience.

Furthermore, the database integration feature promotes effective data management. By centralizing the storage of analyzed plays, users can easily organize, categorize and search for specific plays or data points within the database. This streamlines the data retrieval process, enabling users to access the desired information promptly and efficiently.

Lastly, the database integration feature enhances collaboration and knowledge sharing among users. By enabling multiple users to access and contribute to the database, the application fosters a collaborative environment where researchers, scholars, or enthusiasts can share their insights, annotations and interpretations of various plays. This collaborative approach facilitates the exchange of ideas, encourages scholarly discourse and promotes the advancement of knowledge in the field of literary analysis.

Finally, by incorporating a database into the application, we can offer a robust and efficient way to store and retrieve parsed texts. By preserving the processed data and ensuring its seamless accessibility, the database integration feature enhances data management, promotes continuity in analysis and fosters collaboration and knowledge sharing. Ultimately, this

functionality enriches the user experience, facilitating a more comprehensive and insightful exploration of the plays and contributing to the advancement of literary analysis.

3.5.8 Sentiment Analysis and Analytics

One of the goals of the developed app is to provide users with a comprehensive understanding of the emotional dynamics present throughout the play. By analyzing various aspects such as characters, dialogs, acts and timelines, we aim to offer valuable insights that can enrich the interpretation and evaluation of the play. This chapter presents a detailed overview of the analytics features provided by the application, including the breakdown of characters, dialogs and words per dialog, as well as sentiment analysis at various levels.

The application prides itself on delivering a wealth of analytics to our users. To start, we provide a meticulous breakdown of the characters in the play. Understanding the roles and significance of each character is essential for a thorough analysis and our analytics feature ensures that users have access to this vital information. By examining the number of dialogs and words per dialog, users can gain a deeper understanding of the distribution of interactions and the level of verbosity among the characters.

Moreover, our analysis goes beyond individual characters to encompass the entire play. We provide users with a breakdown of the number of dialogs and words for each role and act. This level of granularity allows users to examine the distribution of dialogs and words across distinct roles, offering insights into the prominence and engagement of various characters throughout the acts. By understanding the patterns and variances in dialog distribution, users can identify pivotal moments and discern the underlying narrative structure of the play.

Sentiment analysis plays a crucial role in uncovering the emotional dynamics within the play. The application conducts sentiment analysis at multiple levels, enabling users to explore and interpret the emotional nuances present throughout the text.

At the dialog level, sentiment analysis allows users to evaluate the emotional tone of individual interactions. By analyzing the sentiment expressed in each dialog, the application provides users with valuable insights into the characters' emotions, attitudes and intentions. This deep dive into the micro-level emotions expressed in dialogs can help shed light on the intricate relationships and conflicts between characters.

Moving beyond individual dialogs, our sentiment analysis extends to the act level. By analyzing the sentiments across acts, users can identify overarching emotional themes and shifts in the play. This macro-level analysis aids in understanding the emotional trajectory of the storyline, highlighting moments of tension, resolution, or emotional climax.

Character-level insights are also included in our sentiment analysis. By examining the sentiments expressed by each character throughout the play, users can gain a comprehensive

understanding of the emotional journey undertaken by individual characters. This analysis facilitates the exploration of character development, emotional arcs and the impact of specific characters on the play's overall emotional landscape.

Finally, the application also provides sentiment analysis at the timeline level. By examining the sentiments expressed over time, users can discern temporal patterns in emotional dynamics. This analysis helps identify the evolution of emotions, the pacing of the play and the ebb and flow of sentiment across different acts and scenes.

The comprehensive analytics and sentiment analysis provided by the application offer users a multitude of benefits. By leveraging the breakdown of characters, dialogs and words per dialog, users can gain a deeper understanding of the distribution of interactions and the level of verbosity within the play. This knowledge empowers users to analyze the prominence of characters, explore their relationships and identify key moments in the narrative.

Moreover, the sentiment analysis conducted at various levels allows users to unlock the emotional intricacies within the play. By examining sentiments expressed in dialogs, acts, characters and timelines, users can unravel the emotional landscape of the text. These insights enable a richer interpretation of the play's themes, character motivations and narrative arcs. Furthermore, the ability to track emotional dynamics over time provides users with a unique perspective on the overall emotional trajectory of the play, enhancing their appreciation and analysis of its structure and impact.

To conclude, the incorporation of comprehensive analytics and sentiment analysis into the application empowers users to engage deeply with the emotional tapestry woven within the play. By unraveling the intricate emotional nuances, uncovering discernible patterns and acquiring valuable insights, users gain an enriched understanding and evaluation of the play's underlying themes, multifaceted characters and intricately crafted narrative. Ultimately, the application serves as an indispensable tool for unlocking the profound emotional dimensions of the play, facilitating a more profound and holistic exploration of its artistic and literary merits.

3.5.9 Analytics Charts

The analytics charts feature allows users to explore and visualize various metrics related to a play that has been processed and saved in the database. These charts offer valuable insights into various aspects of the play's structure and content, providing a deeper understanding of its characteristics. The following are the available analytics charts available:

- Number of dialogues per character

This chart illustrates the distribution of dialogues among different characters in the play. By displaying the frequency of dialogues for each character, users can gain insights into the prominence and involvement of various characters in the storytelling.

- Number of words per character

This chart displays the word count attributed to each character in the play. It offers a comparison of the linguistic contributions of different characters, shedding light on their significance and the depth of their dialogue.

- Number of dialogues per act

This chart presents the distribution of dialogues across the acts of the play. Users can observe how dialogues are distributed throughout the play's structure, potentially revealing patterns, variations, or focal points within each act.

- Number of words per act

This chart provides a visualization of the word count for each act in the play. By examining the word count per act, users can identify variations in the length or intensity of different sections, potentially indicating shifts in plot, character development, or thematic focus.

- Number of words per dialogue

This chart focuses on the length of individual dialogues in terms of word count. It allows users to analyze the brevity or verbosity of dialogues, providing insights into the pacing, style, or depth of conversations within the play.

- Number of sentences per dialogue

This chart explores the composition of dialogues by highlighting the number of sentences in each dialogue. Users can examine the complexity or simplicity of dialogues, potentially identifying patterns or variations in the structure and flow of conversations.

Bellow we display the code that is responsible for generating the analytics charts:

```python
@app.get("/book/{book_id}/analytics")
def get_book_analytics(book_id: int):

    # Open a session and query the database for the book with the given book_id
    db = SessionLocal()
    book = db.query(BookWithDialogue).filter(BookWithDialogue.book_id == book_id).all()

    # Initialize a list to store the combined information for each dialogue
    dialogues_info = []

    for dialogue in book:
        # Calculate the number of words for the dialogue
        num_words = len(dialogue.dialogue.split())

        sentences = re.split(r'\[[^\]]*\]|\.\s*', dialogue.dialogue)
        num_sentences = len([sentence for sentence in sentences if sentence.strip()])



        # Perform sentiment analysis for the dialogue
        sentiment_score = TextBlob(dialogue.dialogue).sentiment.polarity
        if sentiment_score > 0:
            sentiment = "positive"
        elif sentiment_score < 0:
            sentiment = "negative"
        else:
            sentiment = "neutral"

        # Create a dictionary with the combined information
        dialogue_info = {
            "line_number": dialogue.line_number,
            "num_words": num_words,
            "sentiment": sentiment,
            "num_sentences": num_sentences,
            "dialogue": dialogue.dialogue
        }

        # Append the dialogue information to the list
        dialogues_info.append(dialogue_info)

    # Calculate the other statistics
    num_dialogues = len(book)
```

Figure 21 - Code for Generating Analytics (part one)

```python
# Calculate the other statistics
num_dialogues = len(book)
characters = {dialogue.character for dialogue in book}
num_dialogues_per_character = {character: sum(1 for dialogue in book if dialogue.character == character) for character in characters}
num_words_per_character = {character: sum(len(dialogue.dialogue.split()) for dialogue in book if dialogue.character == character) for character in characters}
acts = {dialogue.act for dialogue in book}
num_dialogues_per_act = {act: sum(1 for dialogue in book if dialogue.act == act) for act in acts}
num_words_per_act = {act: sum(len(dialogue.dialogue.split()) for dialogue in book if dialogue.act == act) for act in acts}

# Transform act numbers from Roman numerals to regular numerals
num_dialogues_per_act = {roman.fromRoman(act): count for act, count in num_dialogues_per_act.items()}
num_words_per_act = {roman.fromRoman(act): count for act, count in num_words_per_act.items()}

# Close the session
db.close()

# Return the statistics and dialogues information as a JSON response
return {
    "num_dialogues": num_dialogues,
    "num_dialogues_per_character": num_dialogues_per_character,
    "num_words_per_character": num_words_per_character,
    "num_dialogues_per_act": num_dialogues_per_act,
    "num_words_per_act": num_words_per_act,
    "dialogues_info": dialogues_info
}
```

Figure 22 - Code for Generating Analytics (part two)

The result is displayed in the frontend like this:

Figure 23 - Analytics Charts (top)

Figure 24 - Analytics Charts (down)

These analytics charts offer users a comprehensive visual representation of the play's characteristics, enabling them to analyze and interpret various aspects of its structure and content. By examining the distribution of dialogues and words per character, act and dialogue, users can gain insights into the play's narrative dynamics, character significance and linguistic features. These charts provide a valuable tool for researchers, scholars, or enthusiasts seeking a deeper understanding of the play's composition and its implications for the overall storytelling experience.

3.5.10 Sentiment Timeline

The sentiment timeline feature offers users the ability to delve deeper into the emotional journey of a saved book. When users access this feature, they are presented with a comprehensive view of the book's sentiment analysis over time. This analysis is conducted using the TextBlob library, which assigns a sentiment score to each sentence within the book.

The sentiment timeline chart provides users with a visual representation of the sentiment scores assigned to each sentence. The scores, ranging from -1 to 1, are plotted along the timeline, allowing users to track the fluctuation of sentiments throughout the book. Each sentiment score is associated with its respective sentence, providing users with a contextual understanding of the book's emotional trajectory.

Furthermore, the sentiment timeline feature offers users the flexibility to customize their viewing experience. Users can adjust the step or number of data points displayed on the graph. This customization option allows users to focus on specific sections of the book or zoom in on

particular moments of interest. By selecting a smaller step or fewer data points, users can obtain a more granular view of the sentiment progression within the book.

For example, a user analyzing a novel's sentiment timeline may choose to display data points at a larger step, providing a broader overview of the book's emotional arc. This approach allows them to identify major shifts in sentiment across different chapters or sections. On the other hand, if a user wants to closely examine the sentiment evolution within a specific chapter or scene, they can decrease the step or increase the number of data points, providing a more detailed sentiment analysis for that section.

The sentiment timeline feature offers valuable insights for both casual readers and researchers alike. Casual readers can gain a deeper appreciation for the emotional nuances present within a book, exploring how sentiment varies across different sections and chapters. They can identify pivotal moments, character developments, or thematic shifts that evoke specific emotional responses.

Researchers, on the other hand, can utilize the sentiment timeline feature to conduct in-depth analyses of a book's emotional progression. By examining the sentiment scores in conjunction with the corresponding sentences, researchers can uncover patterns, correlations, or anomalies in the book's sentiment distribution. This information can be utilized to support literary interpretations, study the impact of emotional elements on reader engagement, or explore the relationship between sentiment and other literary aspects.

In conclusion, the sentiment timeline feature provides users with a comprehensive understanding of the emotional trajectory of a saved book. Using sentiment analysis and the TextBlob library, users can explore how sentiments evolve over time, sentence by sentence. The customization options further enhance the user experience, allowing users to adjust the step or number of data points to focus on specific sections of interest. Whether for casual reading or scholarly research, the sentiment timeline feature offers a valuable tool to analyze and appreciate the emotional dimensions within literature.

Bellow we display the code that is responsible for generating the sentiment timeline chart:

```python
@app.get("/book/{book_id}/sentiment-timeline")
def get_sentiment_timeline(book_id: int) -> Dict[str, List[Dict[str, float]]]:

    # Open a session and query the database for the book with the given book_id
    db = SessionLocal()
    book = db.query(BookWithDialogue)\
        .filter(BookWithDialogue.book_id == book_id)\
        .order_by((BookWithDialogue.line_number))\
        .all()

    # Initialize a list to store the sentiment timeline data
    sentiment_timeline = []

    for dialogue in book:
        # Perform sentiment analysis for the dialogue
        sentiment_score = TextBlob(dialogue.dialogue).sentiment.polarity

        if sentiment_score > 0:
            sentiment = "positive"
        elif sentiment_score < 0:
            sentiment = "negative"
        else:
            sentiment = "neutral"

        # Create a dictionary with the line number, sentiment score, character, and act
        sentiment_data = {
            "dialogue": dialogue.dialogue,
            "line_number": dialogue.line_number,
            "sentiment_score": sentiment_score,
            "sentiment": sentiment,
            "character": dialogue.character,
            "act": dialogue.act
        }

        # Append the sentiment data to the timeline
        sentiment_timeline.append(sentiment_data)

    # Close the session
    db.close()

    # Return the sentiment timeline data as a JSON response
    return {"sentiment_timeline": sentiment_timeline}
```

Figure 25 - Code for Generating Sentiment Timeline

The result is displayed in the frontend like this:

Figure 26 - Sentiment Timeline Chart

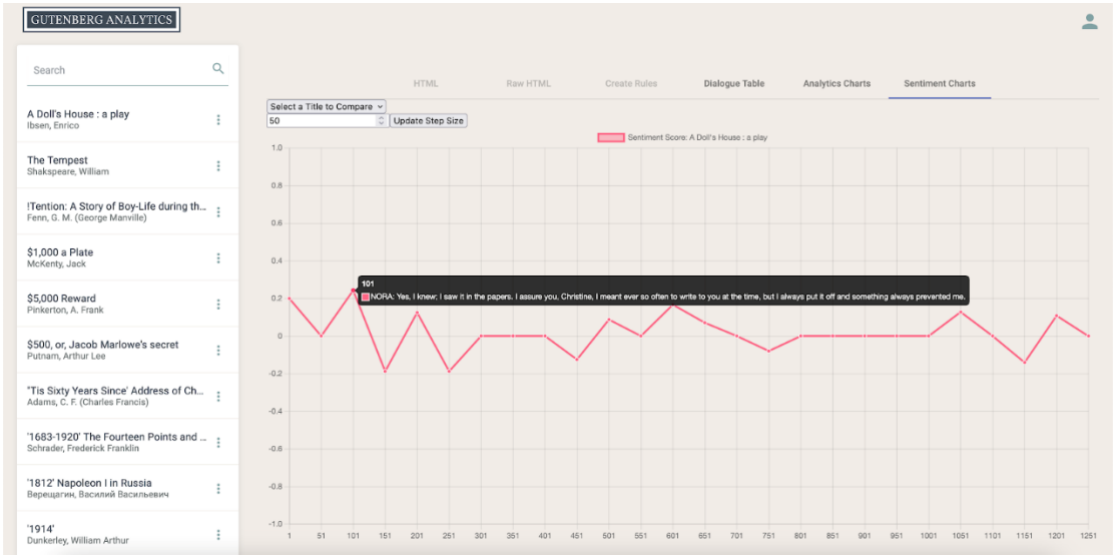

Figure 27 - Sentiment Timeline Chart With Custom Step Size

3.5.11 Chart Comparison

In the analytics charts or sentiment charts tab, users are provided with a convenient feature—a dropdown menu that displays all the saved pre-processed books available in the database. This dropdown menu serves as a navigation tool, allowing users to select a specific title of interest. When a user chooses a book title from the dropdown menu, the analytics associated with that book are dynamically generated and displayed in the existing charts.

This functionality provides users with the ability to compare the analytics of two different books. By selecting two book titles from the dropdown menu, users can view and analyze the charts side by side, gaining insights into the similarities and differences between the two literary works.

The comparison feature offers a valuable opportunity for users to explore the nuances of multiple books simultaneously. They can examine various aspects such as sentiment analysis, key themes, character development, or any other relevant analytics. By observing the visual representation of the analytics, users can quickly identify patterns, trends and potential relationships between different books.

For example, a user interested in studying the sentiment analysis of two classic novels could select the book titles from the dropdown menu and view the sentiment charts side by side. This comparison enables them to analyze the emotional trajectory of each book, identifying variations in positive and negative sentiment throughout the plot. They can also observe if there are any significant differences in the overall emotional tone or if certain themes evoke similar emotional responses in both books.

Additionally, the comparison feature allows users to assess the effectiveness of literature analytics methodologies and tools. By examining how different books are represented in the charts, users can evaluate the accuracy, consistency and reliability of the analytics generated by the system. This assessment contributes to refining and improving the literature analytics platform, ensuring its ability to provide valuable and insightful information to users.

Furthermore, the ability to compare two books in the charts provides researchers and scholars with a powerful tool for conducting comparative literary analysis. They can explore the similarities and differences between books from different genres, time periods, or authors, gaining a deeper understanding of the literary landscape and its evolution over time. This feature empowers users to delve into complex research questions and make meaningful connections between multiple works.

In summary, the dropdown menu in the analytics charts or sentiment charts tab offers users the flexibility to select and compare different books. This functionality enhances the user experience by enabling a comprehensive exploration of the analytics, facilitating comparative analysis and contributing to the ongoing improvement of literature analytics methodologies and tools. Users can delve into the intricacies of multiple books, uncovering valuable insights and fostering a deeper appreciation for the world of literature.

.

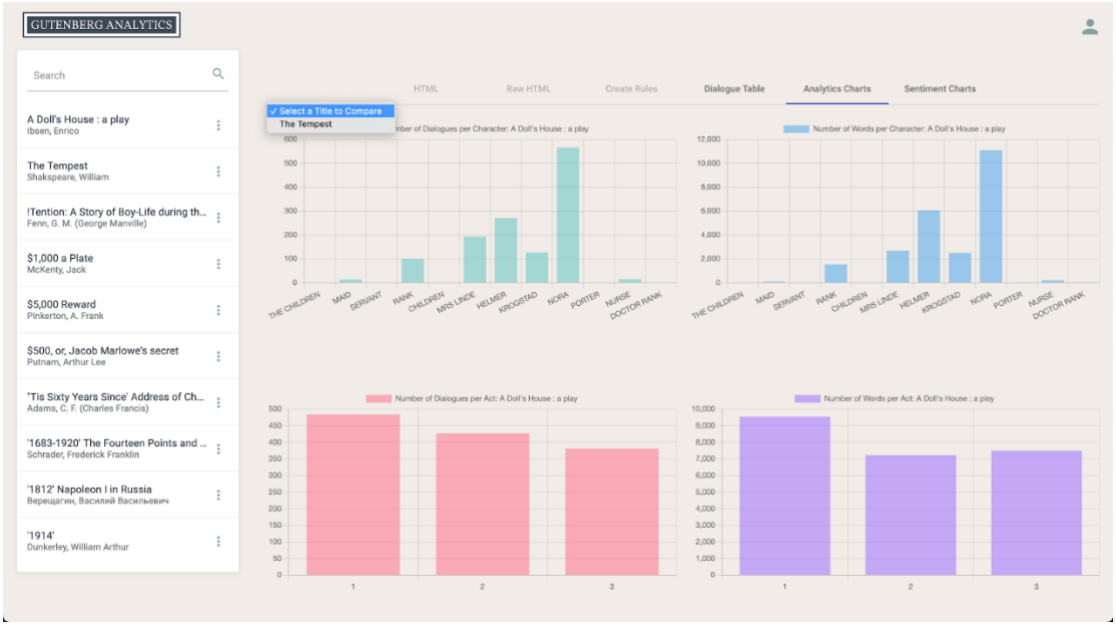

Figure 28 - Analytics Charts Comparison Dropdown Menu

Figure 29 - Analytics Charts With Comparison (top)

Figure 30Analytics Charts With Comparison (down)

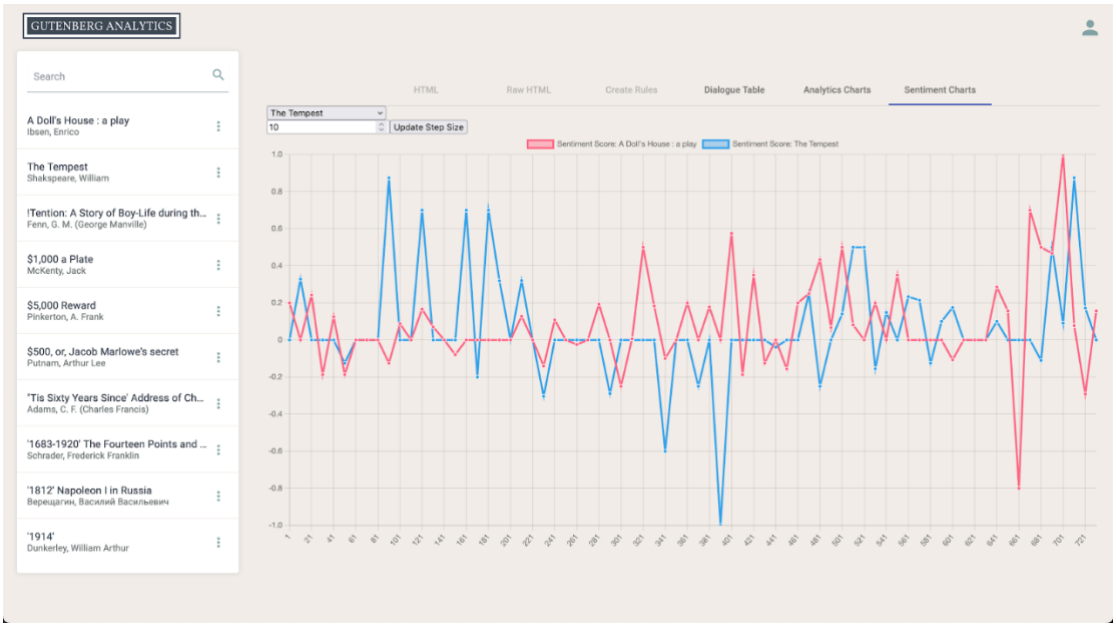

Figure 31 - Sentiment Timeline With Comparison

3.6 Methodology

3.6.1 Development Approach

For the development of the web app, we opted to utilize the Angular framework for the frontend and FastAPI framework for the backend. The choice of these frameworks was based on their robustness, scalability and support within the developer community. Our coding style focused on readability and maintainability, ensuring that the codebase can be easily understood and extended in the future. We followed an agile development approach, dividing the project into different sprints for backend, frontend and database changes, allowing us to efficiently manage the development process.

3.6.2 Requirements Gathering

The initial requirements for the web application were well-defined by the professor in the thesis description. However, as the development progressed, we engaged in discussions during sprint planning and team meetings to refine and modify certain requirements or incorporate new ones based on feedback and evolving project needs. This iterative approach ensured that the application met the desired functionalities and user expectations.

3.6.3 Design and Prototyping

We began the development process by creating mockups and wireframes of the application's user interface. These designs were reviewed and refined based on considerations of user experience, ensuring that the application's layout and functionality were intuitive and user-friendly. Subsequently, we proceeded to implement the code, translating the design concepts into a fully functional web app.

3.6.4 Development Process

Our development process involved adopting coding practices that promoted code quality, maintainability and collaboration. We decided to use the popular hosting platform "GitHub", to track changes and facilitate collaboration among team members. Communication and coordination were facilitated with freely available apps for both project management and communication. Development tasks were organized and prioritized using agile methodologies, allowing us to effectively manage the project timeline and deliver the application's features in incremental iterations.

3.6.5 Data Collection and Preprocessing

To populate the application with plays and books, we utilized data collection techniques, including scraping data from Project Gutenberg and leveraging API integrations where available. The collected data underwent preprocessing to extract relevant information and ensure its compatibility with the application's analysis features. This involved cleaning the data, removing unnecessary elements and organizing it in a structured format that could be easily accessed and analyzed.

3.6.6 User Testing

To ensure the functionality and usability of the app, we conducted extensive user testing. We tested the application's features and functionalities using a variety of plays, evaluating the accuracy of metadata extraction and the effectiveness of the sentiment analysis capabilities. It is important to note that the extraction of metadata relies on the application of CSS rules and in

some cases, users may need some experience or familiarity with CSS to write custom rules for optimal results.

3.6.7 Ethical Considerations

In terms of ethical considerations, the application utilizes publicly available data from Project Gutenberg, which is openly accessible and free to use. We do not retain any user-related data beyond the required username, email and password for the authentication process. Data privacy and security measures were implemented to safeguard user information and ensure compliance with ethical standards throughout the development and deployment of the web app.

3.7 Challenges and Limitations

The development of the web application presented several challenges and limitations that needed to be addressed during the project. Firstly, one of the main challenges was the variability and inconsistency of data within the Project Gutenberg collection. The plays and books available in the collection often varied in formatting, structure and metadata availability, which required extensive data preprocessing to ensure accurate analysis and extraction of relevant information. Additionally, the reliance on CSS rules for metadata extraction posed a challenge as users with limited CSS knowledge may face difficulties in creating custom rules for specific plays. Efforts were made to provide clear instructions and guidance to users, but the learning curve associated with CSS rule creation remained a limitation.

Another challenge encountered during the development process was the need for efficient data storage and retrieval. As the application allows users to save and analyze multiple plays, ensuring optimal performance and scalability of the database became crucial. Handling large volumes of data and enabling quick access to the analyzed text required careful database design and optimization.

Furthermore, the application's functionality and analysis capabilities were limited to the content available in the Project Gutenberg collection. While the collection offers a vast array of plays and books, it may not cover all literary works or genres. This limitation restricts the application's applicability to the specific content provided by Project Gutenberg. Future enhancements could explore integration with other literary sources or expand the collection to encompass a broader range of plays and books, increasing the application's versatility and usefulness.

Overall, despite these challenges and limitations, the web application offers valuable features and functionalities for metadata extraction, sentiment analysis and analytics of plays sourced from Project Gutenberg. The project team worked diligently to address these challenges and

mitigate the limitations to provide users with a robust and user-friendly app. Ongoing efforts in refining the application's performance, expanding the content collection and enhancing the user experience will further overcome these limitations and contribute to the continued success of the web application in facilitating literature analysis and exploration.

4 Evaluation

4.1 Introduction

The evaluation process holds significant importance in assessing the effectiveness and impact of various aspects related to literature analytics. This chapter focuses on the evaluation conducted through Google Forms, an online survey tool widely utilized for gathering feedback and insights. By employing Google Forms, researchers can collect valuable information from participants, allowing for a comprehensive assessment of literature analytics projects, methodologies and tools. This chapter outlines the key components and considerations involved in conducting evaluations via Google Forms.

4.2 Purpose and Objectives of Evaluation

Evaluation serves a specific purpose in assessing the performance and outcomes of literature analytics endeavors. Defining the evaluation goals is crucial to ensure a clear understanding of what needs to be assessed. Identifying the research questions to be addressed through the evaluation helps guide the process and ensures that the results contribute to the overall literature analytics project. The evaluation aims to gather participant feedback, which can provide insights into the effectiveness, usability and impact of literature analytics projects. By understanding the experiences and perspectives of participants, improvements and future developments in the field can be informed.

4.3 Designing the Evaluation Survey

Designing a well-structured and comprehensive survey is essential for gathering relevant data. The selection of evaluation criteria plays a key role in assessing literature analytics tools and methodologies. Criteria such as usability, functionality, accuracy and interpretability are often considered. These criteria should align with the goals and objectives of the literature analytics project. Constructing the evaluation survey using Google Forms involves careful consideration of question types, including multiple-choice, Likert scale and open-ended questions. Demographic and background questions can also be included to gather participant information.

Ensuring the validity and reliability of the evaluation survey is crucial for obtaining accurate and meaningful results. Attention should be given to factors such as question wording, potential response bias and sampling techniques. Pilot testing the survey and refining it based on feedback before distribution can enhance its validity and reliability.

4.4 Administering the Evaluation Survey

Participant recruitment is a vital step in the evaluation process. Strategies for recruiting participants should be implemented, considering the target demographics and the desired sample size. Ethical considerations, such as informed consent, should be addressed when recruiting participants for the evaluation. The distribution of the evaluation survey can be done through various methods, such as email invitations, social media platforms, or dedicated website links. Maximizing response rates can be achieved through reminders and incentives. Ensuring anonymity and confidentiality of participant responses is essential to encourage honest and unbiased feedback.

4.5 Data Collection and Analysis

Data collection procedures involve retrieving and storing survey responses securely. Proper data management techniques should be employed to maintain data integrity and security throughout the process. The collected data can be analyzed using various approaches. Quantitative data can be interpreted through statistical analysis methods, providing insights into participant ratings and preferences. Qualitative data analysis techniques can be applied to analyze open-ended responses, identifying common themes and capturing detailed feedback.

4.6 Interpretation and Application of Evaluation Findings

The interpretation of evaluation results plays a crucial role in understanding the strengths, weaknesses and areas for improvement within the literature analytics project. By presenting and interpreting the evaluation survey findings, researchers gain valuable insights into the performance and impact of the project. These findings can be applied to make informed decisions and guide future developments in literature analytics. Incorporating participant feedback into refining methodologies, tools and projects is essential for continuous improvement. The broader implications of evaluation findings for the field of literature analytics should also be considered, ensuring that the evaluation contributes to the advancement and relevance of the field.

4.7 Survey Results

This chapter presents the results of the survey conducted to gather feedback on the web application developed for this master thesis. The survey consisted of six questions aimed at assessing the usability and effectiveness of the web application. A total of 20 participants responded to the survey.

4.7.1 Was it easy to navigate and use the web application?

100% of the respondents answered "Yes." This indicates that all participants found the web application easy to navigate and use.

4.7.2 Did you find the search feature helpful?

100% of the respondents answered "Yes." This suggests that all participants found the search feature of the web application helpful.

4.7.3 Did the sentiment analysis feature provide valuable insights into the plays?

100% of the respondents answered "Yes." This indicates that all participants found the sentiment analysis feature of the web application to be valuable in providing insights into the plays.

4.7.4 Did the provided analytics enhance your understanding of the plays?

95% of the respondents answered "Yes," while 5% answered "No." This suggests that most of the participants (95%) found the provided analytics to enhance their understanding of the plays, while a small portion (5%) did not feel the same way.

4.7.5 Were you able to effectively organize and manage your selections using the personalized reading lists?

100% of the respondents answered "Yes." This indicates that all participants were able to effectively organize and manage their selections using the personalized reading lists feature of the web application.

4.7.6 Did you find the web application useful for conducting literature analytics?

100% of the respondents answered "Yes." This suggests that all participants found the web application useful for conducting literature analytics.

The survey results demonstrate a high level of satisfaction among the participants regarding the usability and effectiveness of the web application. The positive responses indicate that the web application was easy to navigate, the search feature was helpful, the sentiment analysis feature provided valuable insights, the analytics enhanced understanding for the majority and the personalized reading lists were effective for organizing and managing selections. Furthermore, all participants found the web application useful for conducting literature analytics.

These findings validate the success of the developed web application in meeting the intended goals and requirements outlined in this master thesis.

4.8 Conclusion

Conducting evaluations through Google Forms offers a valuable avenue for collecting participant feedback and assessing the effectiveness of literature analytics projects, methodologies and tools. By carefully designing and administering the evaluation survey, researchers gain comprehensive insights that inform improvements and guide future developments in the field. The evaluation findings contribute to the ongoing refinement and advancement of literature analytics, ensuring its relevance and impact in the realm of literary studies.

5 *Future Work and Improvements*

5.1 Potential enhancements to Metadata extraction

In our current implementation, metadata extraction from HTML documents using CSS rules has proven effective. However, to further advance the capabilities of the application, there are several potential areas for improvement that warrant exploration and investigation. By delving into these areas, we can enhance the accuracy, efficiency and overall effectiveness of the metadata extraction process.

One promising avenue for improvement lies in the exploration of advanced metadata extraction techniques. By incorporating innovative methodologies such as machine learning algorithms and natural language processing (NLP) techniques, we can automate the identification and extraction of relevant metadata elements from HTML documents. Machine learning models can be trained on annotated datasets to recognize patterns and extract metadata based on learned patterns and features. NLP techniques can be employed to analyze the linguistic context and semantics of the text, enabling more precise extraction of metadata. These advanced techniques have the potential to significantly enhance the accuracy and efficiency of the metadata extraction process, ultimately improving the quality of the extracted metadata.

Integrating semantic analysis into the metadata extraction process offers a promising avenue for enhancing the depth and quality of extracted metadata. By analyzing the contextual meaning and relationships between different elements within the HTML documents, we can extract metadata that goes beyond simple surface-level attributes. Semantic technologies, such as ontologies and knowledge graphs, can be leveraged to capture the underlying semantics of the text. This approach enables a more nuanced understanding of the content, allowing for the extraction of richer and more meaningful metadata. By incorporating semantic analysis, we can unlock deeper insights and provide users with metadata that captures the true essence of the plays.

Another avenue for improvement is the enrichment of extracted metadata through the incorporation of external data sources or application programming interfaces (APIs). By

tapping into external resources such as online databases, digital libraries, or ontologies, we can enhance the extracted metadata with additional information. This enrichment can include supplementary details such as author biographies, play summaries, historical context, or related works. By leveraging external data sources, we can augment the extracted metadata, providing users with more comprehensive and insightful information about the plays. This enrichment can significantly enrich the analysis and understanding of the texts, enabling users to delve deeper into the literary context and themes.

5.2 Scalability and Performance Optimization

In the dynamic landscape of software development, scalability and performance optimization are of paramount importance. As the application progresses and garners increased usage, it becomes crucial to ensure that it can handle larger volumes of data efficiently and deliver a seamless user experience. In this chapter, we explore various avenues for future improvements in scalability and performance optimization, focusing on data processing efficiency, caching mechanisms and indexing techniques

As the volume of HTML documents processed by the application grows, optimizing the data processing pipeline becomes essential. One potential approach is to explore parallel processing techniques, where multiple tasks are executed simultaneously, utilizing the available computational resources efficiently. By breaking down the metadata extraction process into smaller, manageable units and executing them concurrently, we can significantly reduce the overall processing time. Furthermore, distributed computing can be leveraged, utilizing multiple machines or cloud-based solutions to distribute the computational workload. This enables the application to scale horizontally, accommodating an ever-increasing number of HTML documents without sacrificing performance.

Caching plays a crucial role in improving the responsiveness and efficiency of the application. By implementing caching mechanisms, we can store previously processed metadata and retrieve it when needed, minimizing redundant computations. This strategy not only reduces the computational burden but also enhances the response time for subsequent requests. Caching can be employed at various levels, such as caching individual metadata elements, intermediate results, or even entire processed documents. The choice of caching strategy depends on the specific requirements of the application and the frequency of data updates. By intelligently managing the cache, we can strike a balance between storage requirements and performance gains, ensuring that users can access metadata swiftly and effectively.

Indexing is another powerful technique that can enhance the retrieval of metadata and improve the overall search and analysis capabilities of the application. By creating appropriate indexes, we can organize the extracted metadata in a structured and optimized manner, allowing for

faster searching and retrieval. Indexing techniques can be applied to various metadata attributes, such as character names, dialogues, acts, or even sentiment scores. By employing efficient indexing algorithms and data structures, we can reduce the time complexity of search operations, enabling users to navigate and analyze plays more effectively. Furthermore, indexing facilitates advanced querying capabilities, such as filtering metadata based on specific criteria or performing complex analytical operations. This empowers users to gain deeper insights and extract meaningful patterns from the vast pool of metadata.

In tandem with scalability and performance optimization, it is vital to evaluate and enhance the system architecture and infrastructure supporting the application. As the user base expands, we must ensure that our infrastructure can handle increased loads and seamlessly scale to meet growing demands. Cloud-based solutions, such as leveraging Infrastructure-as-a-Service (IaaS) or Platform-as-a-Service (PaaS) providers, offer flexibility and scalability. By utilizing the on-demand resources provided by these cloud platforms, we can dynamically adjust the computational resources to match the application's requirements. Additionally, employing load balancing techniques and fault-tolerant design patterns can further enhance the application's resilience and reliability. Continuously monitoring the system's performance, capacity and resource utilization allows us to proactively identify potential bottlenecks and optimize the infrastructure accordingly.

To ensure the effectiveness of our scalability and performance optimization efforts, rigorous performance testing and benchmarking are crucial. By designing comprehensive test scenarios that simulate real-world usage patterns, we can evaluate the application's performance under varying workloads. Performance testing can involve stress testing, where the system is subjected to peak loads to assess its robustness, as well as endurance testing, where the application's performance is evaluated over an extended period to identify any potential performance degradation or resource leaks. Benchmarking against industry standards and best practices allows us to measure the application's performance objectively and identify areas for further improvement. Regularly conducting performance testing and benchmarking enables us to track the impact of optimizations and ensure that the application continues to deliver exceptional performance as it evolves.

5.3 Additional Improvements

Beyond metadata extraction and performance optimization, there are other areas where the application can be further improved. Continuously refining and enhancing the user interface (UI) is crucial to improve the overall user experience. By soliciting user feedback and conducting user studies, we can gain valuable insights into areas that can be enhanced. This feedback can guide us in improving aspects such as navigation, visualizations and workflows,

making the application more intuitive and user-friendly. Implementing user-driven design principles and incorporating modern UI frameworks and best practices can help us create an engaging and aesthetically pleasing interface. Furthermore, offering customization options to users, such as the ability to personalize color schemes, font styles, or layout preferences, can further enhance the reading experience and cater to individual user preferences.

Integrating the application with external tools and platforms commonly used in the field of literary analysis can expand its capabilities and offer users a more comprehensive toolkit. For instance, integration with popular text analysis libraries, collaborative platforms, or citation management tools can enhance the functionality and interoperability of the application. By leveraging the strengths of these external resources, users can seamlessly access additional features, such as advanced linguistic analysis, collaborative annotation, or bibliography management. This integration promotes a more comprehensive approach to literary analysis and allows users to leverage the existing tools and workflows they are already familiar with [5].

Establish a systematic process for monitoring the performance of the application and identifying potential bottlenecks or areas of improvement. Regularly analyzing performance metrics, such as response times and resource utilization, can help identify areas for optimization. By continuously optimizing the application's performance, we can ensure a smooth and efficient user experience.

Summarizing, this chapter explores potential future work and improvements for the application. By focusing on enhancing metadata extraction, optimizing scalability and performance and implementing additional improvements, we can further enhance the functionality, usability and overall value of the application. By addressing these areas, we can continue to provide users with a powerful and efficient tool for metadata extraction and literary analysis.

6 Conclusion

6.1 Summary of Achievements

This thesis has presented a web application that offers a wide range of features and functionalities to empower users in exploring and analyzing plays sourced from Project Gutenberg. The application has achieved significant milestones in enhancing the user experience and providing valuable insights into literary works. By harnessing the power of technology, this application has made a substantial contribution to the field of literature and theater studies.

Throughout the development of the app, several key achievements have been accomplished. Firstly, a vast collection of plays from different eras, genres and playwrights has been curated, allowing users to immerse themselves in the world of theater and access a diverse selection of works. The intuitive and user-friendly interface ensures seamless navigation and exploration, enabling users to easily find and engage with the plays that pique their interest.

Moreover, the application's analytical tools and functionalities have empowered users to delve deeper into the plays and actively engage with the content. The annotation feature, along with the ability to highlight passages and make personal notes, encourages users to develop their interpretations and reflections, fostering a personalized reading experience. The inclusion of critical essays, scholarly articles and expert annotations further enhances users' understanding by providing valuable context, interpretations and historical background.

6.2 Contributions to the Field

This thesis and the web application development have made notable contributions to literature and theater studies. By providing a platform that integrates technology with the exploration and analysis of plays, several significant contributions have been achieved.

Firstly, the web application expands access to theatrical works and promotes inclusivity within the field. By sourcing plays from Project Gutenberg, which offers a vast collection of public domain texts, the application provides a platform for users to engage with a diverse range of

plays from different eras, genres and playwrights. This accessibility ensures that a broader audience can explore and appreciate theatrical works, regardless of their geographical location, socioeconomic background, or institutional affiliation. In doing so, it contributes to democratizing access to theater and literature, allowing for a more inclusive and diverse exploration of the field.

Secondly, the application's analytical tools and scholarly annotations offer valuable resources for students, scholars and researchers. The inclusion of critical essays, scholarly articles and expert annotations provides users with a rich context and deep insights into the plays. This contribution to theater studies enhances the academic rigor of the application and supports research, facilitating in-depth analysis and encouraging critical engagement with the plays. The application becomes a valuable tool for scholars and students alike, enabling them to access comprehensive resources and enrich their understanding and interpretation of the plays.

Additionally, the web application's integration of multimedia elements contributes to the field by bridging the gap between textual analysis and performance appreciation. By incorporating audio recordings of performances, video adaptations and production photos, the application enables users to experience the visual and auditory aspects of the plays. This multimedia approach enhances the understanding of the dramatic elements, acting techniques and staging choices employed in the theatrical productions. It also facilitates the exploration of the plays as living and evolving works of art, expanding the discourse around performance studies and theater history.

Moreover, the application's social features foster a sense of community and encourage discussions among users. By providing a platform for virtual reading groups, participation in discussions and the sharing of thoughts and insights, the application contributes to the formation of a dynamic and collaborative community of theater enthusiasts. This community aspect enriches the exploration and analysis of the plays through the exchange of diverse perspectives and the cultivation of a shared passion for theater. In doing so, the application fosters a vibrant space for discourse within the field, contributing to the overall development of theater studies.

Overall, the web application's contributions to the field of literature and theater studies are significant. It expands access to theatrical works, promotes inclusivity, provides valuable resources for academic research, bridges the gap between textual analysis and performance appreciation and fosters a vibrant community of theater enthusiasts. These contributions have the potential to shape the way plays are explored, analyzed and appreciated in the digital age and they open up new possibilities for further advancements in the field of theater studies.

6.3 Reflection on the Research Process

Reflecting on the research process, the development of this web application has involved extensive research, analysis and collaboration. The selection and curation of the play collection required thorough consideration of numerous factors, including historical significance, popularity and diversity. The integration of analytical tools and multimedia elements necessitated careful evaluation and implementation of appropriate technologies. Collaboration with experts in the field has ensured the inclusion of comprehensive analysis and valuable insights, contributing to the application's academic rigor.

The implications of this research and the resulting web application are far-reaching. The application provides a valuable resource for students, scholars and theater enthusiasts, enabling them to explore and analyze plays with ease and depth. It opens up new avenues for literary and theater studies, facilitating research, critical analysis and discussions within the field. The application's user-friendly interface and accessible content also make it a useful educational tool, fostering engagement and understanding among students of all levels.

6.4 Implications and Future Directions

Future directions for this research and application development are promising. The expansion of the play collection to include more diverse voices, underrepresented playwrights and contemporary works will ensure a more comprehensive representation of theater. Incorporating natural language processing and machine learning techniques can further enhance the application's analytical capabilities, providing automated insights and personalized recommendations. Integration with live streaming platforms or partnerships with theater companies can bring real-time performances and virtual theater experiences to the app.

This thesis has presented a web application that empowers users to explore and analyze plays sourced from Project Gutenberg. Through its extensive features and functionalities, the application enhances the user experience, provides valuable insights and contributes to the field of literature and theater studies. The research process undertaken for this project has yielded significant achievements, with implications and future directions that promise to shape the future of theater exploration and analysis in the digital age.

7 *References*

[1] Bean, R. The Use of Project Gutenberg and Hexagram Statistics to Help Solve Famous Unsolved Ciphers. School of Information Technology and Electrical Engineering, University of Queensland, Australia 4072.

[2] Polak, Yuri Polak, Laboratory for Local Networks, Project Gutenberg celebrates its 50th anniversary.

[3] A Standardized Project Gutenberg Corpus for Statistical Analysis of Natural Language and Quantitative Linguistics. Entropy, 22(1), 126.

[4] Gujjar, J. P., & Kumar, H. R. Sentiment Analysis: Textblob For Decision Making. International Journal of Scientific Research in Engineering and Technology (IJSRET).

[5] Brooke, J., Hammond, A., & Hirst, G. GutenTag: An NLP-driven Tool for Digital Humanities Research in the Project Gutenberg Corpus. In Proceedings of the ACL-IJCNLP 2015 System Demonstrations.