



INTERNATIONAL  
HELLENIC  
UNIVERSITY

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





INTERNATIONAL  
HELLENIC  
UNIVERSITY

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 30<sup>th</sup>, 2023.

*(Signature)*

*(Signature)*

*(Signature)*

.....

, IHU

.....

, IHU

.....

, IHU

Thessaloniki, June 2023

*(Signature)*

.....

**Nikolaos Popis**  
Computer engineer

*(Signature)*

.....

**George Pardis**  
Computer engineer

## Περίληψη

Αυτή η διατριβή παρουσιάζει μια νέα προσέγγιση για την εξαγωγή μεταδεδομένων και την ανάλυση παιχνιδιών από το Project Gutenberg χρησιμοποιώντας μια διαδικτυακή εφαρμογή. Οι χρήστες μπορούν να προβάλλουν βιβλία από το Project Gutenberg, να δημιουργήσουν λίστες αγαπημένων και να τις αποθηκεύσουν για προσωπική χρήση. Η αναπτυγμένη διαδικτυακή εφαρμογή προσφέρει μια ευρεία γκάμα λειτουργιών και αναλύσεων. Με διάφορες τεχνικές, οι χρήστες μπορούν να εξάγουν μεταδεδομένα από τα βιβλία και να πραγματοποιήσουν στατιστική και ανάλυση απόψεων για τα εξαγόμενα δεδομένα.

**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. Users can view books from project Gutenberg, create favorite lists and save them for personal use. The developed web app offers a wide range of functionalities and analytics. With various techniques users can extract metadata from books and perform statistical and sentiment analysis on the extracted data.

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



# Table of contents

<b>1</b>	<b><i>Introduction</i></b> .....	<b>1</b>
1.1	Background .....	1
1.2	Objectives .....	1
1.3	Literature Analytics .....	1
1.4	Thesis structure .....	2
<b>2</b>	<b><i>Project Gutenberg: Purpose and Significance</i></b> .....	<b>3</b>
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
2.4.1	Understanding the Role of Analytics in Literature .....	6
2.4.2	Analytical Approaches in Literature Analytics .....	7
2.4.3	Benefits and Implications of Literature Analytics .....	7
<b>3</b>	<b><i>System Design and Implementation</i></b> .....	<b>9</b>
3.1	Architecture Overview .....	9
3.1.1	Introduction .....	9
3.1.2	System Components .....	9
3.1.3	Client-Server Architecture .....	10
3.1.4	Technologies and Tools .....	10
3.1.5	Communication and Data Flow .....	11
3.1.6	Deployment Considerations .....	12
3.2	Data Collection and Preprocessing .....	12
3.2.1	Introduction .....	12
3.2.2	Data Collection .....	12
3.2.3	Data Preprocessing .....	14
3.3	Application Development .....	15
3.3.1	Introduction .....	15
3.3.2	Architectural Decisions .....	15

3.3.3	User Interface Design .....	17
3.3.4	Feature Implementation.....	19
3.3.5	Testing and Debugging .....	20
<b>3.4</b>	<b>Technologies and Tools Used .....</b>	<b>20</b>
3.4.1	Introduction.....	20
3.4.2	Backend Technologies and Additional Libraries .....	20
<b>3.5</b>	<b>Features and Functionalities of the App .....</b>	<b>24</b>
3.5.1	Login and Logout .....	25
3.5.2	Search Anything.....	27
3.5.3	Book List Management .....	28
3.5.4	Raw Text and Rendered HTML Preview.....	32
3.5.5	CSS Rule Creation.....	34
3.5.6	Table View and Interactive Modifications .....	37
3.5.7	Database Integration .....	40
3.5.8	Sentiment Analysis and Analytics .....	42
3.5.9	Analytics Charts .....	43
3.5.10	Sentiment Timeline .....	46
3.5.11	Chart Comparison.....	47
<b>3.6</b>	<b>Methodology .....</b>	<b>50</b>
3.6.1	Development Approach.....	50
3.6.2	Requirements Gathering .....	51
3.6.3	Design and Prototyping .....	51
3.6.4	Development Process .....	51
3.6.5	Data Collection and Preprocessing .....	51
3.6.6	User Testing.....	51
3.6.7	Ethical Considerations .....	52
<b>3.7</b>	<b>Challenges and Limitations .....</b>	<b>52</b>
<b>4</b>	<b><i>Evaluation</i>.....</b>	<b>54</b>
<b>4.1</b>	<b>Introduction.....</b>	<b>54</b>
<b>4.2</b>	<b>Purpose and Objectives of Evaluation .....</b>	<b>54</b>
<b>4.3</b>	<b>Designing the Evaluation Survey .....</b>	<b>54</b>
<b>4.4</b>	<b>Administering the Evaluation Survey .....</b>	<b>55</b>
<b>4.5</b>	<b>Data Collection and Analysis.....</b>	<b>55</b>

4.6	Interpretation and Application of Evaluation Findings .....	55
4.7	Survey Results.....	56
4.8	Conclusion .....	57
5	<i>Future Work and Improvements</i> .....	58
5.1	Potential enhancements to Metadata extraction .....	58
5.2	Scalability and Performance Optimization.....	59
5.3	Additional Improvements.....	61
6	<i>Conclusion</i> .....	63
6.1	Summary of Achievements .....	63
6.2	Contributions to the Field .....	63
6.3	Reflection on the Research Process.....	65
6.4	Implications and Future Directions.....	65
7	<i>References</i> .....	66

# 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 plays and books plays a crucial role in understanding the nuances of literary works 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 sentiment analysis at different levels (dialog, act, character, and timeline), 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 remainder of this thesis is structured as follows: Chapter 3 delves into the purpose and significance of Project Gutenberg, discussing its history, evolution, and its 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 groundbreaking endeavor aimed at digitally preserving and disseminating 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 digitization 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 digitizing 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.

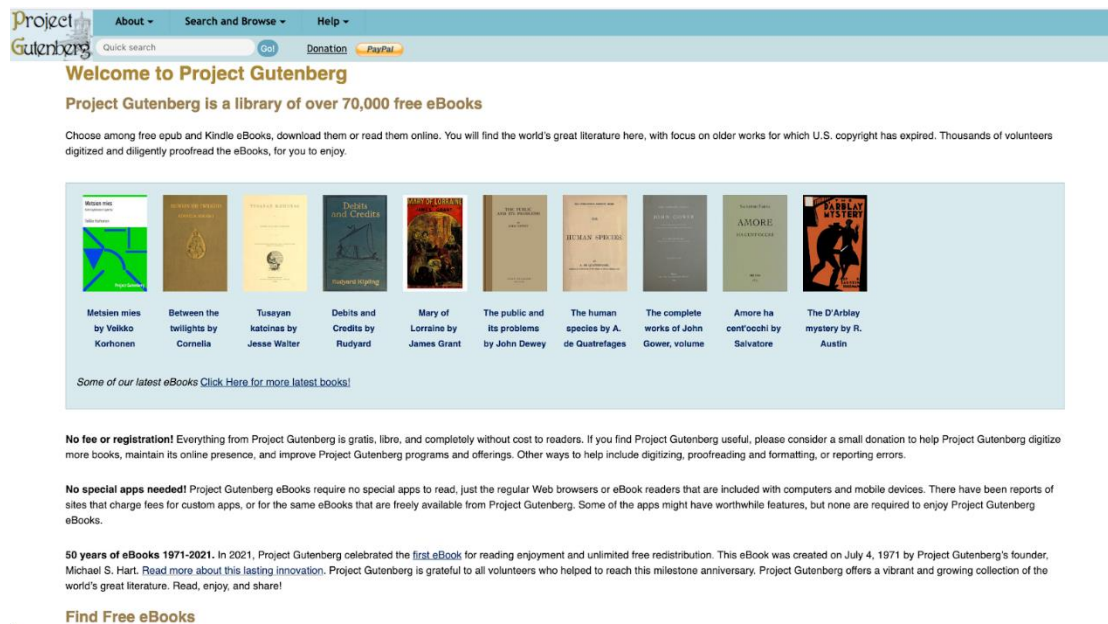


Figure 1 - Project Gutenberg Website

## 2.2 Significance of Project Gutenberg in Literature

### *Preservation*

Project Gutenberg plays a pivotal role in safeguarding 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

PG'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. Our critique can be summarized in two key points. 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 PG. 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.

Secondly, 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. To ensure reproducibility, it becomes imperative to make corpora widely accessible in a standardized format. 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 few.

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 as a result 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.

### ***2.4.1 Understanding the Role of Analytics in Literature***

**Definition and Scope of Literature Analytics:** 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.

**The Rise of Digital Humanities:** 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.

#### ***2.4.2 Analytical Approaches in Literature Analytics***

**Text Mining and Natural Language Processing (NLP):** Text mining techniques enable researchers to extract meaningful information from literary texts, such as identifying recurring themes, analyzing character relationships, and uncovering narrative structures. Natural Language Processing (NLP) techniques complement text mining by providing tools for sentiment analysis, topic modeling and language processing. Through these approaches, researchers gain a deeper understanding of the textual content and its implications.

**Network Analysis and Visualization:** Network analysis allows researchers to examine the relationships and connections between characters, themes and other elements within literary works. By mapping these relationships, scholars can gain insights into the structure and dynamics of a narrative. Network visualization techniques further enhance the analysis by visually representing these connections, revealing patterns, and facilitating a more intuitive understanding of the text.

**Computational Stylistics and Authorship Attribution:** Computational stylistics focuses on analyzing writing style and its variations within literary works. This approach helps researchers identify authorship patterns, detect stylistic changes and attribute authorship in cases of disputed or anonymous works. Leveraging stylometric analysis and machine learning algorithms, computational stylistics has emerged as a valuable tool in authorship studies and literary criticism.

#### ***2.4.3 Benefits and Implications of Literature Analytics***

**Enhanced Literary Analysis:** Literature analytics complements traditional literary analysis methods by providing empirical evidence and quantitative insights. It enriches the interpretation of literary works, revealing underlying patterns, motifs and symbolism that might go unnoticed through subjective analysis alone. By integrating analytics into literary analysis, researchers gain a more comprehensive understanding of the text and its multiple layers of meaning.

**Facilitating Collaborative Research:** Literature analytics fosters collaboration among scholars, researchers and interdisciplinary teams. Digital platforms and tools enable the sharing of data, annotations and analytical approaches, promoting collaborative analysis and knowledge exchange. Such collaborative efforts enhance the robustness and validity of research findings, while also nurturing a sense of community within the field.

**Implications for Teaching and Pedagogy:** The incorporation of literature analytics in educational settings has transformative implications for teaching and pedagogy. By introducing

students to analytics tools and methodologies, educators enhance their critical thinking skills, engage them in data-driven analysis and empower them to explore literary texts in new and dynamic ways. Literature analytics promotes a deeper appreciation of the complexities and nuances of literature, enabling students to become active participants 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.

In summary, this chapter provides 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.

**Method of Data Collection:** The chosen method for collecting book metadata from Project Gutenberg is through web scraping. Web scraping involves programmatically extracting data from web pages by parsing the HTML structure of the website. In this case, the application utilizes libraries like BeautifulSoup or Gutenbergpy to retrieve the desired book metadata.

1. **Retrieving Book URLs:** The first step in the data collection process is to identify the URLs of the books available on Project Gutenberg. This can be achieved by exploring the website's structure or utilizing the search functionality provided by Project Gutenberg. The application may employ techniques such as crawling or targeted search queries to gather a list of book URLs.
2. **Accessing Book Pages:** Once the book URLs are obtained, the application can programmatically access the web pages corresponding to each book. Libraries like BeautifulSoup or Gutenbergpy facilitate parsing and navigating the HTML structure of the web pages.
3. **Extracting Book Metadata:** With access to the book pages, the application can extract the relevant metadata such as the book title, author, publication date and genre. By inspecting the HTML elements containing this information, the application can extract and store the metadata for further processing or storage.

**Limitations and Challenges:** During the data collection process, several limitations and challenges may arise:

1. **Website Structure Changes:** Project Gutenberg may periodically update its website structure, which could potentially impact the web scraping process. Changes in HTML elements, class names, or page layouts might require adjustments to the scraping code to ensure continued extraction of book metadata.
2. **Rate Limiting:** To prevent excessive traffic and ensure fair usage, Project Gutenberg may enforce rate limiting or have restrictions on the number of requests per unit of time. It is important to adhere to any guidelines or limitations to maintain a positive relationship with the website and prevent potential disruptions to the data collection process.
3. **Data Quality and Completeness:** While Project Gutenberg aims to provide accurate and comprehensive book metadata, occasional errors or missing information may occur. The application should handle such cases by implementing error handling and data validation mechanisms to ensure the collected metadata is reliable and usable.

4. **Legal and Ethical Considerations:** When scraping data from external sources, it is essential to review and comply with the terms of service, copyright laws and any applicable usage policies. Respecting the rights and intentions of content providers, such as Project Gutenberg, is crucial to maintaining ethical data collection practices.

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 is crucial to ensure the quality and usability of the collected book metadata. This subsection explains the preprocessing steps applied to the data, which may include the following:

**Data Cleaning:** Data cleaning involves removing any irrelevant or duplicate entries, handling missing values and addressing inconsistencies in the data. This step ensures that the collected book metadata is accurate and complete. Common techniques for data cleaning, such as removing duplicate records, filling in missing values based on statistical measures and correcting inconsistent data formats, are described in detail.

**Data Transformation:** Data transformation involves converting the collected book metadata into a suitable format for further processing and analysis. This may include transforming the data into a structured format, such as JSON or CSV, or integrating it into a database management system like PostgreSQL. The transformation process may also involve applying data normalization techniques to ensure uniformity and comparability across different attributes.

**Data Integration:** Data integration involves combining the collected book metadata with any existing data sources or external datasets, if applicable. This step enables enriched analysis and enhances the functionality of the application. The integration process may involve matching and merging records based on common attributes or utilizing data linking techniques to establish relationships between different datasets.

**Data Validation and Quality Assurance:** Data validation and quality assurance measures are applied to ensure the accuracy, consistency and reliability of the collected book metadata. This may involve performing data validation checks, such as verifying the correctness of book titles and authors against authoritative sources or conducting statistical analysis to identify potential

outliers or anomalies in the data. Quality assurance techniques, such as data profiling and data auditing, may also be employed to assess the overall data quality and integrity.

### ***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.

**Software Architecture Patterns:** The chosen software architecture pattern for the application is the Model-View-Controller (MVC) pattern. MVC is a widely adopted architectural pattern that separates the concerns of data management, user interface presentation and user interactions. In the context of the application, the MVC pattern facilitates the organization of code into three distinct components:

1. **Model:** The model represents the data and the business logic of the application. It encapsulates the data structures, algorithms and rules for manipulating and managing the application's data. In the backend, the model layer includes components responsible for data validation, database interaction and business logic implementation.
2. **View:** 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 backend, the view layer focuses on generating the API responses and serializing data for client consumption.
3. **Controller:** The controller acts as an intermediary between the model and the view. It receives user input, processes it and updates the model and view accordingly. In the backend, the controller handles incoming requests, processes the data, invokes the appropriate model operations and determines the appropriate view to render as a response.

The MVC pattern provides separation of concerns, allowing for better maintainability, reusability and testability of the codebase. It enables developers to make modifications or extend functionalities in one component without impacting the others, promoting code organization and reducing code duplication.

**Design Principles:** Several design principles, such as SOLID (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation and Dependency Inversion) and DRY (Don't Repeat Yourself), were applied during the development of the application. These principles guide the design and implementation of software components, promoting modularity, flexibility and code readability.

1. **Single Responsibility Principle (SRP):** Each class or module should have a single responsibility or reason to change. By adhering to SRP, the codebase is organized into smaller, focused components that are easier to understand, maintain and test.
2. **Open-Closed Principle (OCP):** Software entities should be open for extension but closed for modification. The application design is structured in a way that new functionality can be added by extending existing components rather than modifying them. This reduces the risk of introducing bugs or breaking existing code when making changes.
3. **Liskov Substitution Principle (LSP):** Subtypes should be substitutable for their base types without altering the correctness of the program. Adhering to LSP ensures that derived classes or modules can be used interchangeably with their base classes or modules, enhancing code reusability and promoting polymorphism.
4. **Interface Segregation Principle (ISP):** Clients should not be forced to depend on interfaces they do not use. ISP advocates for the creation of small, specific interfaces instead of large, general-purpose interfaces. This reduces the impact of changes and facilitates better decoupling between components.
5. **Dependency Inversion Principle (DIP):** High-level modules should not depend on low-level modules. Both should depend on abstractions. DIP promotes the use of abstractions and interfaces to decouple components and facilitate easier integration, testing and maintenance.

Applying these design principles helps to create a codebase that is more flexible, extensible and easier to understand. It reduces code duplication, improves code maintainability and facilitates the implementation of changes or new features.

**Scalability and Maintainability:** 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.

**Design Process:** 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 and stakeholders 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.

**Usability and Accessibility:** 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.

**Design Framework or Library:** The chosen design framework or library, such as Angular Material or Bootstrap, 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.

The UI design process involves wireframing, prototyping and iterative refinement, guided by usability and accessibility principles. The chosen design framework or library, such as Angular Material or Bootstrap, helps in creating a visually appealing and consistent UI. Customization and theming options ensure that the UI aligns with the application's goals and branding. By considering usability, accessibility and the iterative design process, the UI design aims to provide an intuitive and inclusive user experience.

### ***3.3.4 Feature Implementation***

This subsection focuses on the implementation of key features in the application. It provides a detailed explanation of each feature, including the underlying functionality and the technical implementation approach. The features to be implemented may include:

**Book Listing:** The book listing feature allows users to view the available books from Project Gutenberg. It includes functionalities such as pagination, sorting and filtering to enhance the user's browsing experience. The implementation details, including the retrieval of book metadata from the database or API and the display of relevant information, are described.

**Favorite Lists:** The favorite lists feature enables users to create and manage custom lists of their favorite books. Users can create, edit and delete lists, as well as add or remove books from the lists. The implementation covers the creation of database schemas or data models to store the favorite list information and the development of user interfaces for managing the lists.

**Book Details:** The book details feature allows users to view the detailed information of a selected book, including the book title, author, publication date and genre. Users can also view the book content in its raw HTML format as it appears in Project Gutenberg. The implementation involves retrieving and displaying the book details and content, as well as handling user interactions, such as zooming or scrolling through the book content.

**HTML Parsing and Table Extraction:** This feature focuses on parsing the raw HTML content of the selected book to extract specific information, such as acts, persons and dialogue. The implementation may involve utilizing HTML parsing libraries or implementing custom parsing algorithms to extract the desired data. The extracted information can then be used to generate formatted tables or provide additional insights to the user.

**Table Viewing:** Here after a user has saved a book in the database, the table data comes from the database and the html tabs are disabled.

**Statistical Analysis:** Once a book is fetched from the database the user sees a couple of graphs regarding the book. These graphs include word count, sentence count, words per character, sentences per character, words per act, sentences per act.

**Sentiment timeline:** Along with the statistical analysis the user can also view in another tab the sentiment for the timeline of the book. Each sentence is analyzed and has a sentiment score from -1 to 1 depending on the emotion of the sentence (negative, neutral, positive). This is viewed on the sentiment timeline tab.

**Comparison:** For the statistical analysis and sentiment timeline, the user can select up to 2 processed books to compare in all the above graphs.

### ***3.3.5 Testing and Debugging***

This subsection covers the testing and debugging strategies employed during the development process. It discusses the different types of testing performed, such as unit testing, integration testing and user acceptance testing. The tools and frameworks utilized for testing, such as Jasmine or PyTest, are explained. Additionally, the debugging techniques employed to identify and resolve issues or errors are described, ensuring a stable and robust application.

In summary, the Application Development section provides a comprehensive overview of the implementation process for the application. It covers the architectural decisions, user interface design and the implementation of key features. The chosen software architecture patterns and design principles are explained, ensuring a maintainable and scalable codebase. The features, including book listing, favorite lists, book details and HTML parsing, are implemented and described in detail. The testing and debugging strategies employed ensure the functionality and stability of the application. This section demonstrates the practical implementation of the system design, resulting in a fully functional and user-friendly application that meets the specified requirements.

## ***3.4 Technologies and Tools Used***

### ***3.4.1 Introduction***

The Technologies and Tools Used 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.

Alembic (version 1.8.1): Alembic is a database migration tool used for managing database schema changes. It enables easy and automated handling of database migrations.

AnyIO (version 3.6.2): AnyIO is a library that provides a unified interface for asynchronous programming in Python. It allows developers to write asynchronous code that can run on different event loops seamlessly.

Asyncpg (version 0.27.0): Asyncpg is a high-performance asynchronous PostgreSQL database driver for Python. It provides efficient and reliable communication with the PostgreSQL database.

Certifi (version 2022.9.24): Certifi is a Python package that provides a curated collection of Root Certificates for verifying SSL/TLS connections.

Chardet (version 5.1.0): Chardet is a Python library used for character encoding detection. It helps in automatically detecting the character encoding of textual data.

Charset-normalizer (version 2.1.1): Charset-normalizer is a library that normalizes character encodings, ensuring consistent and standardized handling of text data.

Click (version 8.1.3): Click is a Python package used for creating command-line interfaces. It simplifies the process of building command-line applications with intuitive and user-friendly interfaces.

Databases (version 0.6.2): Databases is a library that provides a simple and consistent API for interacting with databases. It abstracts away the underlying database driver and offers a unified interface for executing database queries.

Dnspython (version 2.2.1): Dnspython is a DNS toolkit for Python. It provides functionalities for querying DNS records and performing DNS-related operations.

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.

Future (version 0.18.2): Future is a Python package that provides support for running Python 2.x code on Python 3.x. It facilitates compatibility and smooth transition between different Python versions.

Greenlet (version 2.0.1): Greenlet is a library that provides lightweight cooperative multitasking in Python. It enables the efficient execution of concurrent tasks without the need for threading or multiprocessing.

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.

H11 (version 0.14.0): H11 is a pure-Python, minimalistic HTTP/1.1 parser library. It provides a low-level interface for parsing and handling HTTP requests and responses.

Httptools (version 0.5.0): Httptools is a Python package that provides fast and efficient HTTP parsing and handling capabilities. It is commonly used in high-performance web applications.

Idna (version 3.4): Idna is a Python package that provides support for Internationalized Domain Names (IDN). It facilitates the handling and conversion of domain names containing non-ASCII characters.

Isodate (version 0.6.1): Isodate is a library for parsing and formatting dates and times in Python. It provides a comprehensive set of functions for working with ISO 8601 date and time representations.

Lxml (version 4.9.1): Lxml is a library for processing XML and HTML documents in Python. It provides a fast and efficient API for parsing, manipulating and generating XML and HTML content.

Mako (version 1.2.4): Mako is a template engine for Python. It allows for the generation of dynamic content by combining templates with data.

MarkupSafe (version 2.1.1): MarkupSafe is a library that provides utilities for escaping and preserving the semantics of HTML, XML and other markup languages.

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.

Psycopg2-binary (version 2.9.1): Psycopg2-binary is a PostgreSQL adapter for Python. It allows Python applications to communicate with the PostgreSQL database using the psycopg2 module.

Pydantic (version 1.10.2): Pydantic is a library for data validation and parsing. It simplifies the creation of data models and provides automatic data validation and serialization.

PyJWT (version 1.7.1): PyJWT is a Python library for encoding and decoding JSON Web Tokens (JWT). It provides functions for generating and verifying JWTs, which are commonly used for authentication and authorization.

PyMongo (version 4.3.3): PyMongo is a Python driver for MongoDB, a popular NoSQL database. It enables Python applications to interact with MongoDB and perform database operations.

**Pyparsing (version 2.4.7):** Pyparsing is a library for parsing and processing textual data. It provides a flexible and expressive syntax for defining grammars and parsing rules.

**Python-dateutil (version 2.8.2):** Python-dateutil is a library that provides extensions to the standard Python datetime module. It simplifies date and time manipulation and supports parsing and formatting of various date and time formats.

**Python-dotenv (version 0.21.0):** Python-dotenv is a library that allows the loading of environment variables from a .env file. It simplifies the management of sensitive information and configuration settings.

**Pytz (version 2022.6):** Pytz is a library for working with time zones in Python. It provides functions for converting and manipulating time zone information.

**PyYAML (version 6.0):** PyYAML is a library for parsing and serializing YAML (Yet Another Markup Language) data. It allows for the easy integration of YAML data in Python applications.

**Rdflib (version 4.2.2):** Rdflib is a Python library for working with Resource Description Framework (RDF) data. It provides tools for creating, parsing and querying RDF graphs.

**Rdflib-sqlalchemy (version 0.5.3):** Rdflib-sqlalchemy is an extension to Rdflib that allows RDF graphs to be stored and queried using the SQLAlchemy ORM (Object-Relational Mapping) library.

**Requests (version 2.28.1):** Requests is a popular Python library for making HTTP requests. It provides a simple and intuitive API for interacting with web services.

**Six (version 1.16.0):** Six is a Python package that provides compatibility utilities for bridging the gap between Python 2 and Python 3. It enables code compatibility and smooth migration between different Python versions.

**Sniffio (version 1.3.0):** Sniffio is a library for detecting the current event loop in an asynchronous environment. It helps in ensuring compatibility and interoperability between different event loop implementations.

**SPARQLWrapper (version 1.8.5):** SPARQLWrapper is a Python library for working with SPARQL endpoints. It provides a simple and convenient API for querying and retrieving RDF data using SPARQL queries.

**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.

**Starlette (version 0.22.0):** Starlette is a lightweight asynchronous web framework for Python. It provides a minimalistic and efficient foundation for building high-performance web applications.

Typing-extensions (version 4.4.0): Typing-extensions is a package that provides additional type hinting capabilities for Python. It extends the standard typing module with additional features.

Urllib3 (version 1.26.13): Urllib3 is a powerful HTTP client library for Python. It provides a high-level interface for making HTTP requests and supports various features such as connection pooling and SSL/TLS.

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.

Uvloop (version 0.17.0): Uvloop is a fast, drop-in replacement for the asyncio event loop. It provides substantial performance improvements over the default asyncio event loop implementation.

Watchfiles (version 0.18.1): Watchfiles is a library for monitoring file changes in Python. It allows for the detection of modifications to files and directories in real-time.

Websockets (version 10.4): Websockets is a library for building WebSocket servers and clients in Python. It provides a high-level API for establishing and managing WebSocket connections.

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.

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, our 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 our 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, our application offers a diverse selection that caters to various interests and preferences.

To facilitate seamless navigation and exploration, our 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. This comprehensive database ensures that users can easily find the plays they are interested in and gain a deeper understanding of their content and context.

Once users have selected a play to delve into, our application provides them with a range of analytical tools and functionalities. Users can annotate the text, highlight specific passages and make personal notes to capture their thoughts and interpretations. These features not only encourage active engagement with the plays but also enable users to create a personalized reading experience, allowing them to reflect on and revisit their insights.

The application offers comprehensive analysis and insights into the plays. Users can access critical essays, scholarly articles, and annotations from experts in the field, providing them with valuable context, interpretations and historical background. This feature enables users to gain a deeper appreciation of the plays, understand their significance within the literary canon and explore various critical perspectives.

In addition to textual analysis, our application also incorporates multimedia elements to enhance the user experience. Users can access audio recordings of performances, watch video adaptations and view production photos to get a holistic understanding of the plays. These multimedia components bring the plays to life, allowing users to witness the nuances of performance and experience the visual and auditory aspects of the theatrical art form.

To foster a sense of community and facilitate discussion, our application incorporates social features. Users can join virtual reading groups, participate in discussions and share their thoughts and insights with fellow enthusiasts. This interactive element creates a vibrant and dynamic community of theater lovers, fostering a collaborative and enriching environment.

Overall, our web application offers a comprehensive suite of features and functionalities that empower users to explore, analyze and appreciate plays sourced from Project Gutenberg. By combining a vast collection of plays, intuitive navigation, analytical tools, comprehensive analysis, multimedia elements and social features, our application provides a rich and immersive experience that enhances the user's understanding, engagement and enjoyment of literary works.

### ***3.5.1 Login and Logout***

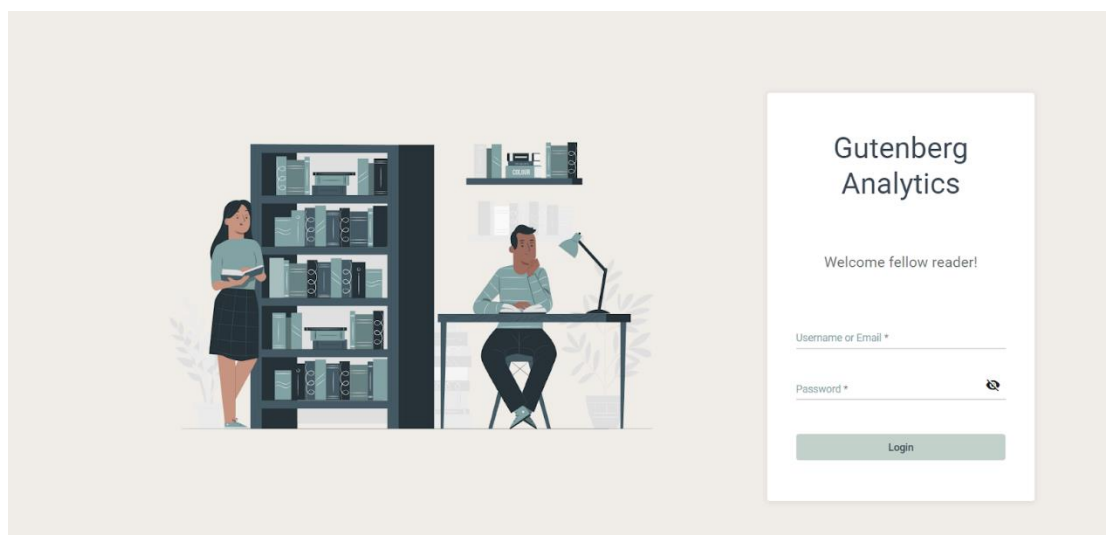
The application includes secure and user-centric login and logout functionality to ensure that only authorized individuals can access its features and content. To gain access to the

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.

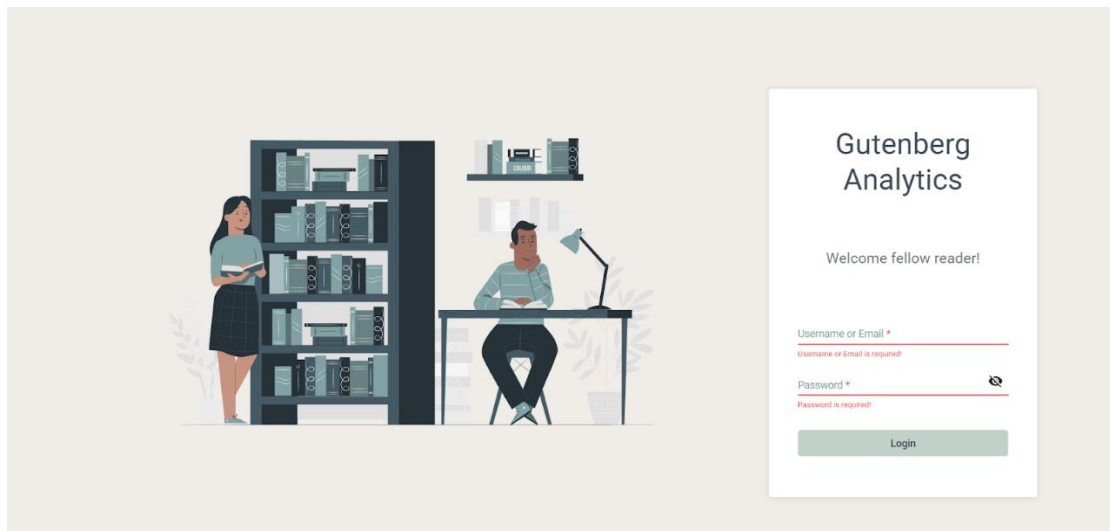
For users who already have an account, the login feature allows them to enter their credentials and gain personalized access to the application's features and functionality. By logging in, users can unlock a range of personalized settings, preferences and content tailored to their specific needs and 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.

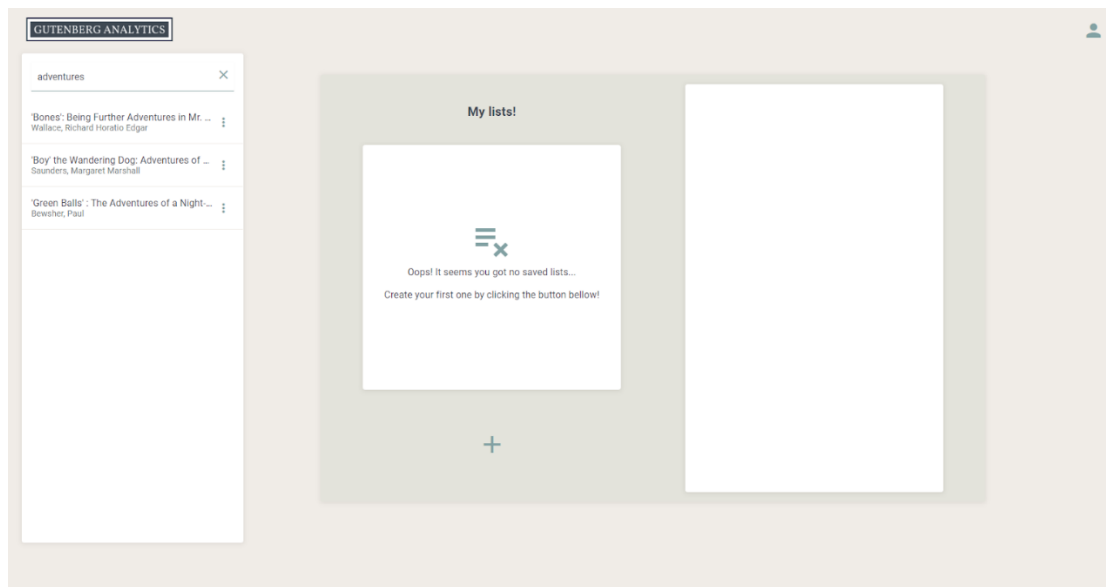
The Gutenberg ID search option allows users to directly enter the unique identifier associated with a particular play or literary work. This streamlined approach allows users to quickly retrieve the exact content they are looking for, bypassing any potential ambiguity. By using the Gutenberg ID as a search criterion, users can enjoy a direct and precise search experience, ensuring quick access to their preferred play or author.

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. This user-friendly approach simplifies the exploration process, allowing users to easily find plays by title. 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 Landing Page*

### **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.

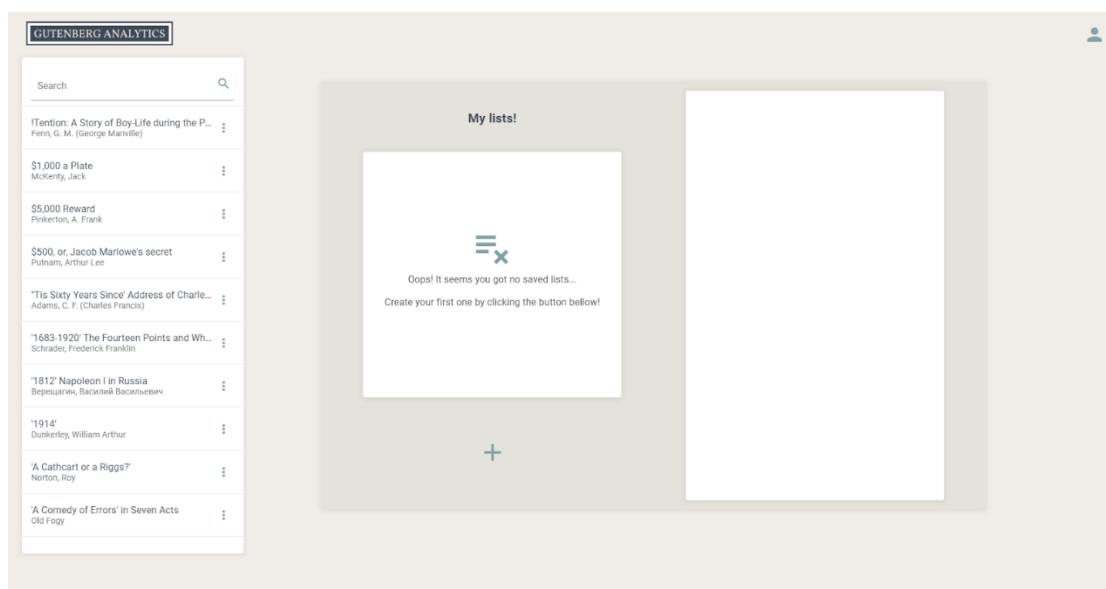


Figure 5 - Application Book List Menu

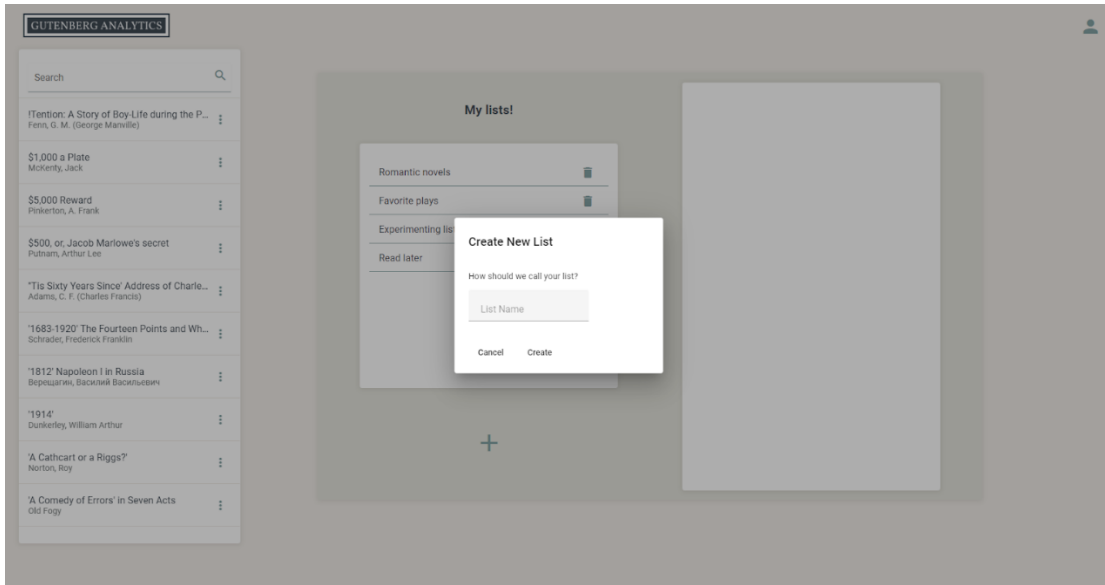


Figure 6 - Book List Creation

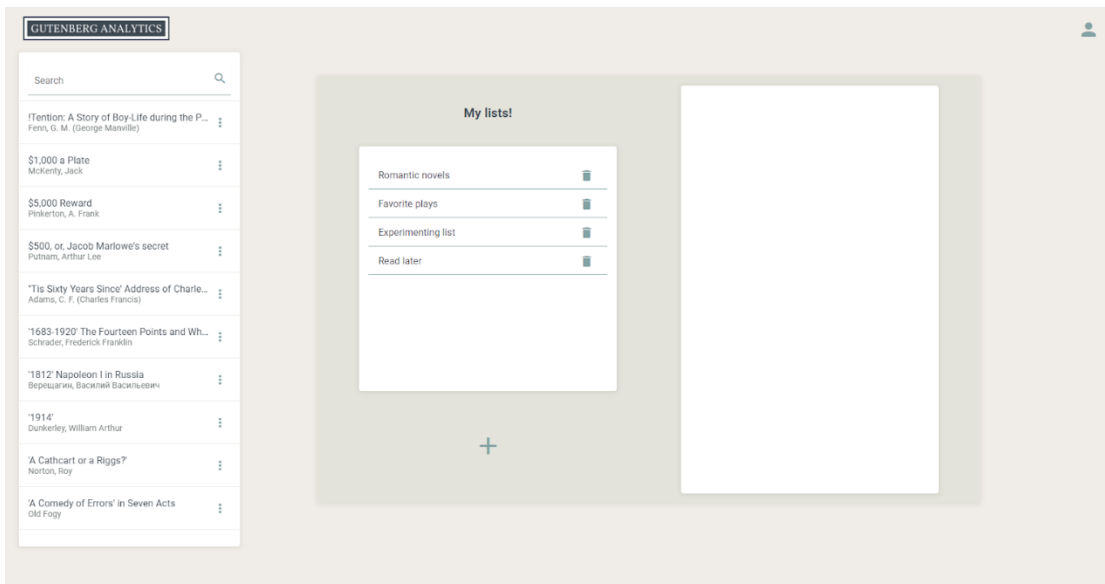


Figure 7 - Book List View

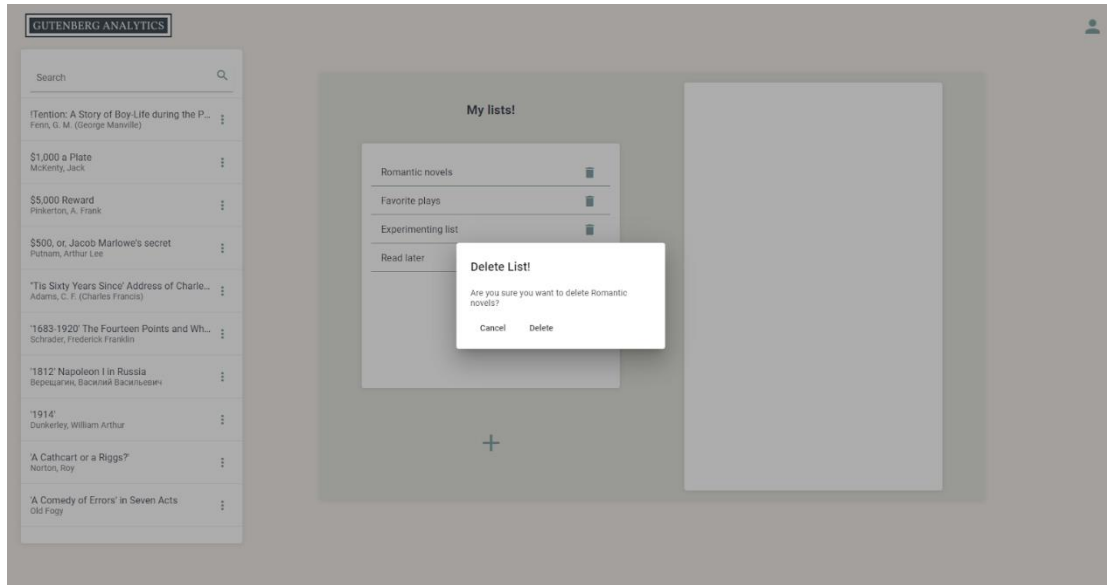


Figure 8 - Book List Delete Modal

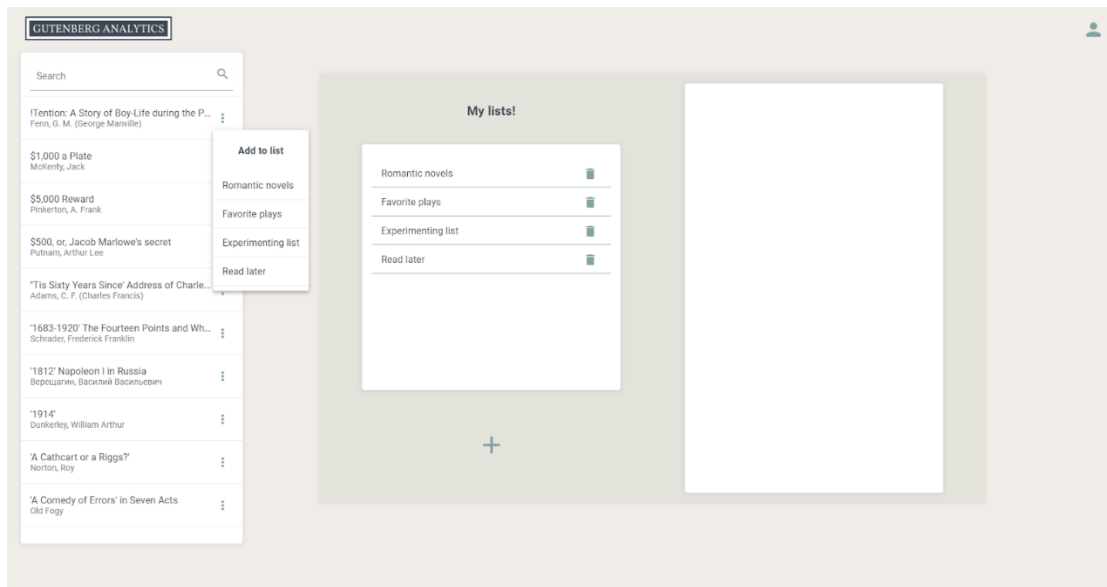


Figure 9 - Add Book to List Menu

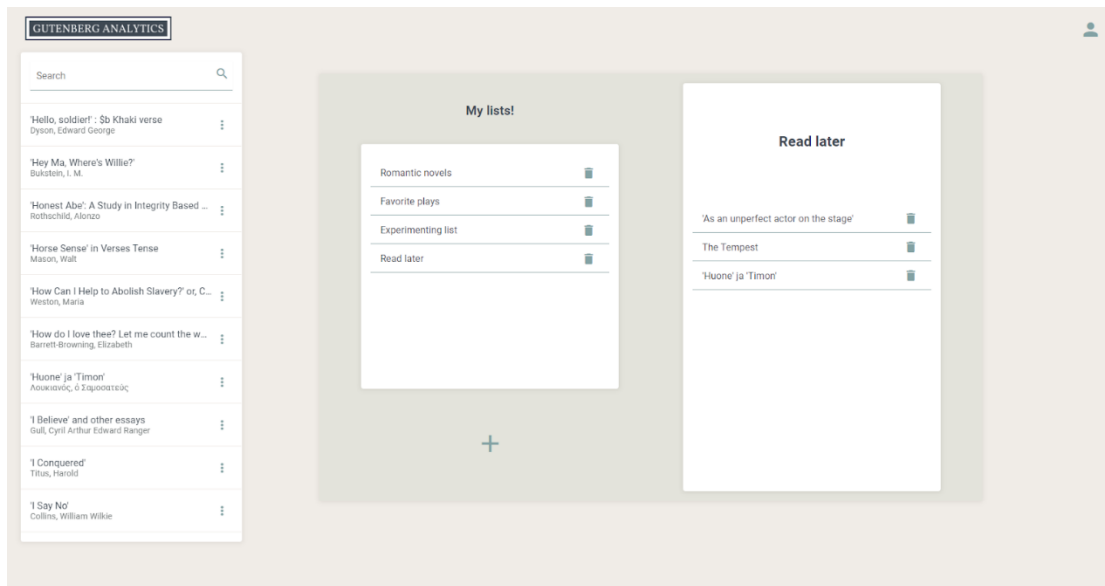


Figure 10 - Book List Item View

### 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 up 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.

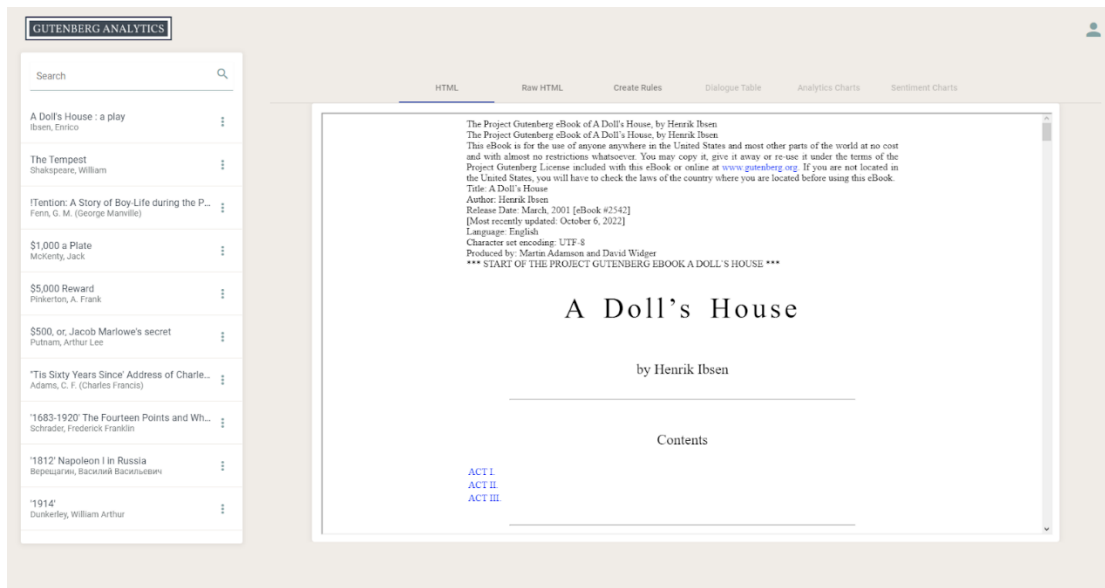


Figure 11 - Book Html View

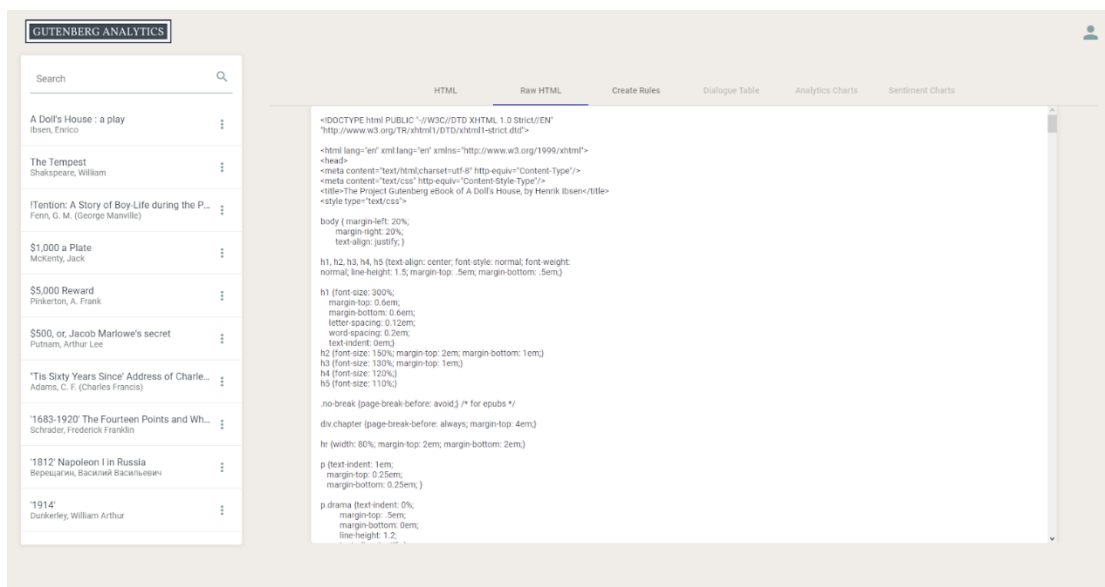


Figure 12 - Book Raw Html View

### 3.5.5 *CSS Rule Creation*

This chapter delves into the powerful CSS (Cascading Style Sheets) rule creation functionality within our application. Recognizing the importance of visual presentation and customization in enhancing the reading experience, our 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.

#### Dynamic HTML Tag Selection

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 HTML document for customization.

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.

#### Assigning Desired Colors

A key component of CSS rule creation is the ability to assign desired colors to the selected HTML tags. Our 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.

#### Custom CSS Rule Application

In addition to selecting predefined HTML tags and assigning colors, our 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.

#### Benefits and Customization

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 customise the visual aspects of the game. Users can adjust font styles, colour contrasts and other visual parameters to optimise readability, ensuring that text is comfortably readable for people with different preferences and visual needs.

The CSS rule creation functionality within our 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.

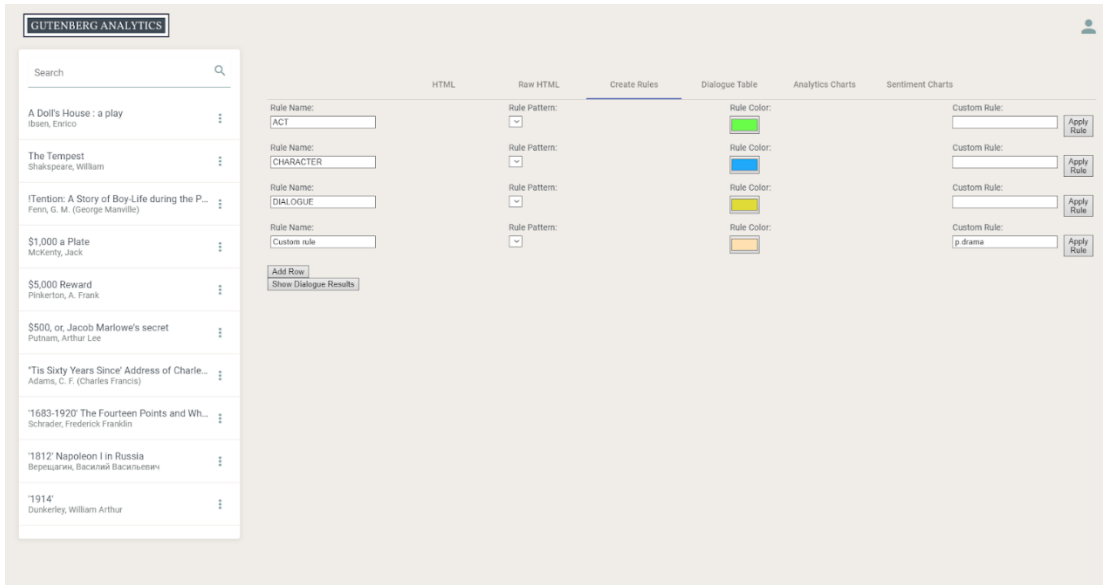


Figure 13 - Create Rules Default Options

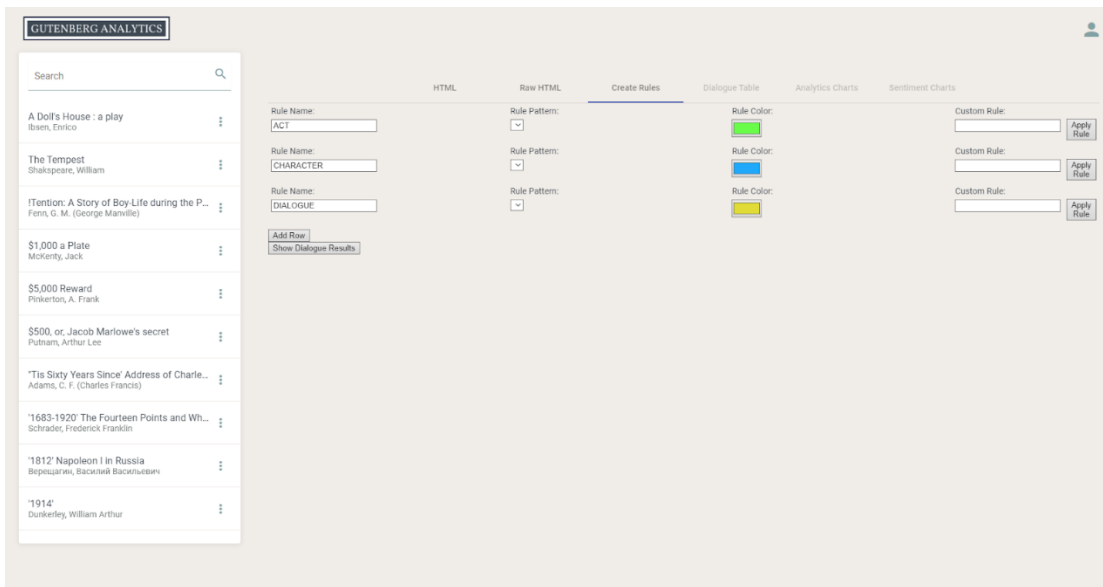


Figure 14 - Create Rules Custom Rule

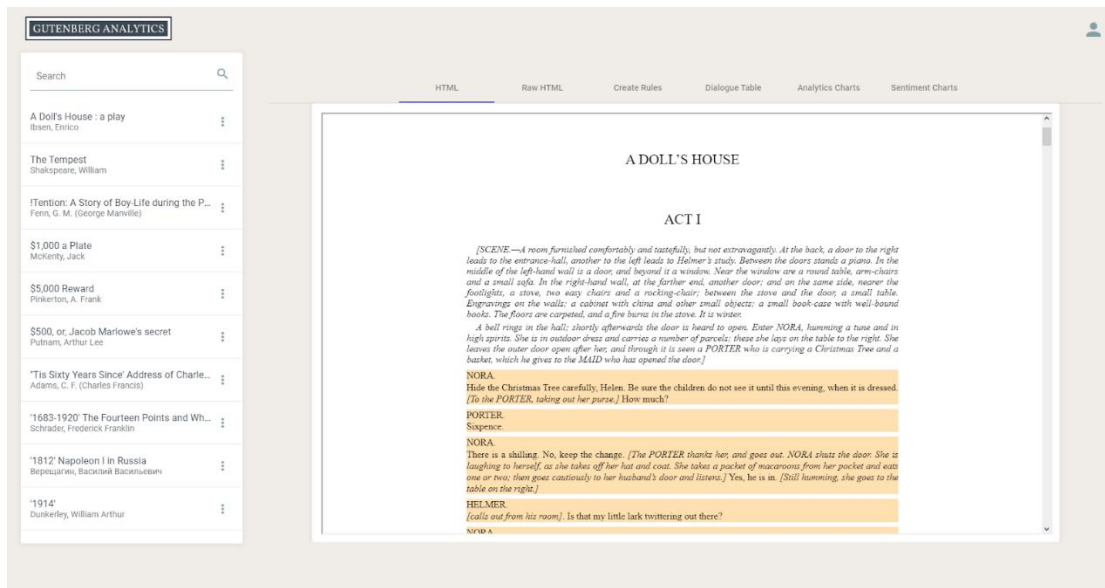


Figure 15 - Rules In Html View

### 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.

**GUTENBERG ANALYTICS**

Search

A Dolls House : a play  
Ibsen, Enrico

The Tempest  
Shakespeare, William

Itention: A Story of Boy-Life during the P...  
Fern, G. M. (George Manville)

\$1,000 a Plate  
McKenry, Jack

\$5,000 Reward  
Pinkerton, A. Frank

\$500, or, Jacob Marlowe's secret  
Pulham, Arthur Lee

'Tis Sixty Years Since' Address of Charle...  
Adams, C. F. (Charles Francis)

'1683-1920 The Fourteen Points and Wh...  
Schrader, Frederick Franklin

'1812' Napoleon I in Russia  
Бережанин, Василий Васильевич

'1914'  
Dunkerley, William Arthur

HTML Raw HTML Create Rules Dialogue Table Analytics Charts Sentiment Charts

Filter by Act Filter by Character Search Dialogue

ACT	CHARACTER	Dialogue
I	NORA	Hide the Christmas Tree carefully, Helen. Be sure the children do not see it until this evening, when it is dressed [To the PORTER, taking out her purse.]How much?
I	PORTER	Sixpence.
I	NORA	There is a shilling. No, keep the change [The PORTER thanks her, and goes out. NORA shuts the door. She is laughing to herself, as she takes off her hat and coat. She takes a packet of macaroons from her pocket and eats one or two, then goes cautiously to her husband's door and listens.]Yes, he is in. [Still humming, she goes to the table on the right.]
I	HELMER	[calls out from his room.] Is that my little lark twittering out there?
I	NORA	[Busy opening some of the parcels.] Yes, it is!
I	HELMER	Is it my little squirrel bustling about?
I	NORA	Yes!
I	HELMER	When did my squirrel come home?
I	NORA	Just now.[Puts the bag of macaroons into her pocket and wipes her mouth.]Come in here, Torvald, and see what I have bought.
I	HELMER	Don't disturb me. [A little later, he opens the door and looks into the room, pen in hand.]Bought, did you say? All these things? Has my little spendthrift been wasting money again?
I	NORA	Yes but, Torvald, this year we really can let ourselves go a little. This is the first Christmas that we have not needed to economise.
I	HELMER	Still, you know, we can't spend money recklessly.
I	NORA	Yes, Torvald, we may be a wee bit more reckless now, mayn't we? Just a tiny wee bit! You are going to have a big salary and earn lots and lots of money.
I	HELMER	Yes, after the New Year, but then it will be a whole quarter before the salary is due.
I	NORA	Pooch! we can borrow until then.
I	HELMER	Nora!Goes up to her and takes her playfully by the ear [The same little featherhead! Suppose, now, that I borrowed fifty pounds today, and you spent it all in the Christmas week, and then on New Year's Eve a slate fell on my head and killed me, and--
I	NORA	[putting her hands over his mouth]. Oh! don't say such horrid things.

Save to Database

Figure 16 - Dialogues Table

**GUTENBERG ANALYTICS**

Search

A Dolls House : a play  
Ibsen, Enrico

The Tempest  
Shakespeare, William

Itention: A Story of Boy-Life during the P...  
Fern, G. M. (George Manville)

\$1,000 a Plate  
McKenry, Jack

\$5,000 Reward  
Pinkerton, A. Frank

\$500, or, Jacob Marlowe's secret  
Pulham, Arthur Lee

'Tis Sixty Years Since' Address of Charle...  
Adams, C. F. (Charles Francis)

'1683-1920 The Fourteen Points and Wh...  
Schrader, Frederick Franklin

'1812' Napoleon I in Russia  
Бережанин, Василий Васильевич

'1914'  
Dunkerley, William Arthur

No Filter

HTML Raw HTML Create Rules Dialogue Table Analytics Charts Sentiment Charts

Filter by Act Filter by Character Search Dialogue

ACT	CHARACTER	Dialogue
II	NORA	[opens her cloak.] Someone is coming now[Opens to the door and listens. No—it is no one. Of course, no one will come today, Christmas Day—not tomorrow either. But, perhaps—opens the door and looks out]. No, nothing in the letterbox, it is quite empty.[Comes forward.]What rubbish! of course he can't be in earnest about it. Such a thing couldn't happen, it is impossible—I have three little children.
II	NURSE	At last I have found the box with the fancy dress.
II	NORA	Thanks; put it on the table.
II	NURSE	[doing so]. But it is very much in want of mending.
II	NORA	I should like to tear it into a hundred thousand pieces.
II	NURSE	What an idea! It can easily be put in order—just a little patience.
II	NORA	Yes, I will go and get Mrs. Linde to come and help me with it.
II	NURSE	What, out again? In this horrible weather? You will catch cold, ma'am, and make yourself ill.
II	NORA	Well, worse than that might happen. How are the children?
II	NURSE	The poor little souls are playing with their Christmas presents, but—
II	NORA	Do they ask much for me?
II	NURSE	You see, they are so accustomed to have their mamma with them.
II	NORA	Yes, but, nurse, I shall not be able to be so much with them now as I was before.
II	NURSE	Oh well, young children easily get accustomed to anything.
II	NORA	Do you think so? Do you think they would forget their mother if she went away altogether?
II	NURSE	Good heavens!—went away altogether?
II	NORA	Nurse, I want you to tell me something I have often wondered about—how could you have the heart to put your own child out among strangers?

Save to Database

Figure 17 - Act Filtering

The screenshot shows the GUTENBERG ANALYTICS interface. On the left is a search sidebar with a list of plays. The main area is titled 'Dialogue Table' and shows a table of dialogue. A dropdown menu is open over the 'Filter by Act' column, listing characters: NORA, PORTER, HELMER, MAID, and MRS LINDE. The dialogue table below shows a conversation between NORA and NURSE.

ACT	CHARACTER	Dialogue
II	NORA	I should like to tear it into a hundred thousand pieces.
II	NURSE	What an idea! It can easily be put in order—just a little patience.
II	NORA	Yes, I will go and get Mrs. Linde to come and help me with it.
II	NURSE	What, out again? In this horrible weather? You will catch cold, ma'am, and make yourself ill.
II	NORA	Well, worse than that might happen. How are the children?
II	NURSE	The poor little souls are playing with their Christmas presents, but—
II	NORA	Do they ask much for me?
II	NURSE	You see, they are so accustomed to have their mamma with them.
II	NORA	Yes, but, nurse, I shall not be able to be so much with them now as I was before.
II	NURSE	Oh well, young children easily get accustomed to anything.
II	NORA	Do you think so? Do you think they would forget their mother if she went away altogether?
II	NURSE	Good heaven!—went away altogether?
II	NORA	Nurse, I want you to tell me something I have often wondered about—how could you have the heart to put your own child out among strangers?

Figure 18 - Character Filtering

The screenshot shows the GUTENBERG ANALYTICS interface with text filtering applied. The 'Filter by Act' dropdown is set to 'NORA' and the 'Search Dialogue' dropdown is set to 'this is'. The dialogue table below shows only the lines spoken by NORA.

ACT	CHARACTER	Dialogue
II	NORA	Oh, yes, that one; but this is another. I ordered it. <u>Torvald</u> mustn't know about it—
II	NORA	This is the way.
II	NORA	I don't know; I think there is; but you must not read anything of that kind now. Nothing horrid must come between us until this is all over.

Figure 19 - Text Filtering

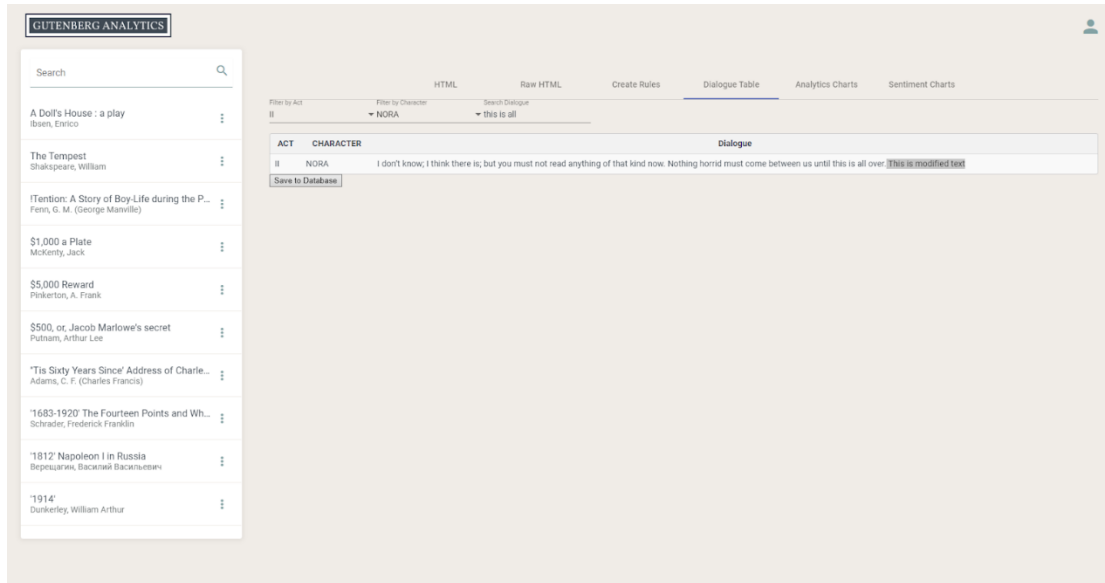


Figure 20 - Text Update

### 3.5.7 Database Integration

In this chapter, we delve into the crucial aspect of database integration within our application. Recognizing the significance of storing and retrieving processed data for future reference, our 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 app.

#### Storing Analyzed Text in the Database

One of the key functionalities of our 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.

#### Retrieving Analyzed Text from the Database

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.

#### Benefits and Importance

The integration of a database within our 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 our 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***

In this chapter, we delve into the realm of sentiment analysis and analytics within our application. Our goal 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 our application, including the breakdown of characters, dialogs and words per dialog, as well as sentiment analysis at various levels.

Our 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. Our 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, our 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, our 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 our 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 our 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, our 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.

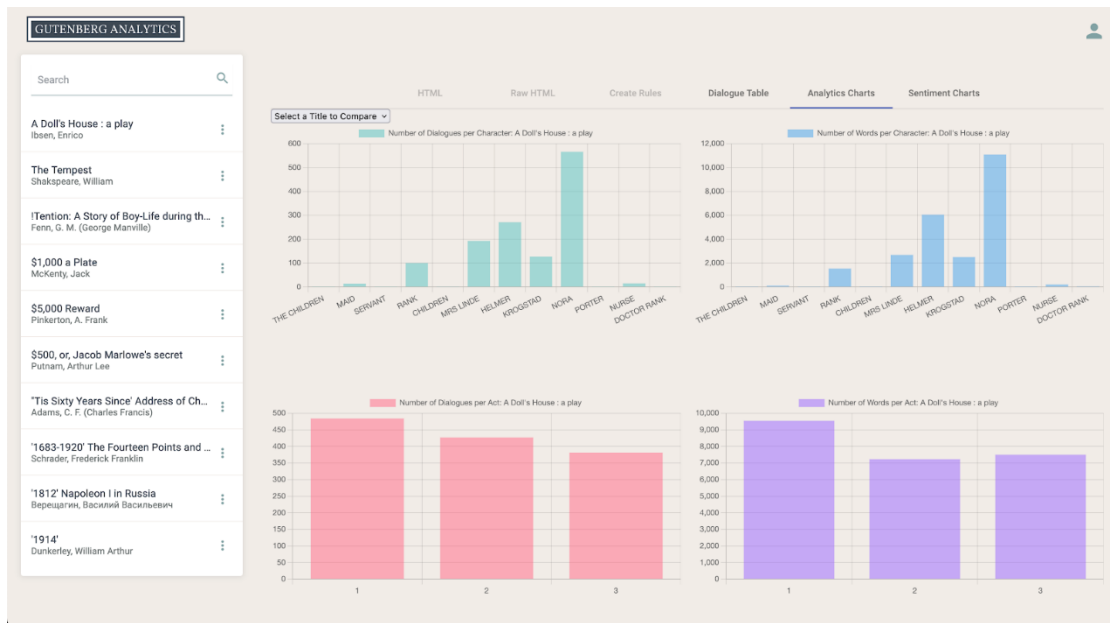


Figure 21 - Analytics Charts (top)



Figure 22 - 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. Through the use of 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.

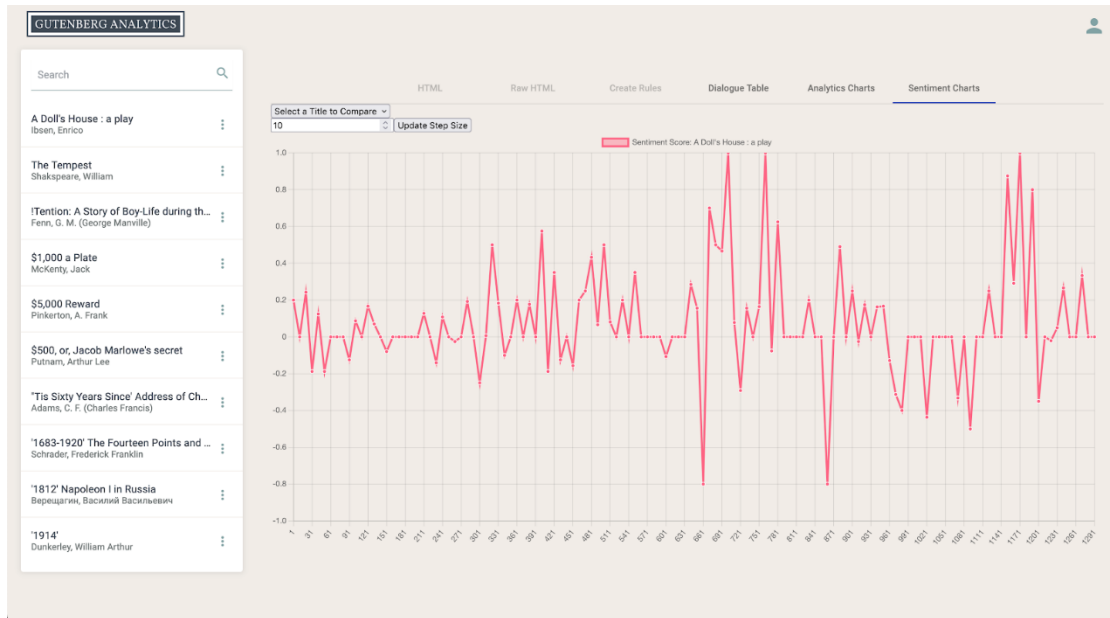


Figure 23 - Sentiment Timeline Chart

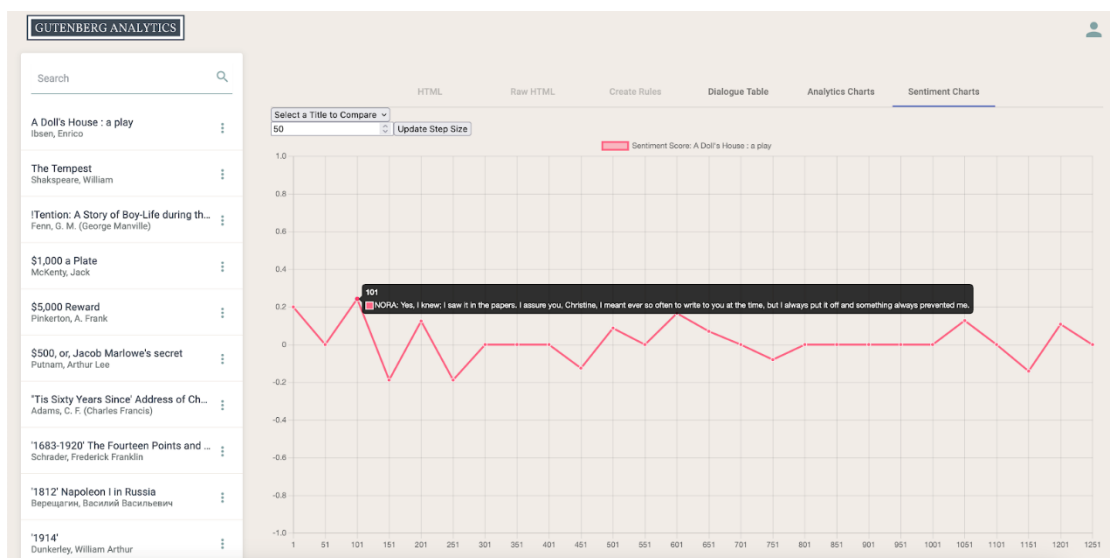


Figure 24 - Sentiment Timeline Chart With Custom Step Size

### 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.

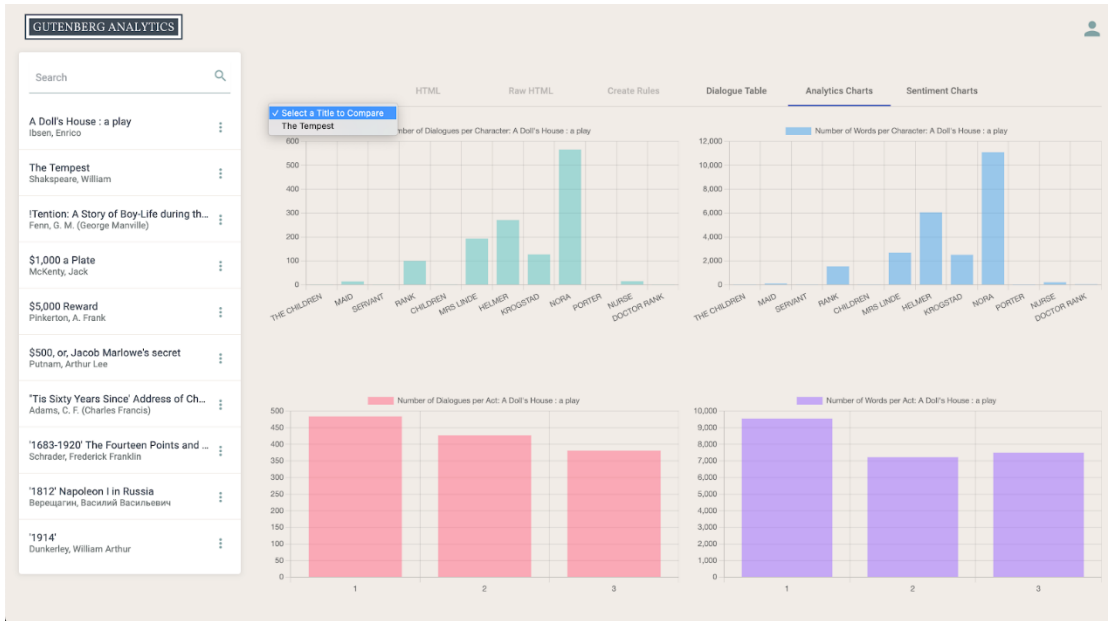


Figure 25 - Analytics Charts Comparison Dropdown Menu



Figure 26 - Analytics Charts With Comparison (top)

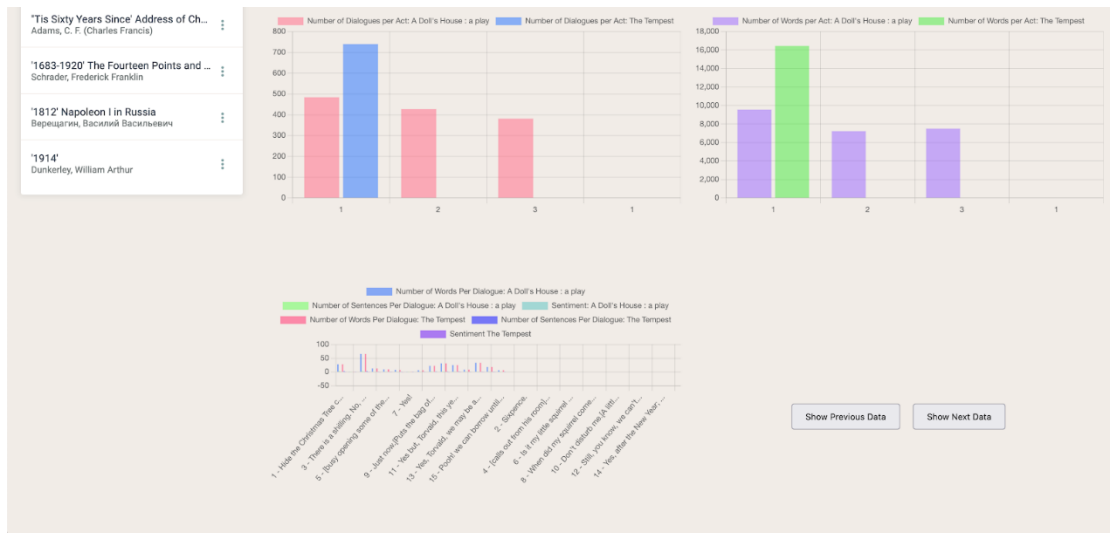


Figure 27 Analytics Charts With Comparison (down)

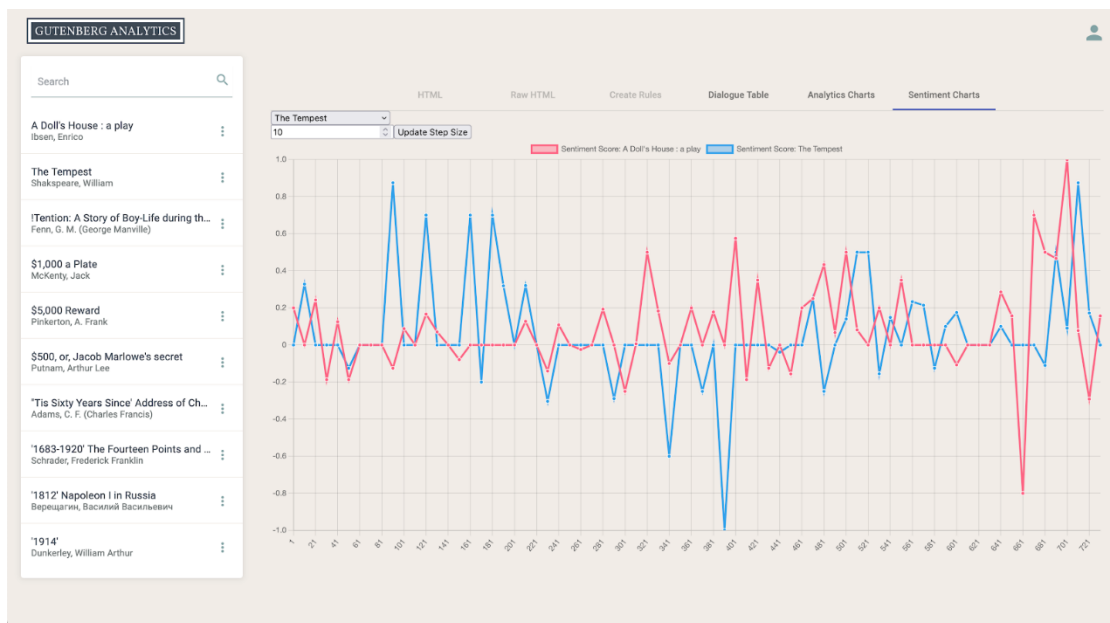


Figure 28 - Sentiment Timeline With Comparison

## 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 app were well-defined by the professor in the thesis description. However, as the development progressed, we engaged in discussions during sprint plannings 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 app 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 app's user interface. These designs were reviewed and refined based on considerations of user experience, ensuring that the app'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 through the use of 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 app's features in incremental iterations.

### ***3.6.5 Data Collection and Preprocessing***

To populate our app 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 app'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 app'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, our app 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 app 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 app allowed 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 app'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 app'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 app's versatility and usefulness.

Overall, despite these challenges and limitations, the web app 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 app'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 app 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.

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.
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.
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. Did the provided analytics enhance your understanding of the plays?
  - 95% of the respondents answered "Yes," while 5% answered "No." This suggests that the majority of participants (95%) found the provided analytics to enhance their understanding of the plays, while a small portion (5%) did not feel the same way.
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.
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 our 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. The following sections outline these potential enhancements:

#### Advanced Metadata Extraction Techniques:

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.

#### Semantic Analysis for Metadata Extraction:

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.

Metadata Enrichment:

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 our 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:

Data Processing Efficiency:

As the volume of HTML documents processed by our 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 and Indexing:

Caching plays a crucial role in improving the responsiveness and efficiency of our 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 our 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 our 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 our 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 our application continues to deliver exceptional performance as it evolves.

### ***5.3 Additional Improvements***

Beyond metadata extraction and performance optimization, there are other areas where our application can be further improved. Let us look at two such areas: user interface enhancements and integration with external tools and platforms:

#### User Interface Enhancements:

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.

#### Integration with External Tools and Platforms:

Integrating our 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 our 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]

#### Continuous Performance Monitoring and Optimization:

Establish a systematic process for monitoring the performance of our 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 our 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 our 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* (pp. xxx-xxx).