



# GUTEN-BOT

Group 06



# Problem Description

## Objective:

Establish an interactive platform for exploring classic book summaries from "Project Gutenberg." provide users with concise yet insightful summaries of classic literature, making it more accessible and engaging.

## Summarization Component:

- Generate 200-300 word summaries for selected classic books.
- easy-to-read summaries of classic books
- Foster a more enjoyable reading experience by presenting key ideas and themes

## Chatbot Interface:

- Implement a user-friendly chatbot interface for deeper engagement.
- Allow users to interact with the chatbot for additional information beyond the summaries.

# Problem Description

## Empowering Accessibility:

Establish an interactive platform for exploring classic book summaries from "Project Gutenberg."

Enable users to access condensed yet insightful summaries of classic books.

## Validation and Enhancement

Using the evaluation metric 'Rouge' , and the time consumed, we evaluated the trade-off of each summarizing methodologies.

# Data Collection & Preparation

## Data Collection

Obtain classic book texts from the provided data source Project Gutenberg, which provides over 70,000 free ebooks.

- Curate diverse classics for comprehensive coverage.
- Collect substantial data to ensure variety.
- extracting the content of a book from the "Project Gutenberg" website using Python and the BeautifulSoup library
- search for a book by name, find the best match based on the similarity ratio, retrieve the plain text link

There are over 70,000 classics on the Project Gutenberg site.

## Preparation

Locate and access the plain text version of each book

### **Metadata Identification:**

1. Identify Metadata: Recognize elements that constitute metadata, such as book title, author's name, and licensing information, Project Gutenberg project details
2. Exclusion of Metadata: Focus on extracting the main content while excluding metadata to ensure clean data for summarization.

### **Start Marker Identification:**

- Use regular expressions to identify possible start markers that indicate the beginning of the book content.
- If a start marker is found, extract the book name and starting index for content extraction.

### **End Marker Identification:**

- Define a list of possible end marker patterns to accommodate variations in the end marker.
- Search for the end marker using regular expressions within the identified content range.
- Once found, capture the end index for content extraction.

## Preparation

### **Content Extraction:**

- Utilize the start and end indices to extract the text content that needs to be summarized.
- This extracted content represents the main body of the book text for summarization.

# Methodology

## Extracting and Summarizing the book content

### 1. **Search and Identification:**

- Utilize the BeautifulSoup library to extract book information from the "Project Gutenberg" website.
- Search for a book by name to find the best match URL using search functionality.

### 2. **Start Marker Identification:**

- Define a list of possible start markers using regular expressions.
- Match the start markers within the book's plain text content to identify the beginning of the main text.
- Capture the start index for content extraction.

# Methodology

Extracting and Summarizing the book content

## 3. End Marker Identification:

- Define a list of possible end marker patterns to accommodate variations in the end marker.
- Search for the end marker within the identified content range using regular expressions.
- Capture the end index for content extraction.

## 4. Content Extraction:

- Utilize the start and end indices to extract the main text content that requires summarization.
- Isolate the book's main content by excluding any metadata, start markers, and end markers.



# Methodology

Extracting and Summarizing the book content

## 5. Summarization Process with Pipelined Models

### 1. Initial Large Summary:

- We start with a comprehensive summary generated from the original, extensive text, the content of the book. This initial summary is produced using a model specifically designed for book summarization, known as "led-large-book-summary."

### 2. Hugging Face Transformers:

- Our approach leverages the Hugging Face Transformers library, a leading platform for natural language processing models.
- With this library, we create a summarization pipeline using the same "led-large-book-summary" model.

# Methodology

## Extracting and Summarizing the book content

### 3. Input Text:

- The text we want to summarize, often spanning thousands of words, is stored in a variable known as "wall\_of\_text.", which is the output string of the previous large summary. This serves as our second starting point.

### 4. The Summarization Process:

- We apply the summarization pipeline to the "wall\_of\_text."
- Parameters for summarization are set, including the minimum and maximum length of the summary and constraints on repeated n-grams (phrase patterns).
- Multiple beams are employed for generating summaries, ensuring that the final output is coherent and informative.

### 5. Resulting Concise Summary:

- The outcome of this process is a concise summary that distills the most essential information from the original extensive text.

# Methodology

## ChatBot

### 1. Search and Identification:

- Same as in Summarizing
- Utilize the BeautifulSoup library to extract book information from the "Project Gutenberg" website.
- Search for a book by name to find the best match URL using search functionality.

### 2. Data Loading and Pre-processing:

- Document loaders like "TextLoader" were utilized to load book data efficiently.
- Data pre-processing involved cleaning and formatting text data for optimal performance.

# Methodology

ChatBot

## 3. Language Models (LLMs):

- To generate a coherent and context-aware text we are currently using **“llama2-13b”** LLM
- Other LLMs that we have identified
  - wizardlm
  - llama2-7b
  - bloom
  - falcon'

## 4. Vector Embeddings:

- Used the **HuggingFaceInstructEmbeddings** to load the instruction-finetuned text embedding model, **“hkunlp/instructor-base”** to facilitate vector embeddings, and representing text data in a high-dimensional space.
- Embeddings significantly improved the accuracy of search and retrieval operations.

# Methodology

## ChatBot

### 5. FAISS and Search Retrieval:

- FAISS is a vector similarity search library, was utilized for efficient retrieval of relevant passages.
- The integration of FAISS enhanced the chatbot's ability to provide contextually accurate answers.

### 6. Prompt Templates and QA Chains:

- Custom prompt templates guided the chatbot's responses based on user input and context.
  - Answers should only be based on the book provided
- QA chains enabled the seamless integration of LLMs with prompt templates for robust responses.

# Evaluation of Models

## Accuracy Metric- ROUGE

1. Reference and Generated Summaries: Our evaluation begins by defining two main text summaries: a reference summary and a generated summary. These summaries represent different versions of a text, with the reference summary typically being a human-generated summary that serves as the gold standard.
2. Splitting Sentences: To perform a detailed evaluation, the reference and generated summaries are split into individual sentences. This is crucial for sentence-level evaluation, as ROUGE assesses how well the generated sentences match the reference sentences.

# Evaluation of Models

## Accuracy Metric- ROUGE

1. ROUGE Scores Calculation: We then proceed to calculate ROUGE scores for each sentence. It iterates through the reference sentences and, for each reference sentence, calculates the maximum ROUGE score achieved when compared to the generated sentences. The maximum score is retained.
2. Average ROUGE Score: After computing ROUGE scores for all reference sentences, the code calculates the average ROUGE score. This average score represents the overall similarity or quality of the generated summary compared to the reference summary.
3. Printing the Result: Finally, your code prints the average ROUGE score to the console, allowing you to interpret the quality of the generated summary based on the ROUGE metric.

# User Interaction

## Design Philosophy

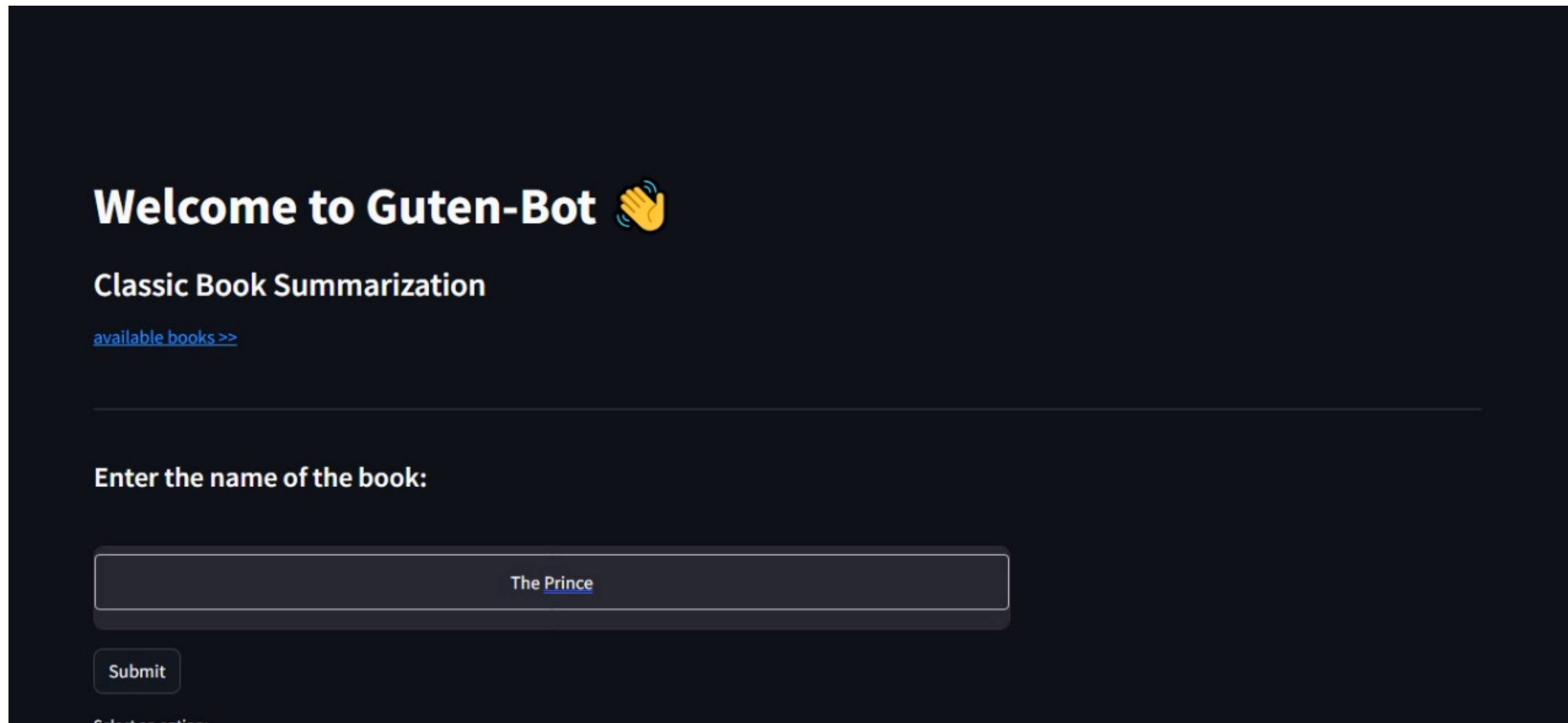
**Our user interface was meticulously designed to provide a seamless and enjoyable experience for users. We've prioritized the following design principles:**

- **Selection of Web Application Framework:** We chose Streamlit, a web application framework, as the foundation for our user interface. Streamlit is known for its ease of use and rapid development capabilities.
- **User-Centric Design:** The interface revolves around user needs, making it easy for users to interact.
- **Visual Appeal:** We've focused on aesthetics to create an engaging and visually pleasing environment.



# User Interaction

First enter the name of the book and click "Submit"

A screenshot of a web application interface for 'Guten-Bot'. The interface has a dark blue background. At the top, it says 'Welcome to Guten-Bot' with a yellow hand icon. Below that, it says 'Classic Book Summarization' and has a link 'available books >>'. There is a horizontal line. Below the line, it says 'Enter the name of the book:'. There is a text input field containing 'The Prince'. Below the input field is a 'Submit' button. At the bottom left, there is a small text 'Select an option:'.

**Welcome to Guten-Bot** 🖐️

Classic Book Summarization

[available books >>](#)

---

Enter the name of the book:

Select an option:

# User Interaction

Select the options as 'Get Submit' or 'Get Answer'

## Classic Book Summarization

[available books >>](#)

---

Enter the name of the book:

The Prince

Submit

Status: success

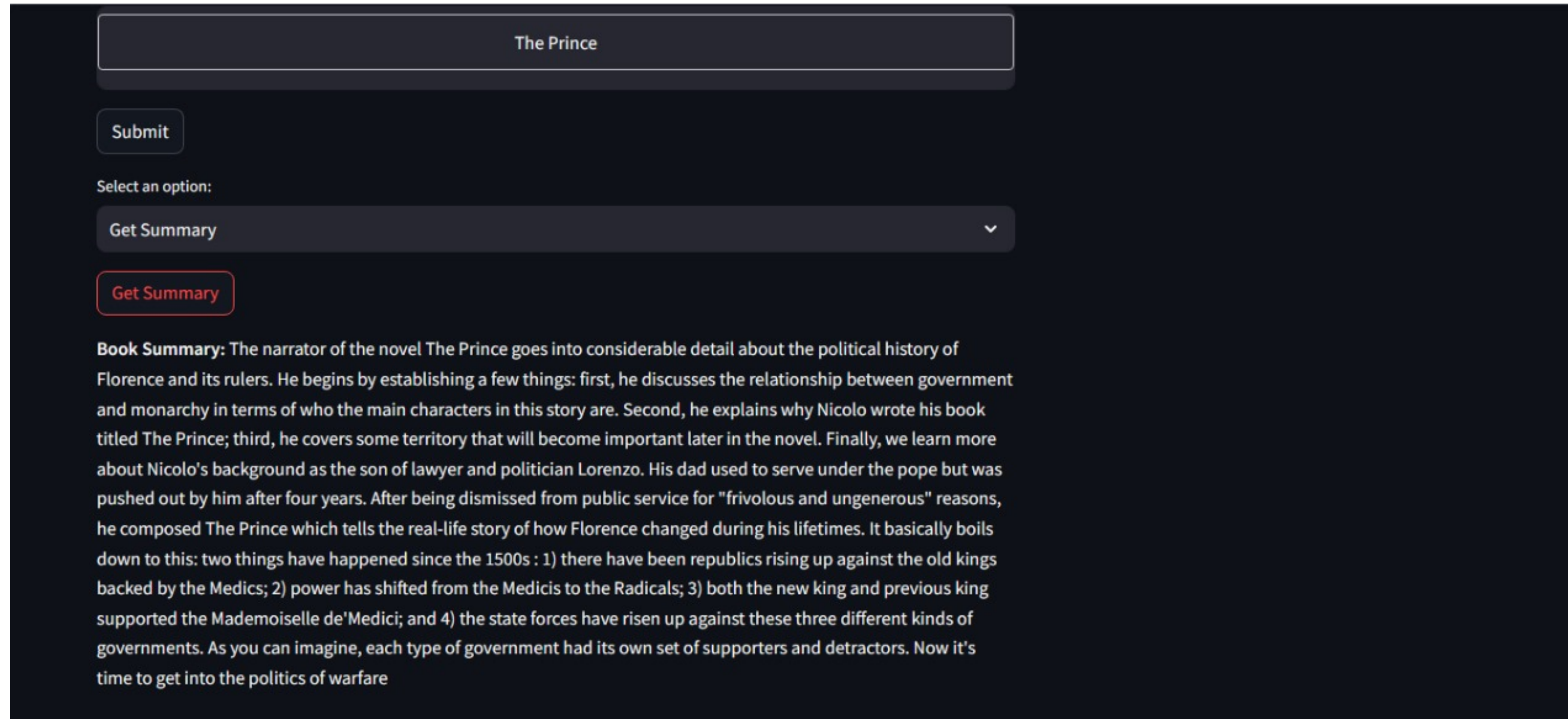
Select an option:

Get Summary

Get Summary

# User Interaction

## Get the summary by click 'Get Summary' button



The screenshot shows a dark-themed web application. At the top, there is a search bar containing the text "The Prince". Below the search bar is a "Submit" button. Underneath the submit button is a label "Select an option:" followed by a dropdown menu that currently displays "Get Summary" with a downward arrow. Below the dropdown menu is a button labeled "Get Summary" in red text, which is highlighted with a red border. Below this button is a paragraph of text providing a book summary for "The Prince".

The Prince

Submit

Select an option:

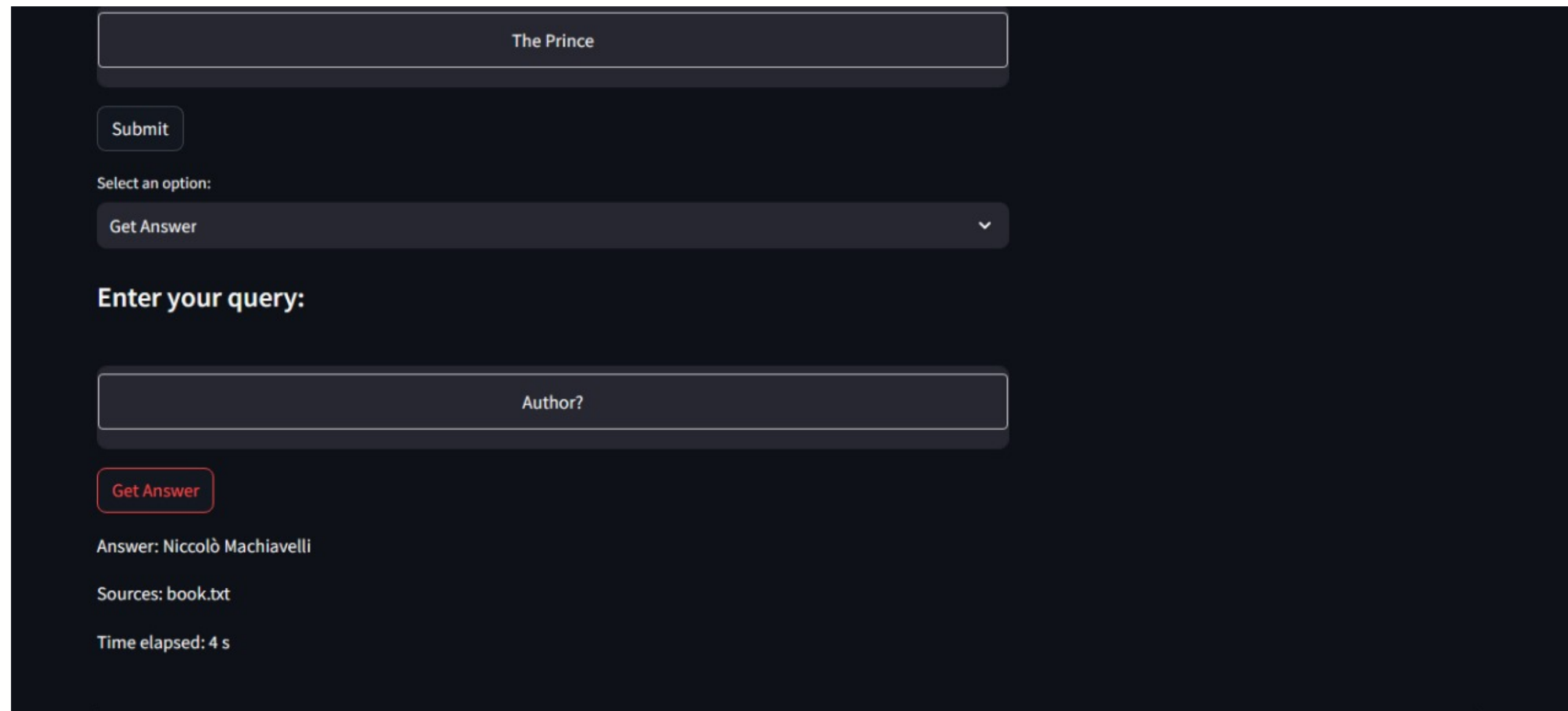
Get Summary

Get Summary

**Book Summary:** The narrator of the novel The Prince goes into considerable detail about the political history of Florence and its rulers. He begins by establishing a few things: first, he discusses the relationship between government and monarchy in terms of who the main characters in this story are. Second, he explains why Nicolo wrote his book titled The Prince; third, he covers some territory that will become important later in the novel. Finally, we learn more about Nicolo's background as the son of lawyer and politician Lorenzo. His dad used to serve under the pope but was pushed out by him after four years. After being dismissed from public service for "frivolous and ungenerous" reasons, he composed The Prince which tells the real-life story of how Florence changed during his lifetimes. It basically boils down to this: two things have happened since the 1500s : 1) there have been republics rising up against the old kings backed by the Medics; 2) power has shifted from the Medicis to the Radicals; 3) both the new king and previous king supported the Mademoiselle de'Medici; and 4) the state forces have risen up against these three different kinds of governments. As you can imagine, each type of government had its own set of supporters and detractors. Now it's time to get into the politics of warfare

# User Interaction

Get the Answer of the query by clicking 'Get Answer' button



The screenshot shows a dark-themed web interface. At the top, there is a text input field containing "The Prince". Below it is a "Submit" button. Underneath the button is the text "Select an option:" followed by a dropdown menu that currently displays "Get Answer" with a downward arrow. Below the dropdown is the text "Enter your query:". This is followed by another text input field containing "Author?". Below this field is a "Get Answer" button, which is highlighted with a red border. At the bottom of the interface, there are three lines of text: "Answer: Niccolò Machiavelli", "Sources: book.txt", and "Time elapsed: 4 s".

The Prince

Submit

Select an option:

Get Answer

Enter your query:

Author?

Get Answer

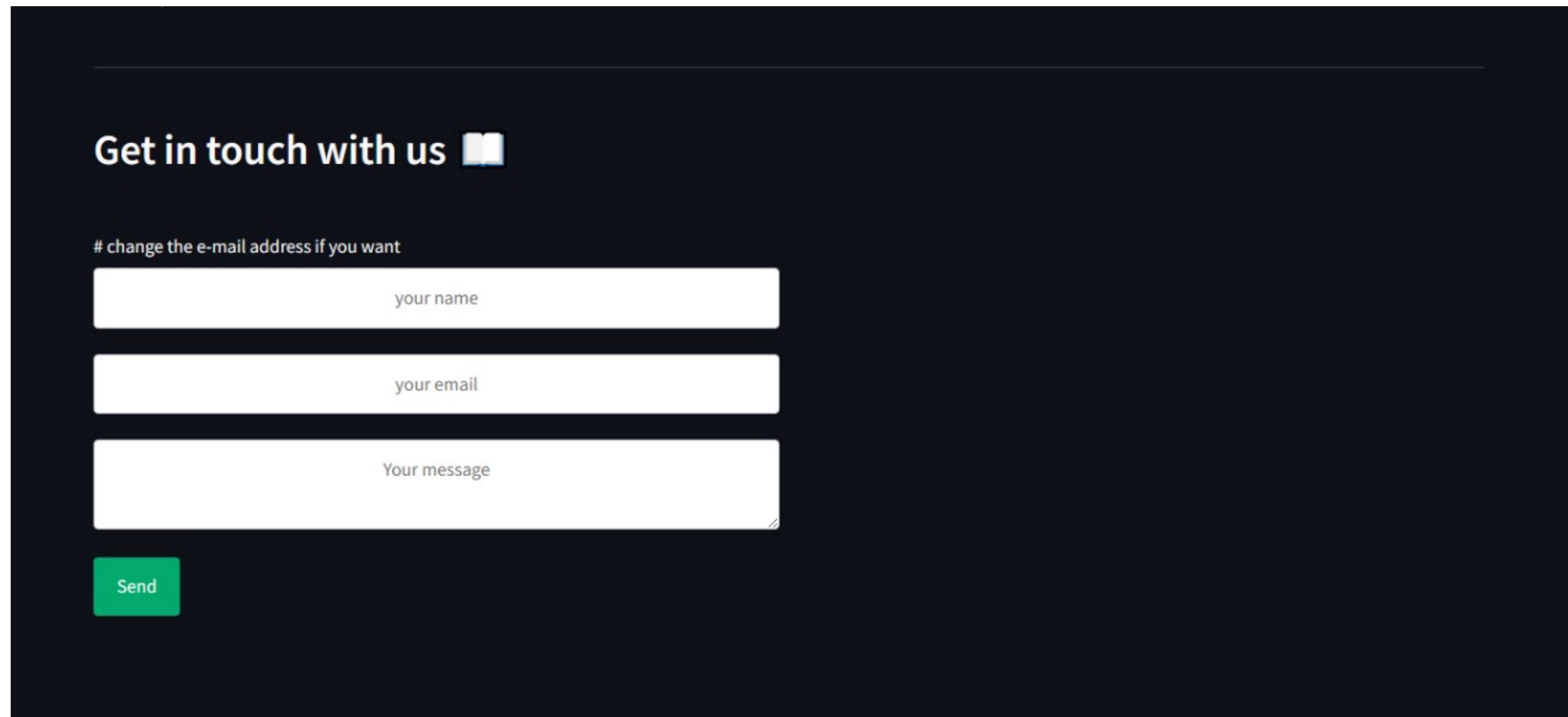
Answer: Niccolò Machiavelli


Sources: book.txt

Time elapsed: 4 s

# User Interaction

Add the feedback message at the bottom of the page



Get in touch with us 

# change the e-mail address if you want

# User Interaction

## Functionality Highlights

The user interface offers the following key functionalities:

- Book Summarization: Users can enter the name of a book and receive a concise summary, facilitating understanding of the book's content.
- Chatbot Interaction: With our chatbot feature, users can ask questions or seek information related to specific books.
- User Feedback: We have implemented a feedback system to gather user input and continuously improve the platform.

# Further Improvements

## Platform Improvement

Ensure the model's integration is seamless and responsive, delivering timely and contextually relevant responses.

We hope to use further user feedback to utilitate this process.

# Thank You

## Group memebers

200251X-Jayasekara J.K.A.M.P

200269J-Jayaweera U.D.

200276D-Johnson S.