**A PROJECT REPORT ON**

**DraftCheck - Web Application for Article Analysis**

***Major project submitted in partial fulfillment of the requirements for the***

***award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**(2017-2021)**

**By**

**R. LOHITAA 17241A12G4**

**K. DEEPTHI REDDY 17241A12D3**

**SREETHI MUSUNURU 17241A12E3**

**J. PADMA PRIYA 17241A12D1**

***Under the esteemed guidance of***

**Dr. N. Rajasekhar**

**Professor, Dept of IT**

******

**Department of Information Technology**

**Gokaraju Rangaraju Institute of Engineering and Technology**

**(AUTONOMOUS)**

**Hyderabad**



**CERTIFICATE**

This is to certify that it is a bonafide record of Major Project work entitled **“Draft Check - Web Application for Article Analysis”** done by **Rayala Lohitaa(17241A12G4), K.Deepthi Reddy(17241A12D3),M.Sreethi(17241A12E3),J.Padma Priya(17241A12D1),** students of **B.Tech (IT)** in the Department of Information Technology, Gokaraju Rangaraju Institute of Engineering and Technology during the period 2017-202 in the partial fulfillment of the requirements for the award of degree **of BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from GRIET,Hyderabad.

Dr. N. Rajasekhar Dr. K. Prasanna Lakshmi

(Project Guide) (Head of the Department)

(Project External)

# **ACKNOWLEDGEMENT**

We take immense pleasure in expressing gratitude to our External guide **Dr. N. Rajasekhar, Professor,** Information Technology, GRIET.We express our sincere thanks for her encouragement, suggestions and support, which provided the impetus and paved the way for the successful completion of the project work.

We would like to thank **Dr. K. Prasanna Lakshmi, P. Gopala Krishna,** our Project Co-coordinators **G. Vijender Reddy and Dr. V. Akila,** for their constant support during the project.

We express our sincere thanks to **Dr. Jandhyala N Murthy,** Director, GRIET**, and Dr. J. Praveen,** Principal, GRIET, for providing us the conductive environment for carrying through our academic schedules and project with ease.

We also take this opportunity to convey our sincere thanks to the teaching and non-teaching staff of GRIET College, Hyderabad.

Email : [lohitaarayala2000@gmail.com](mailto:lohitaarayala2000@gmail.com) Email:[deepthireddy2462000@gmail.com](mailto:deepthireddy2462000@gmail.com)

Contact No : 9182352227 Contact No : 9121432159

Address : Nizampet, Hyderabad Address : Attapur, Hyderabad

**** 

Email : [musunurusreethi@gmail.com](mailto:musunurusreethi@gmail.com) Email : jankalapriya9@gmail.com

Contact No : 8332813351 Contact No : 9885473300

Address : Sanathnagar, Hyderabad Address : Kompally, Hyderabad

# **DECLARATION**

This is to certify that the project entitled **“Draft Check - Web Application for Article Analysis”** is a bonafide work done by us in partial fulfillment of the requirements for the award of the degree **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad**.**

We would like to declare that this project is a result of our own efforts and has not been copied or imitated from any source. Citations from any websites, books and paper publications are mentioned in the Bibliography.

This Project was not submitted earlier at any other University or Institute for the award of any degree.

**Rayala Lohitaa 17241A12G4**

**K. Deepthi Reddy 17241A12D3**

**M. Sreethi 17241A12E3**

**J. Padma Priya 17241A12D1**

**TABLE OF CONTENTS**

| **Serial No.** | **Name** | **Page No.** |
| --- | --- | --- |
|  | **Certificates** | **ii** |
|  | **Contents** | **v** |
|  | **Abstract** | **vii** |
| **1** | **INTRODUCTION** | **1** |
| 1.1 | Introduction to Project | 1 |
| 1.2 | Project Overview | 1 |
| 1.3 | Existing System | 2 |
| 1.4 | Proposed System | 2 |
| **2** | **REQUIREMENTS ENGINEERING** | **4** |
| 2.1 | Hardware System Configuration | 4 |
| 2.2 | Software System Configuration | 4 |
| **3** | **LITERATURE SURVEY** | **5** |
| **4** | **DESIGN ENGINEERING** | **6** |
| 4.1 | Introduction to UML | 6 |
| 4.2 | UML Diagrams | 6 |
| 4.2.1 | Use Case Diagram | 7 |
| 4.2.2 | Activity Diagram | 8 |
| 4.2.3 | Sequence Diagram | 9 |
| 4.2.4 | Collaboration Diagram | 10 |
| 4.2.5 | Statechart Diagram | 11 |
| **Serial No.** | **Name** | **Page No.** |
| **5** | **METHODOLOGY** | **12** |
| 5.1 | Approach | 12 |
| 5.2 | API | 13 |
| 5.3 | Anaconda Navigator | 15 |
| 5.4 | Flask in Python | 15 |
| **6** | **IMPLEMENTATION** | **18** |
| 6.1 | app.py | 20 |
| 6.2 | Summarizer.html | 23 |
| 6.3 | Wordcount.html | 24 |
| 6.4 | Keywords.html | 24 |
| 6.5 | Grammarcheck.html | 25 |
| 6.6 | Home.html | 25 |
| **7** | **TESTING** | **27** |
| 7.1 | Test objectives | 28 |
| 7.2 | Test results | 28 |
| **8** | **RESULTS** | **29** |
| **9** | **CONCLUSION** | **34** |
| **10** | **FUTURE WORK** | **35** |
| **11** | **BIBLIOGRAPHY** | **36** |

**12. LIST OF FIGURES**

| **Serial No.** | **Figure Name** | **Page No.** |
| --- | --- | --- |
| **12.1** | Use Case Diagram | 7 |
| **12.2** | Activity Diagram | 8 |
| **12.3** | Sequence Diagram | 9 |
| **12.4** | Collaboration Diagram | 10 |
| **12.5** | Statechart Diagram | 11 |

**ABSTRACT**

Digitisation is the process of converting information from physical formats into a digital one so that we can increase data quality and apply advanced techniques like summarising, extracting keywords etc. Text or article summarization refers to the technique of lessening long pieces of text. The intention is to create a well organized and fluent summary having only main points outlined in the document. Similarly, the keyword extraction process not only separates the articles but also helps in saving time when we are searching for a document. On the other hand, plagiarism checking helps to evaluate our articles and documents and generate reports on the suspicious statements that are possibly plagiarised. Also when there is a limit on the number of words that can be used in a document, we need the help of a word counter. Manually summarising the articles, extracting keywords, counting words or testing for plagiarism is a time consuming process and a waste of human resources. DraftCheck provides a single platform for all the analysis tasks of articles.

This project focuses on using flask to provide a framework for the web application, natural language processing to perform tasks like summarising, keyword extraction etc, web designing and services to maintain the application and other techniques to achieve the article analysis. In order to provide a better user experience, a software intermediary or an api is used so that a smooth communication is achieved between various parts of the application. The main aim of this project is to increase the accuracy of the existing systems and provide a one stop solution to the user requirements. Adding other functionalities and customising the application to make it more user friendly can be considered as the future work for this project.

**1.INTRODUCTION**

**1.1 INTRODUCTION TO PROJECT**

As the amount of public content increases, so does the difficulty of reading it in a limited span of time. Moreover, digesting all of this content in a meaningful fashion requires time and effort. An efficient way of skimming through the content is to read summaries, if available. This application helps you in getting the summary of long articles which can save your time. The article can be directly pasted in the textbox to get a summary of it.

DraftCheck also helps in keeping the author at check when it comes to maintaining a word limit while writing articles. Another feature of our application is word counter and it helps calculating the number of words present in the article. The content can be pasted in the textbox provided on the webpage.

There’s also a need to identify keywords in an article especially when it comes to writing research papers, thesis or dissertations. By providing the keywords, the author can help the readers identify the important concepts or areas covered in the article beforehand or while taking notes.

**1.2 PROJECT OVERVIEW**

Although there are a plethora of platforms available that can perform various article analysis activities, not most of them provide accurate results. At the same time not many tools or systems are available online where we can perform all of the article analysis activities at one place.

The project allows the user to choose an operation that he/she wishes to perform from among those provided by the application. Then on providing the parameters on which the operation is to be performed, the result will be returned and displayed.

This project focuses on using flask to provide a framework for the web application, natural language processing to perform tasks like summarizing, keyword extraction etc. In order to provide a better user experience, a software intermediary or an API is used so that a smooth communication is achieved between various parts of the application. The main aim of this project is to increase the accuracy of the existing systems and provide a one stop solution to the user requirements.

**1.3 EXISTING SYSTEM**

The existing systems have a different approach towards analysing articles. Through our sources we have observed that the existing applications cater to very specific needs such as counting words or extracting keywords.

Sites like Smmry, Resoomer, Scholarcy etc. do the job of summarising articles but we aim at providing more accurate summaries along with other ways of analysing the articles.

Cortical, MonkeyLearn, Comprehend etc are the tools available online for keyword extraction. Keyword tool, easywordcount, wordcountertool etc are the sites available online for counting words. Duplichecker, smallseotools, grammarly etc provide a platform for plagiarism checking.

But when it comes to an application that has multiple functionalities and could serve as the one stop solution for finalising or improving drafts, thesis or dissertations, there are not a lot of applications on the web that are doing it efficiently.

**1.4 PROPOSED SYSTEM**

The proposed system focuses on providing a single application that provides various article analysing features. Our application will first require the user to select one of the options of analysing. Next, the user will be requested to provide the url of the website he/she is using as a source or paste the article itself in the textbox. The following are the operations available on the system:

1. Summarization: An Adaptive Text Summarization API will be used to summarize the article
2. Keyword Extraction: We will use NLP to extract keywords from the provided article
3. Word Counter: The application will require a python program for word counter
4. Digitization: We will be using a Smart OCR for digitization of the documents
5. Plagiarism Check: Plagiarism checker can be developed using the Google Search API

6. Spell Checking: We will be using Python libraries to achieve this.

After the operation is performed on the user’s article, the result of that operation is returned to the user browser.

**Limitations Of Existing System**

Consideryou have written an article and your article is expected to contain 1000 words, you should list the key terms in your article, your article should be free of plagiarism and then give a summarised report of your article. In order to perform all the above operations, you have to visit four different sites if you use the existing system.

This is a major drawback of the existing. There are tools like Grammarly that perform almost all of the grammar checking activities at one place. But there are not many online tools that perform more than 2 article analysis tasks at one place.

**Objective**

The main objective of this project is to provide various article checking features at one place. It also aims at increasing the accuracy of the individual operations.

This project also aims at providing a user-friendly interface and a supreme experience which not only makes it easy for the user to access the web application but also keeps him coming back to the application whenever he has to perform article analysis tasks.

**2. REQUIREMENTS ENGINEERING**

**2.1 HARDWARE SYSTEM CONFIGURATION**

* Processor- Core i3
* Hard Disk- 160 GB
* Memory- 1 GB RAM
* Monitor
* Camera

**2.2 SOFTWARE SYSTEM CONFIGURATION**

* Anaconda Navigator
* Flask
* Python
* Python Web Frameworks
* GitHub
* Windows 7 or higher

**3. LITERATURE SURVEY**

Examining the articles is an important task. There are about 30,000 to two million articles published every year. There are an overwhelming number of articles published in every field each year. One of the main tasks of any research scholar who wishes to publish an article should make sure that there is no other article published before which includes the same content as theirs. They should also make sure that their article follows certain standards set beforehand. This is the main reason that makes article examination important. There are many research papers published that interpret these article examining operations.

The journal, Detecting The Plagiarism For Text Documents On The World Wide Web by Durga Bhavani Dasari and Dr. Venu Gopal Rao. K discusses various techniques and methods that detect and prevent plagiarism in articles, journals and publications. The research paper, Text Summarization: A Brief Review, by Laith Abualigah, Mohammed Shehab and others discusses the most recent and relevant research in the field of the text summarization to study and analysis for future research. The research paper, An Empirical Study of Important Keyword Extraction Techniques from Documents, discusses the use of various algorithms in the extraction of keywords.

There are many journals, research papers and publications on the topics of text summarization, keyword extraction, plagiarism checkers and word counters. After surveying the existing work on these processes it was understood there is a gap in the applications available as they don't provide a single platform. This project is developed to fill these gaps.

**4. DESIGN ENGINEERING**

**4.1 INTRODUCTION TO UML**

The Unified Modeling Language is a general-purpose, developmental, modeling language in the field of software engineering that is meant to provide a standard way to observe the design of a system. It's an international industry which is used for describing, visualizing, constructing and documenting the process of a software system.

**4.2 UML DIAGRAMS**

UML diagrams are used to communicate different characteristics of a system. UML diagram serves as a complete design that requires the actual implementation of the software.The purpose of UML diagram is to read about the system behavior, to detect errors and omit them, to present the designs and communicate with stakeholders, to understand requirements and to implement them. They provide a more systematized way of modeling workflows and also a wider range of features to improve readability and proficiency. There are several types of UML diagrams and each one of them have a different purpose of regardless of whether it is being designed before the implementation or after(as part of documentation):

**4.2.1 Use Case Diagram**

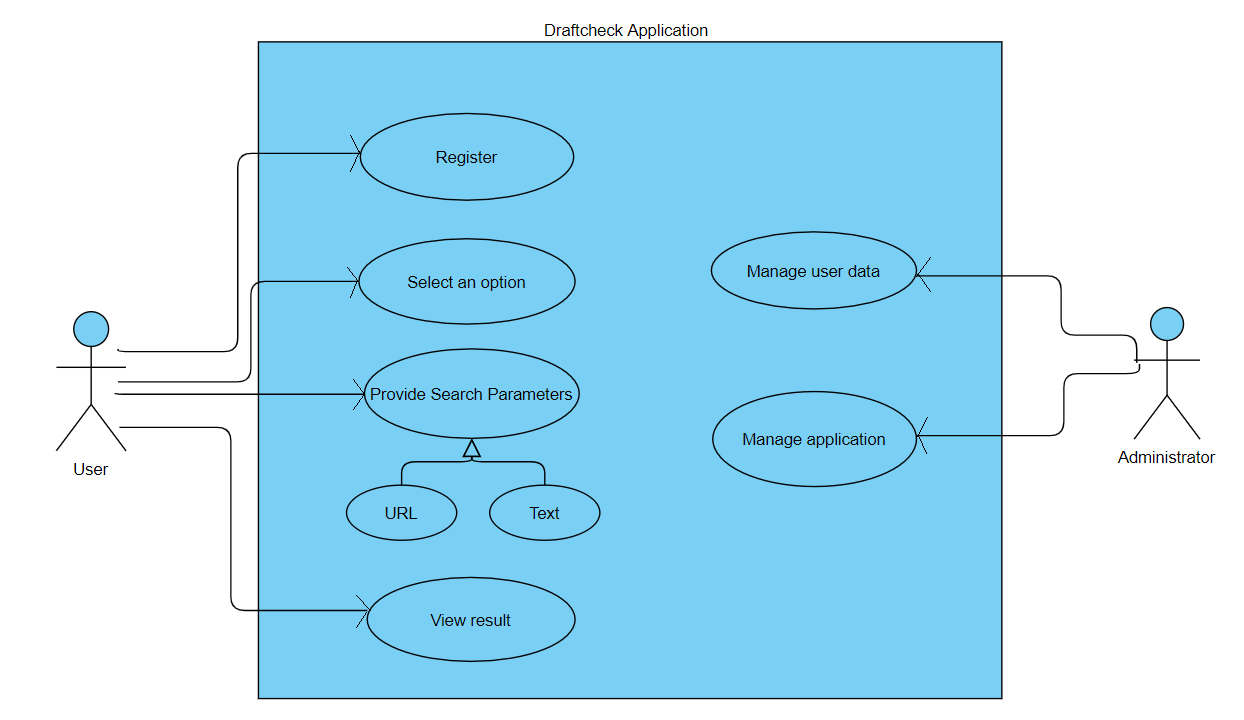
A use-case diagram is used to model dynamic behavior of the system. It contains a set of actors and use-cases.

**Actors:**

User and Administrator

**Use-cases:**

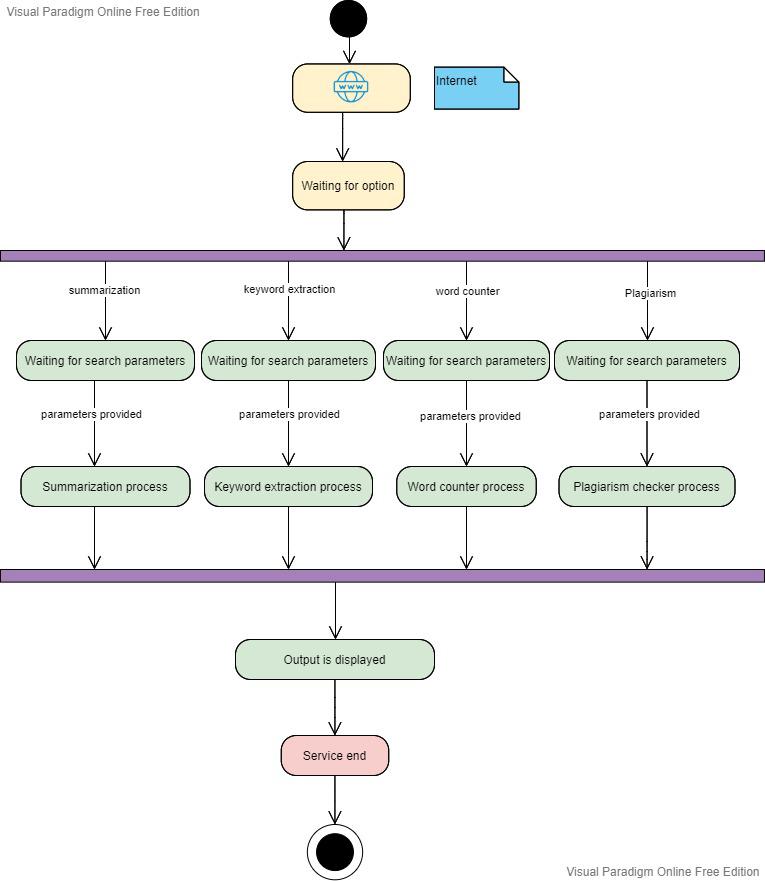
Register, Choose an operation, Give parameters, View the results, Manage user data and Manage the application.



4.2.1 Use Case Diagram

**4.2.2 Activity Diagram**

An activity diagram is a behavioral diagram. It depicts the control flow from start point to a finish point showing various decision paths that exist while an activity is being executed.

**The activities in this system are:**Waiting for an option, Waiting for search parameters, Performing the operation, Displaying the output.

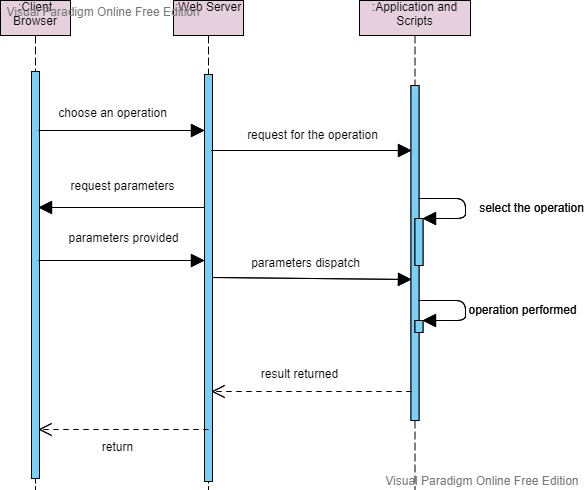
4.2.2 Activity Diagram

**4.2.3 Sequence Diagram**

A sequence diagram is an interaction diagram. It shows the interaction between objects in a sequential order. It contains lifelines, objects and messages exchanged between them.

**The objects in this system are:**

Client browser, Web server, Application and scripts.



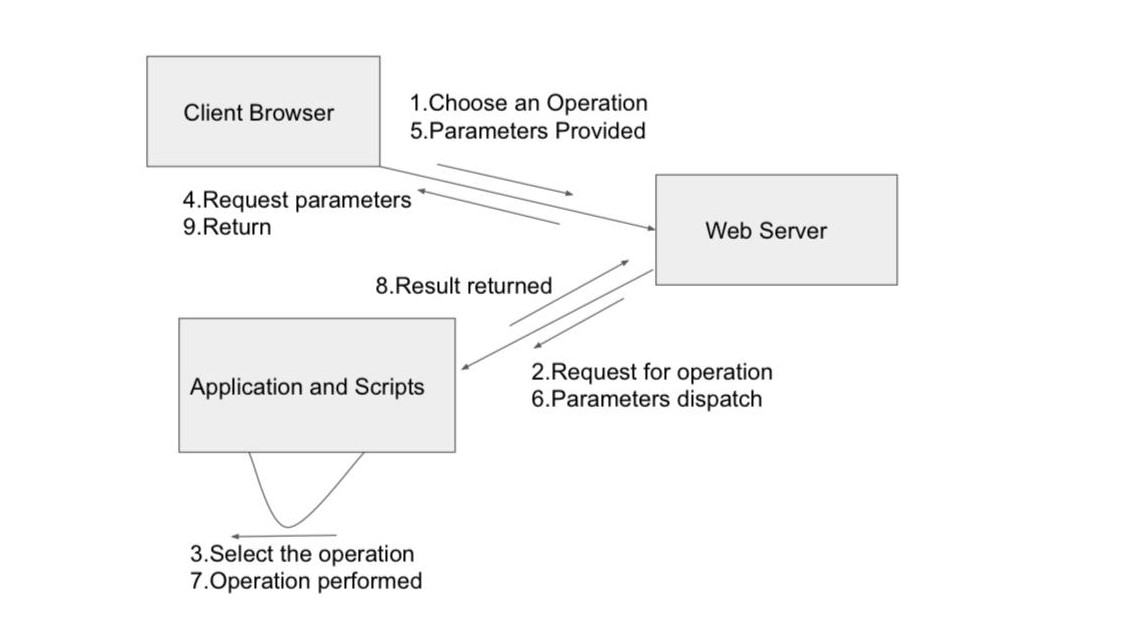
4.2.3 Sequence Diagram

**4.2.4 Collaboration Diagram**

Collaboration diagram is also an interaction diagram. It illustrates the relationships and interactions between software objects. The major elements of a collaboration diagram are objects, links and messages.

**The objects in this system are:**

Client browser, Web server, Application and scripts.



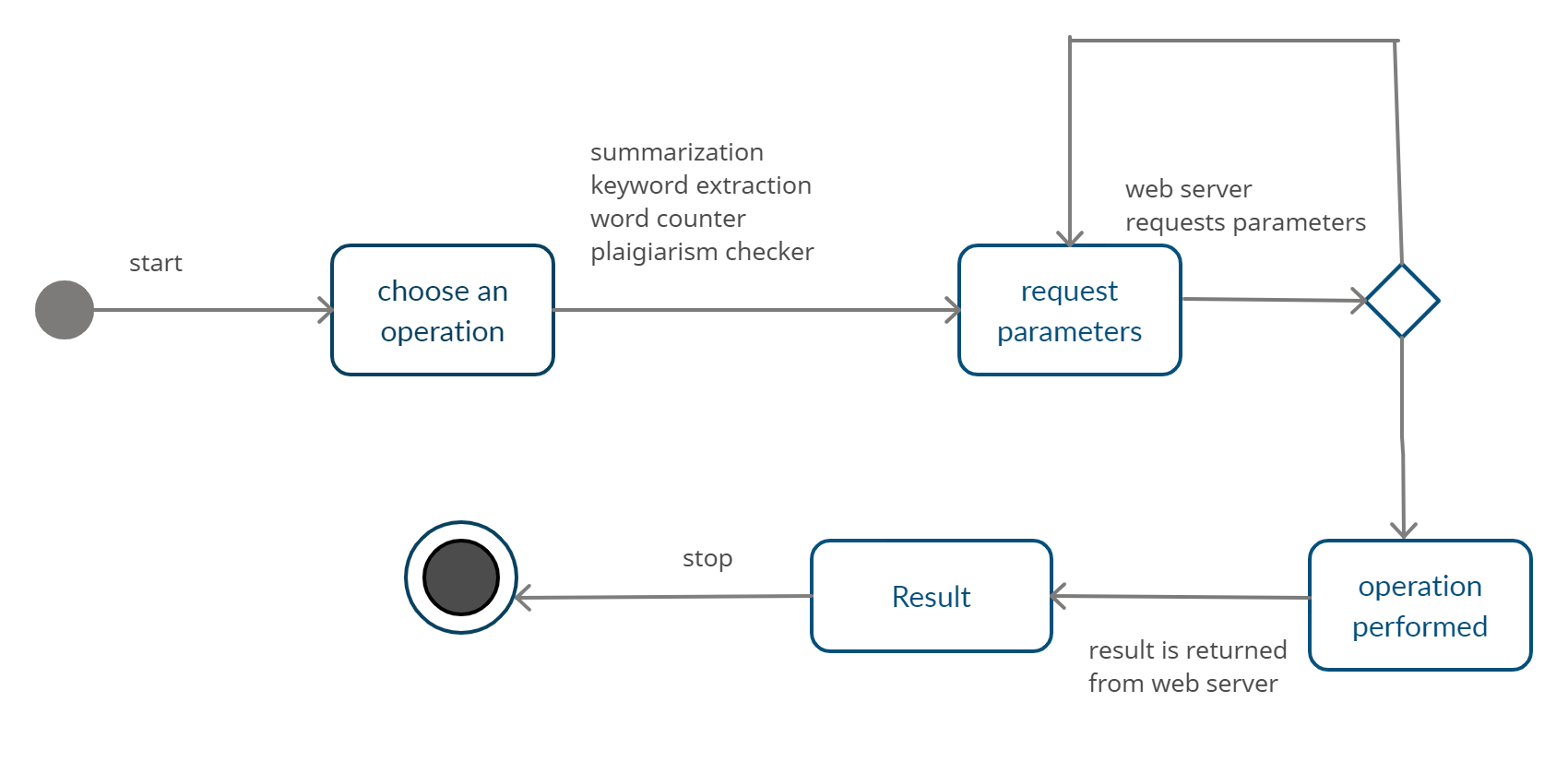
4.2.4 Collaboration Diagram

**4.2.5 Statechart Diagram**

A statechart diagram models the dynamic nature of the system. It defines different states of objects during its lifetime and the events that trigger these changes.

**The main elements of statechart diagram are:**

Events, States and Transitions.



4.2.5 Statechart Diagram

**5. Methodology**

**5.1 APPROACH**

* The approach at how Draft Check, the website will be designed begins with the understanding that there is an abundance of public content available and it will keep increasing by the minute, there is a need to make it easier for students and various academicians who have to go through tons of information just to find a few relevant points.
* With this understanding, we have identified a few features that could be a part of our application that would cater to our goal of building an application that would be a one-stop solution for various academic and non-academic article analysis situations.
* We have conducted market-research to understand what students and researchers usually look for when trying to analyse their articles, thesis or dissertations. We have also scoured the web to recognise the features our application might need to become a one-stop solution.
* We have listed a few features that could contribute to our goal.
* The result of the system is to analyse articles, thesis, dissertations etc. efficiently.

**ARTICLE SUMMARISATION**

* One of the first features identified was Article Summarisation. Through our research we identified that the abundance in public content makes it harder for researchers to sort through relevant and irrelevant data.
* To even know if the article they are studying will help them in any way, they have to read it till the end to understand what’s discussed. This could take up so much time especially when you consider the fact that a document could be 20, 30 or probably even more than 100 pages long.
* We included this feature in our application by summarizing the articles using an Adaptive Text Summarization API.
* Firstly, we need to understand-

**5.2 What is an API?**

We need to understand that a software intermediary needs to exist for two applications to communicate with each other.

Imagine you are checking weather or using an App for instant messaging, so what you are doing is using an API (Application Programming Interface).

If you send a request to the provider, the API delivers it to the provider and returns with the response from the provider.

**-How does an API work?**

For example, imagine you go to a library for a book but you can’t just take the book and walk out with the book without reporting it to someone, you need the librarian to issue the book to you. Therefore, the librarian - or API - takes your request and issues the book to you.

APIs also provide a layer of security as they are standardized, governed and monitored for performance.

APIs make complex processes reusable with just a little bit of code. This makes the process of app development faster.

**-How to send and retrieve data using an API?**

The architecture of an API consists of three parts-

User, who makes a request, Client, the computer which sends the request to the server and server, which responds to the request made by the client.

In order for the server to send a response and hold data it has to be built by someone first. Once the server starts running, the programmer can publish its documentation and endpoints so that other developers can understand the structure of data on the server.

Then the developer can send a query to search the data in the database and transform the data into a different desired format.

* Process Flow-

1. The User interacts with the UI:

On our application the user can choose from multiple options available to analyse the article. On choosing the option, the particular page related to it will open displaying information about the chosen option and what can be done next.

1. Choose next option:

Next, the user is provided with the following option for performing Article Summarization.

* Paste the text in the TextBox

Here, the user will get to paste the article he wants to analyse in the textbox provided on the page for summarization.

1. The provided text is analysed by the API
2. Finally, the summarized text in displayed on the UI

**5.3 Anaconda Navigator**

For this project we will be needing the Anaconda Navigator to launch our python programs and manage our conda code. You can use any IDE for this project, but it is highly recommended to use the Jupyter notebook for the project. This is because python is an interpreted language, so one can take its full advantage by running a few lines of code and see and understand what’s happening, step by step, instead of writing the whole script and then running it.

**5.4 Flask in Python**

To establish more control over the components in our application, we used flask for the development. Flask was used as it is basically a third-party Python library for developing Python web applications.It establishes more control over the components such as databases and lets you interact with them. It added more flexibility to our project and increased the speed of developing the application. If you don’t have it installed on your system, download and install it by running the following command in the command prompt.

**pip install flask:**

You can use any IDE for this project, but it is highly recommended to use the Jupyter notebook for the project. This is because python is an interpreted language, so one can take its full advantage by running a few lines of code and see and understand what’s happening.

* **Project Structure**

We started by creating a project folder named DraftCheck and included all our files in it.

The DraftCheck folder consisted the following files for developing the Article Summarization page:

1. Templates folder to include our HTML pages
2. The Flask Python Scripting file named App.py
3. And the testing file containing static images

**WORD COUNTER**

* Having to stick to a word limit is not unheard of when writing articles. We are expected to write our entire article within the word limit given to us.
* When a limit is set, writers are put under an obligation to keep a check at the number of words or characters they use.
* For this purpose we added another feature to our application that lets users get a word count of their articles.
* The users can provide the article directly in the textbox on the page.
* **Project Structure**

Using the DraftCheck folder that was created earlier to save all our files, we saved the files related to the WordCount feature as well. We saved the HTML pages and the testing files in the same folder.

**KEYWORD EXTRACTION**

* Reading twenty, thirty or even a hundred pages to just understand what was discussed in an article takes up so much time.
* Researchers have an abundance of information to go through everyday, it gets harder for them especially when they have deadlines to meet.
* We imagined a system that would help us build an application that would help our users sort through hundreds and thousands of texts to get an understanding of the important or relevant keywords and expressions used in the article provided by them.
* Our application takes the input in the following way:

1. Text in the Text Box provided on the page.

* It definitely becomes a game-changer when we automate the process of extracting the most relevant keywords or expressions from the input provided by the user.

**Spell Checking**

* How often than not do we accidentally spell a word incorrectly? And the incorrect spellings could get lost in the abundance of the information we write.
* To recheck each and every spelling and make the required corrections could take up a lot of time and for this reason we have included another option on our application that lets our users paste their articles in the textbox provided.
* On clicking the spell check button, the user will be provided with their article that has been checked for incorrect spellings and been corrected.

**6. IMPLEMENTATION**

To run the application firstly, all of our files were stored in the same folder named Article. The templates folder contains all the html files. And the Article Summarizer folder also contains the python file named app.py that’ll run our application.

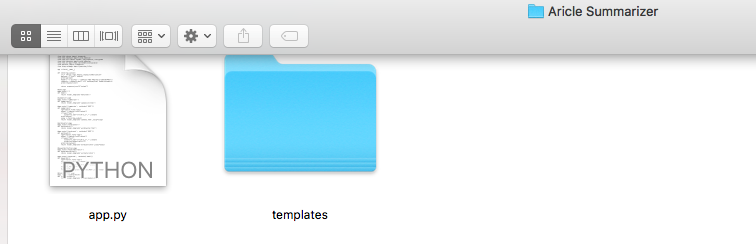


Figure 6.1.1

Within the templates folder all the html files are stored. Each html file takes the application to a web page designed for its purpose. For example, the summarizer file takes you to the webpage that allows the user to provide the article in the form of text in the textbox.

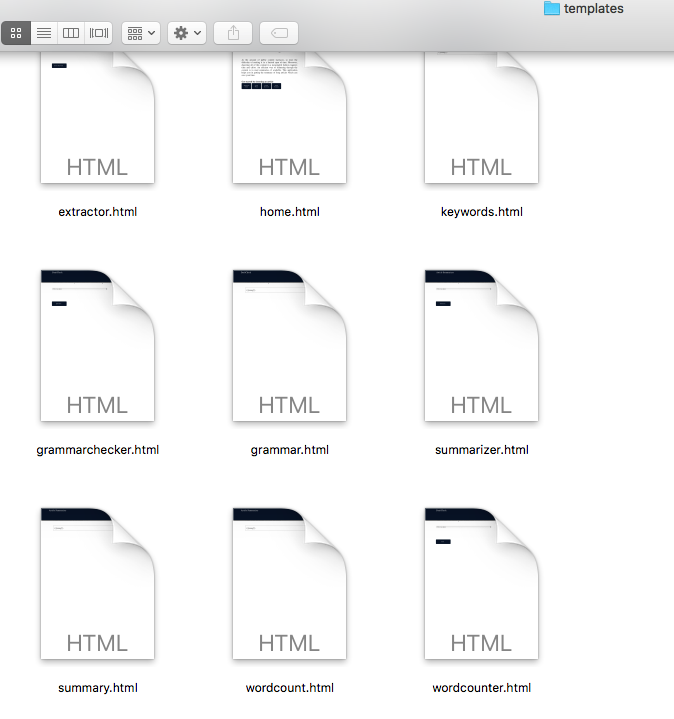


Figure 6.1.2

**6.1 app.py**



Figure 6.1.3 app.py

The app.py file is our application’s route. We’ve imported flask here and on installing flask, we have also installed the flask command-line script.

On running flask, we prompt the flask package in the virtual environment to run an HTTP server using the app object.

****

Figure 6.1.4

Under the summarize(), we have used the re.sub() function to replace the occurrences of the substring with another substring to replace.

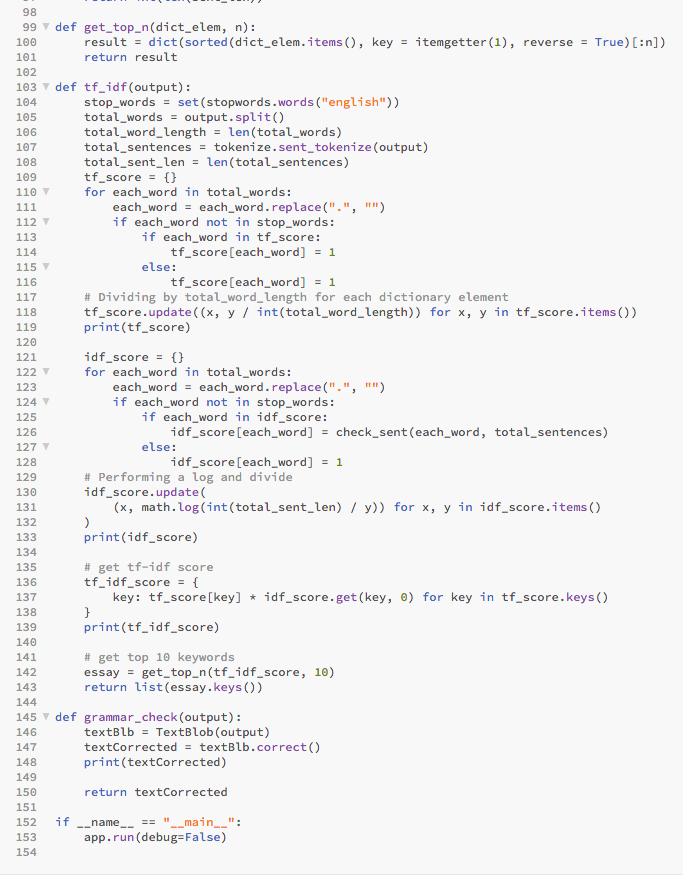
check(typ,output) was used to check if the particular instance is present in the string or not.****

Figure 6.1.5

Under the wordcount() function, we used the split() function to split the string into a list using the separator. len() was used to print the number of words in the lists to give the output, i,e, the word count.

Under the keywords() function, we used the TF-IDF technique for keyword extraction.

The TF-IDF(Term Frequency-Inverse Document frequency) uses the concept where TF calculates the frequency of a term in a text where it is measured as the number of times a term,t, appears in a text / the total number of words in the document.

The Inverse Document Frequency calculates how important the word is in a document by measuring it as log(the total number of sentences/ number of sentences with the term t).

TF-IDF- The importance of the words is measured by this score and it is measured as TF\*IDF.

We use the TextBlob package for spell check. The package is used for processing textual data and it provides an API for tasks like Natural Language Processing .

**6.2 Summarizer.html**

****

Figure 6.1.6 summarizer.html

It provides a textbox for the user to paste his article in and summarises the article provided on clicking submit.

**6.3 Wordcount.html**

****

Figure 6.1.7 wordcounter

It provides a textbox for the user to paste his article in and counts the words in the article provided on clicking submit.

**6.4 Keywords.html**

****

Figure 6.1.7 Keywords

It provides a textbox for the user to paste his article in and extracts keywords in the article provided on clicking submit.

**6.5 grammarchecker.html**



Figure 6.1.8 grammarchecker

It provides a textbox for the user to paste his article in and checks and corrects the spellings in the article provided on clicking Spellcheck...

**6.6 Home.html**

The home.html page is used as the default page for the application. It briefs the user about the options available on our application and links to let the user access those options. Using href, we provided the urls of the other html pages mentioned above.

****

Figure 6.1.9 home

**7. TESTING**

The purpose of testing is to discover the errors. Testing is a process of discovering every possible fault or weakness in a work product. This is a process of validating the software to make sure that the system meets its requirements and user expectations and does not fail in an unacceptable manner.

Most of the time, the end user might not be equipped with technical knowledge and hence it is important that the developers think from the end user point of view as well.

**System Testing :**

System testing is important as it enables us to validate and verify both the application architecture and business requirements. Here the application or system will be tested in an environment similar to where it will be deployed. Thus it helps us to find out all possible ways in which the application can fail when used by the end user.

**Unit Testing :**

Unit testing is performed on the modules constructed during system design. Each module is tested by providing inputs that are specific to the modules. These modules are also assembled into larger units and tested in the process of unit testing. The main aim of the unit testing is to make sure that the information flow into and out of the units is efficient. The outputs generated are checked for their accuracy.

**Integration Testing :**

The DraftCheck application consists of four main tasks - article summarization, keyword extraction,spell checker and word counter. Since it is a web application, it contains front end and back end. Therefore it is important to verify whether these tasks are performing efficiently and if there is any difficulty in the communication between different parts of the application. Here we test the software using two methods: white box testing and black box testing. Both techniques help in finding maximum errors with minimum effort.

**7.1 Test Objectives :**

* All the methods and algorithms used must function properly.
* Validate the output returned.
* Test how the system performs in unprecedented conditions.

**7.2 Test Results:**

The above criteria have passed successfully. Though we had encountered a few issues with our application we have rectified them to improve the results.

Further, a few other issues that could help us frame the future scope of our application have been listed below in the results and conclusion section.

**8. RESULTS**

We have tested our application to check if the output matches with what’s desired from the application.

The following images show the webpages that come up when a request is made and when the response has to be displayed.

The image below shows the homepage of the application. It provides the user with four options- Summarize article, count words, extract keywords and check grammar.

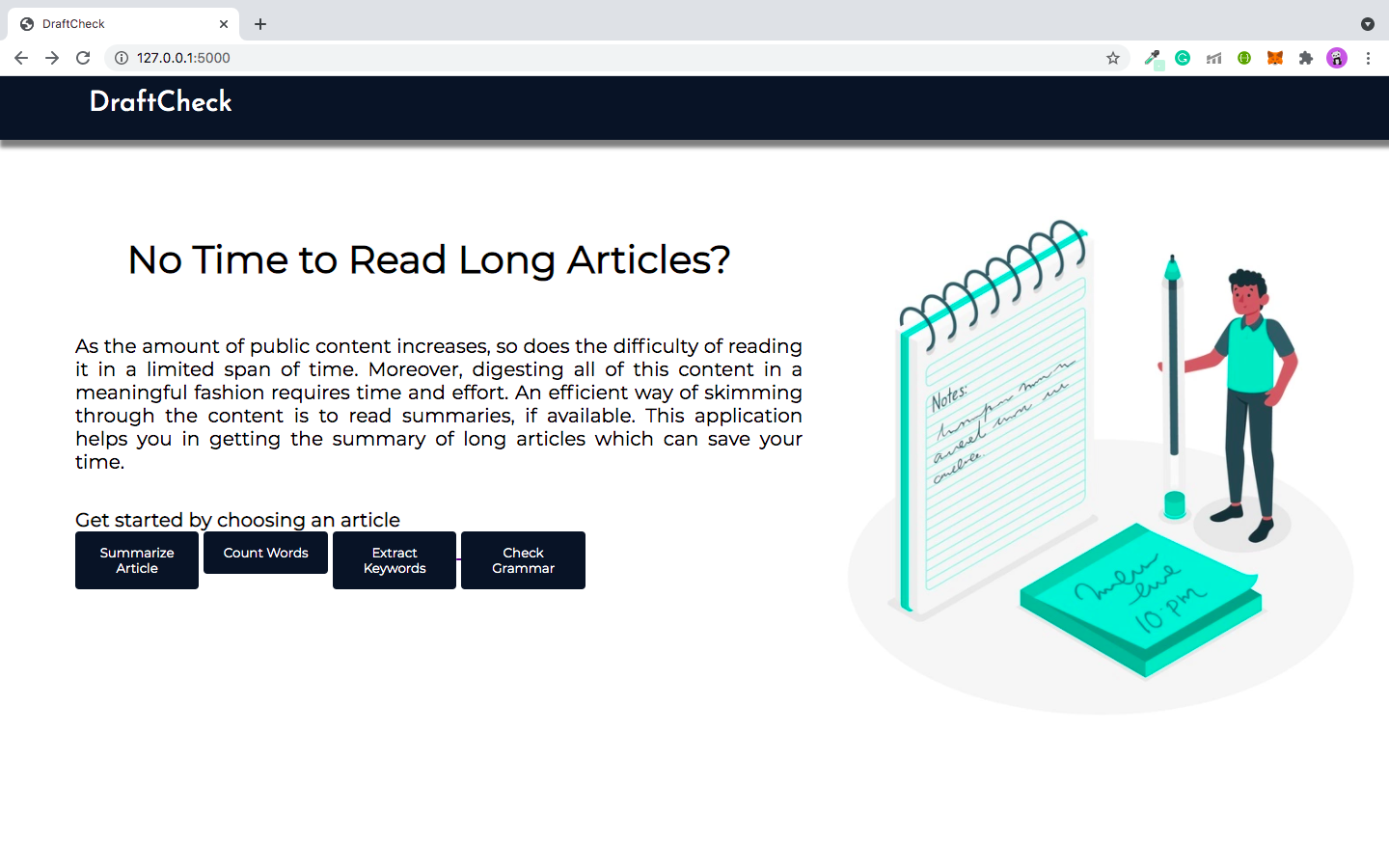
****

figure 7.2.1

The following image shows the webpage that provides the user with a textbox to paste his article.

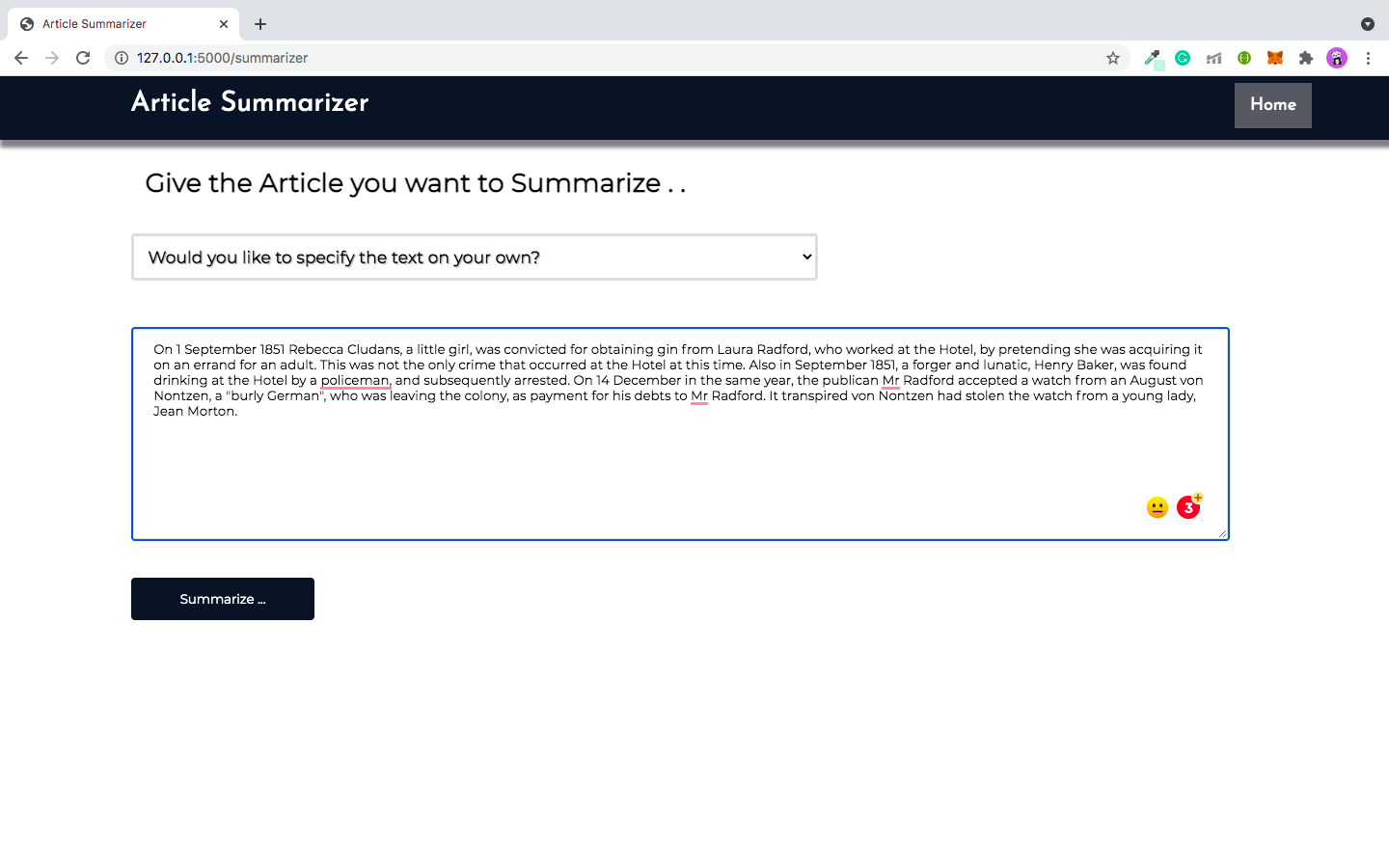
****

figure 7.2.2

The image below shows the summary of the article provided by the user in the textbox.

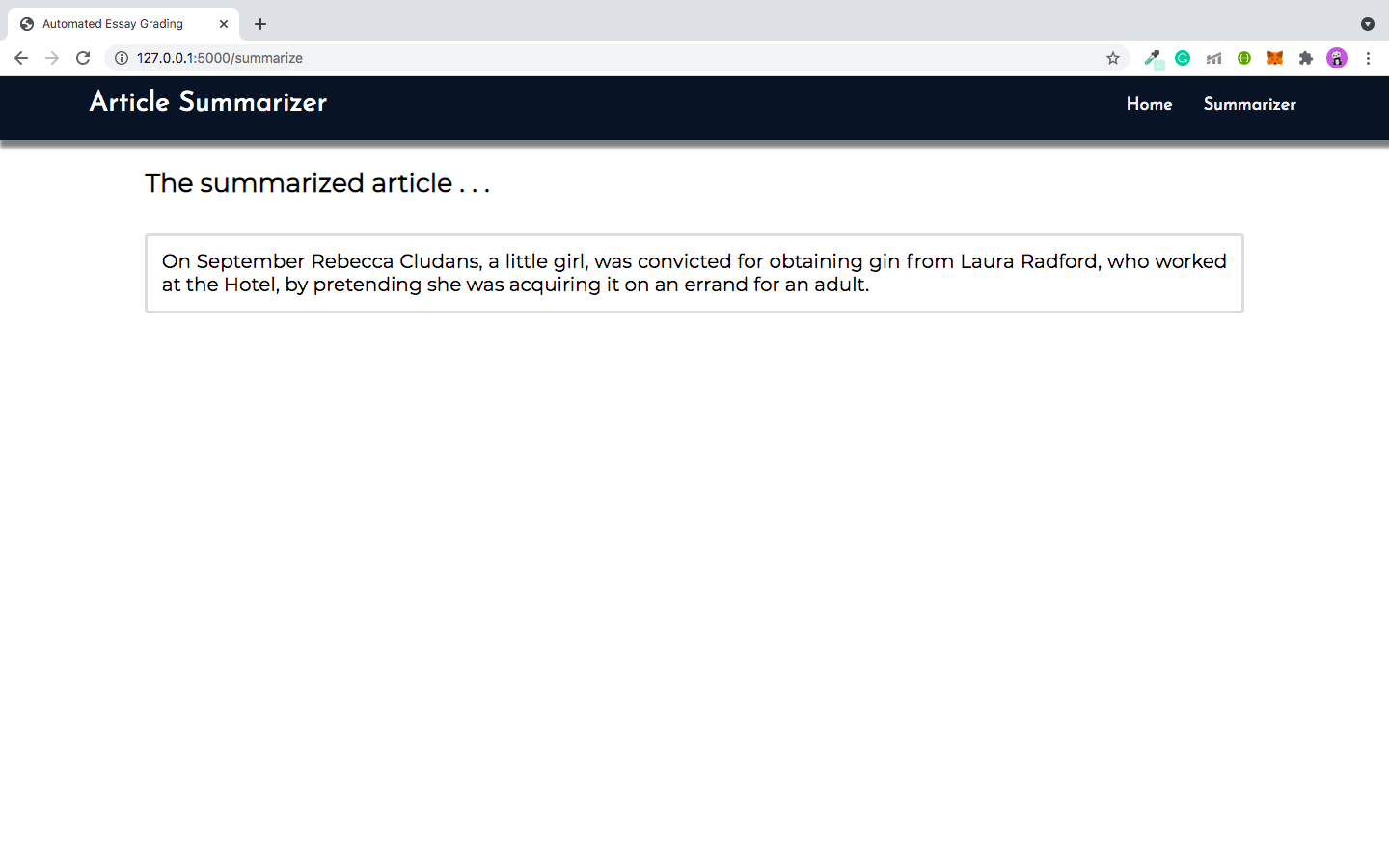
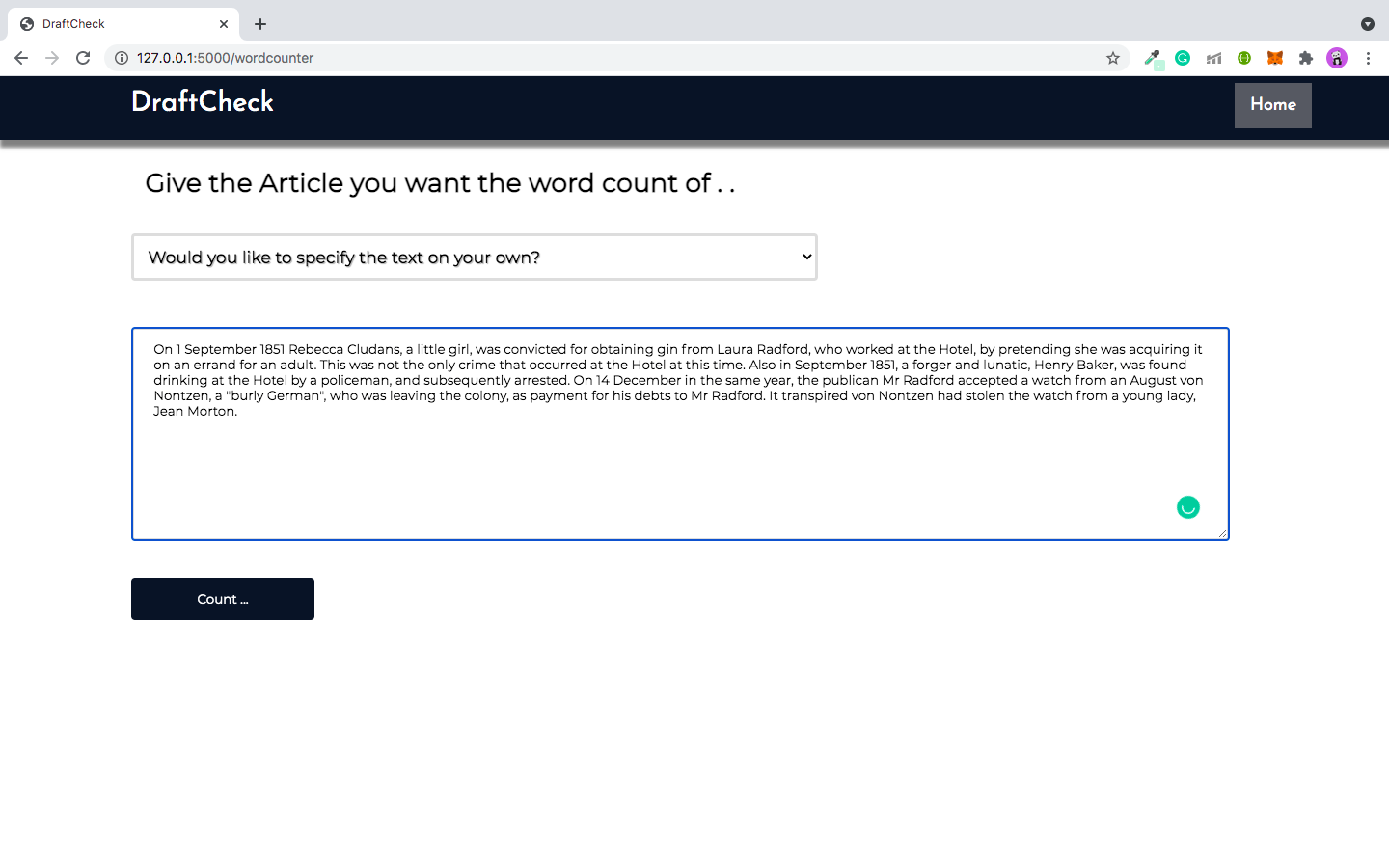
****

figure 7.2.3

The following image shows the webpage that provides the user with a textbox to paste his article.

Here, the user can paste the article for which he would like to get the word count of.

****figure 7.2.4

The image below shows the word count of the article provided by the user in the textbox.

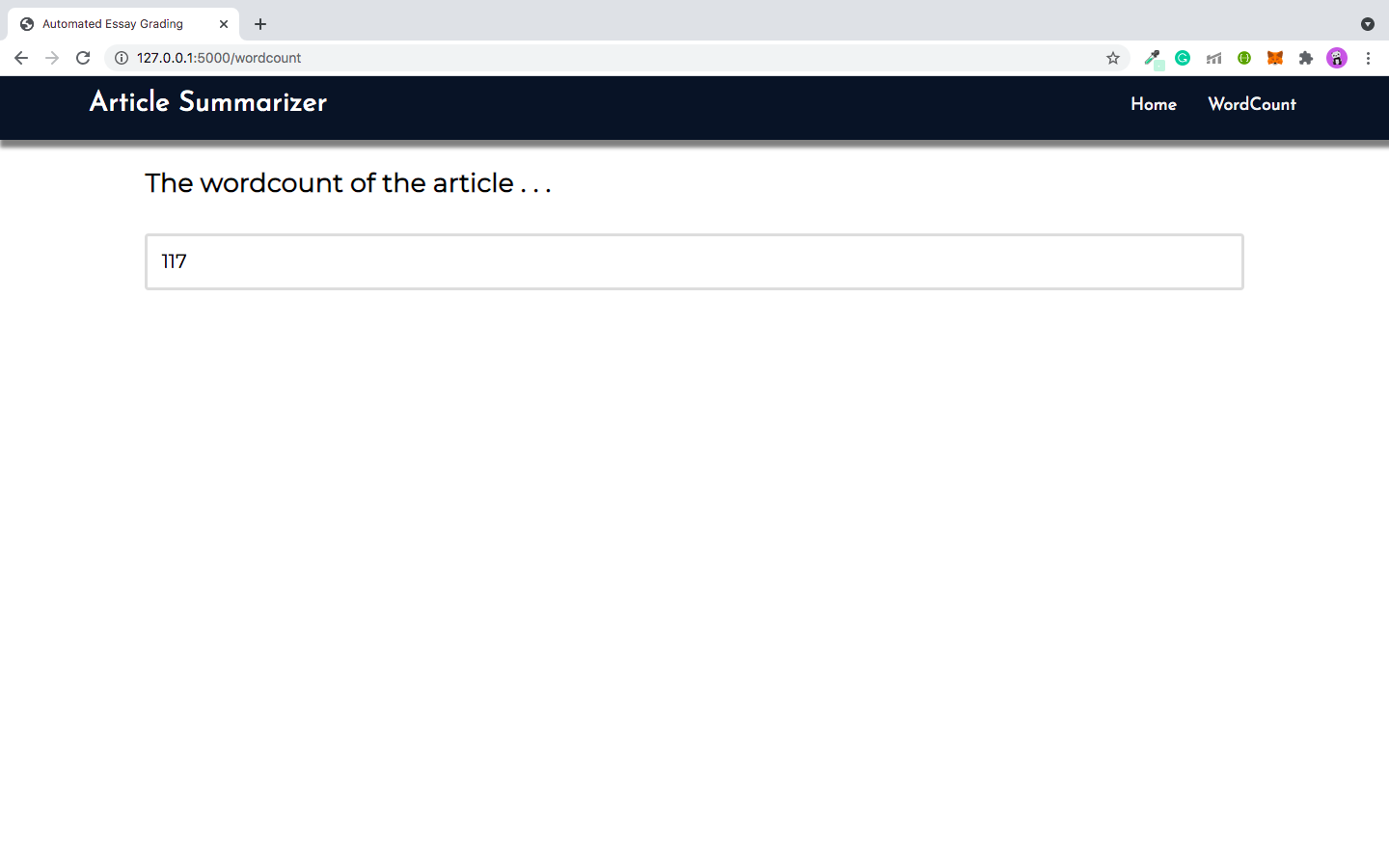
****

figure 7.2.5

The following image shows the webpage that provides the user with a textbox to paste his article.

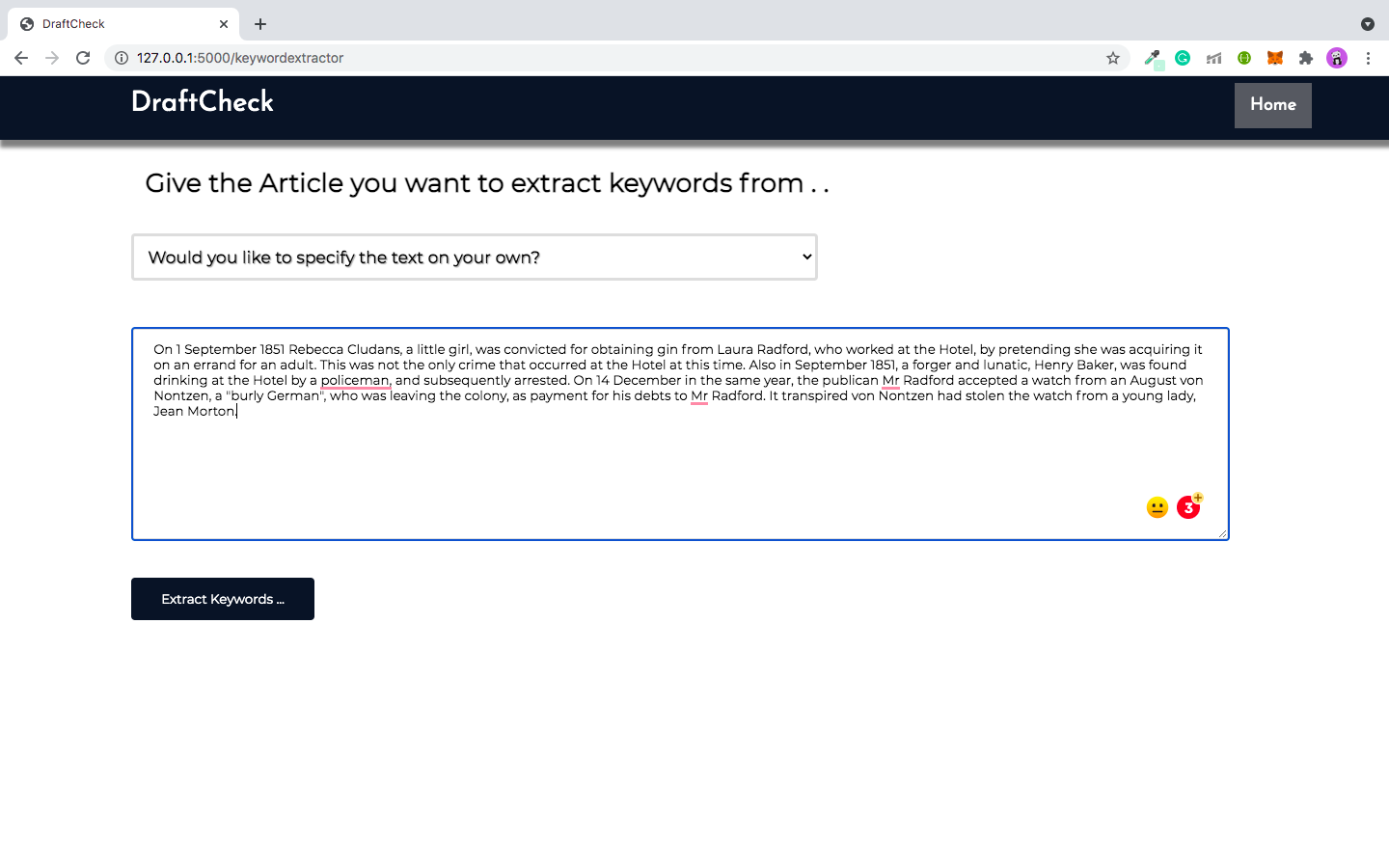
****

figure 7.2.6

The image below shows the keywords of the article provided by the user in the textbox.

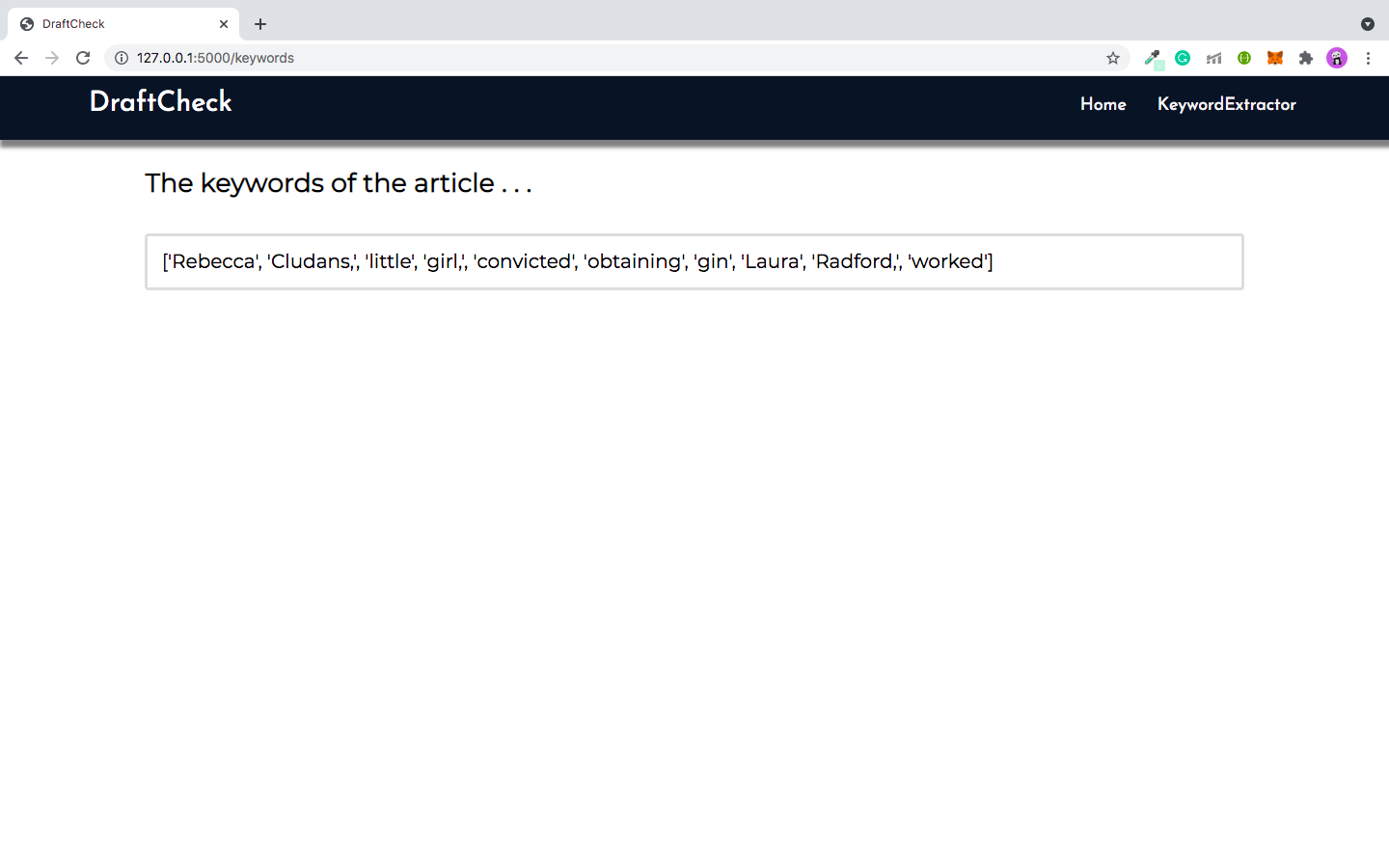
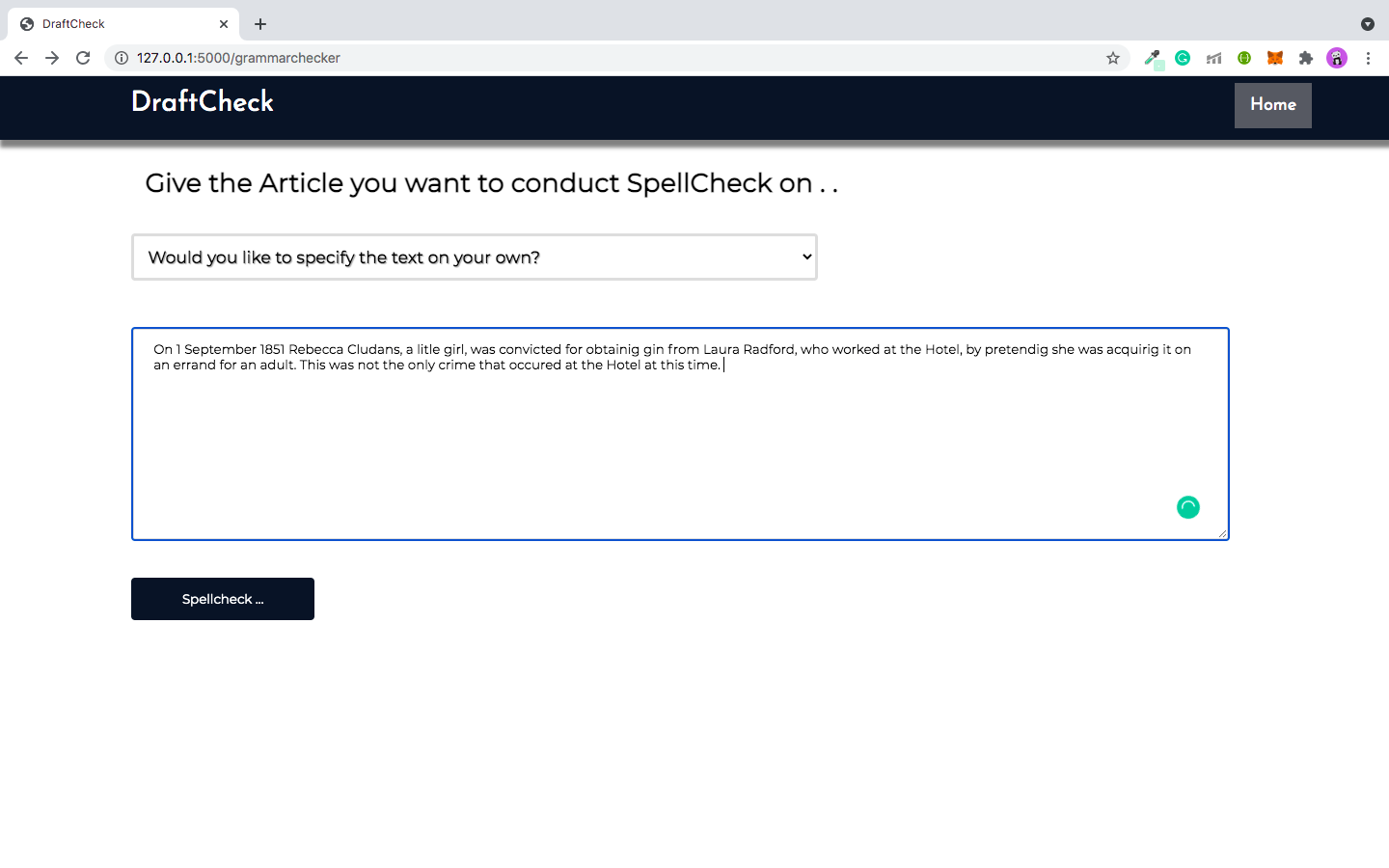
****

figure 7.2.7

The following image shows the webpage that provides the user with a textbox to paste his article.

****figure 7.2.8

The image below is the output after spell checking the article provided by the user in the textbox.

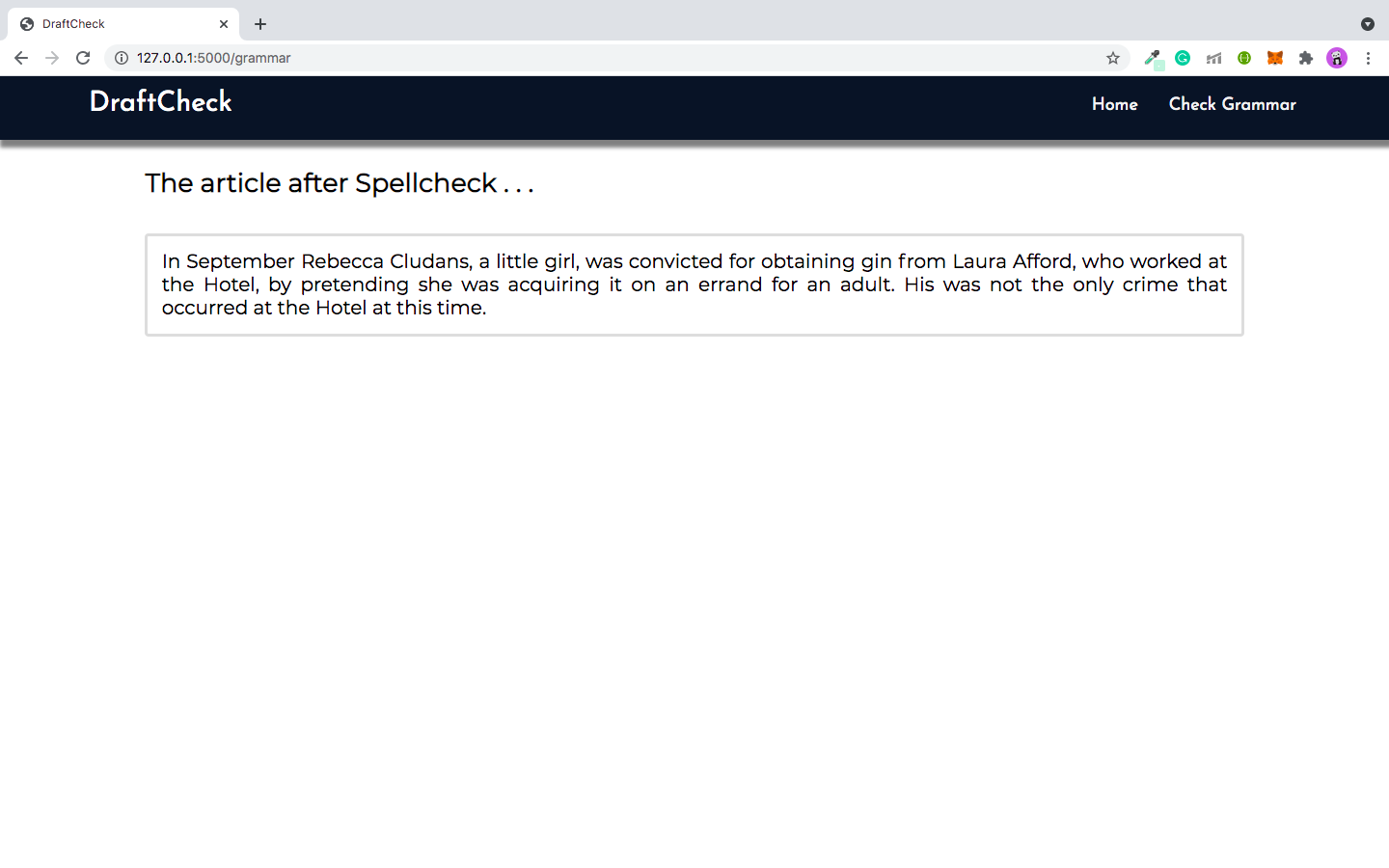
****

figure 7.2.9

**9. CONCLUSION**

Our application, DraftCheck has functioned the way we initially expected it to. The options, summarize an article article, word count, keyword extraction and spell check, that we have provided on our application have worked well with the articles that were pasted in the textbox provided.

Though we had initially provided multiple ways for the user to provide the article to the application, i.e. Either through an URL or by directly pasting the article in the textbox, the latter had given better output due to which we had to eliminate the option of providing the article through an URL and work on it further as it would be a great addition to the application.

We have also taken into account that our application is currently catering only to articles written in english and is making changes to the article based on US(en) but our users might want to provide articles in various languages, and we added it to the future scope of the application.

The word counter worked well with the data provided. The keyword extractor is currently extracting keywords based on the TF-IDF score and has given us the results only based on it but we believe we can further research methods that would give better results.

Therefore, we believe that our application has a great scope in the real-time world and provides a platform for further improvement and many more additions that would benefit researchers, students, academicians and many more.

**10. FUTURE WORK**

The future development of this application is endless.

Integrating other article analysing tasks such as plagiarism checking, grammar checking , spell checking etc helps the user to perform various tasks accurately at one place. This makes the life easier for the user as he need not search for sites to perform each one of these tasks. Providing an option to perform all the operations at once and returning results of each operation separately will save time for the user. Including various other operations can also make DraftCheck a widely used application for article analysis.

Making DraftCheck available as a mobile application can make it more user friendly. This saves user time. Once he downloads the application he can use it any time unlike web applications where he should load the link/url to the web app every time he wishes to perform a task.

Another possible development of DraftCheck is making it available in multiple languages. Although the application currently performs operations only on articles and documents written in english language, developing it to handle articles written in multiple languages can be an added advantage. This increases the complexity of the application. But if implemented the number of users of the application will increase.

**11. BIBLIOGRAPHY**

1. <https://stackoverflow.com/questions/1193408/how-to-develop-a-plagiarism-detector>
2. <https://smmry.com/>
3. <http://freesummarizer.com/>
4. <https://wordcounter.net/>
5. <https://www.docsvault.com/features/document-scanning-digitization/#:~:text=In%20document%20management%2C%20Document%20Scanning,repository%20for%20easy%20retrieval%20later>.
6. <https://en.wikipedia.org/wiki/Optical_character_recognition>
7. <https://www.analyticsvidhya.com/blog/2020/11/words-that-matter-a-simple-guide-to-keyword-extraction-in-python/>
8. <https://opensource.com/article/18/4/flask>
9. <https://www.w3resource.com/python-exercises/string/python-data-type-string-exercise-12.php>
10. <https://towardsdatascience.com/understand-text-summarization-and-create-your-own-summarizer-in-python-b26a9f09fc70>
11. <https://nanonets.com/document-digitization>
12. <https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/>
13. <https://flask.palletsprojects.com/en/2.0.x/>
14. <https://www.educative.io/blog/web-development-in-python>
15. <https://textblob.readthedocs.io/en/dev/>