

**Tribhuvan University**

Faculty of Humanities and Social Sciences

**“Online Text Summarization tool”**

Submitted to

Department of Computer Application

**Asian College of Higher Studies**

In partial fulfillment of the requirements for the Bachelors in Computer Application

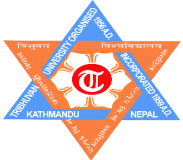
Submitted by

Gopal Dhakal

2023/12/08

Under the Supervision of

**Mr. Bidur Sapkota**



Tribhuvan University

Faculty of Humanities and Social Sciences

Asian College of Higher Studies

**Supervisor’s Recommendation**

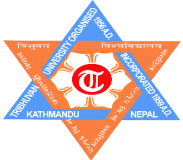
I hereby recommend that this project prepared under my supervision by Gopal Dhakal entitled “Online Text Summarization Tool” in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

………………………..

**Mr. Bidur Sapkota**

**Lecturer**

Asian College of Higher Studies



Tribhuvan University

Faculty of Humanities and Social Sciences Asian College of Higher Studies

**LETTER OF APPROVAL**

This is to certify that this project prepared by Gopal Dhakal entitled “**Online Text Summarization Tool**” in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| **Signature of Supervisor**  ……………………………  **Mr.Bidur Sapkota**  Asian College of Higher Studies | **Signature of Coordinator**  -------------------------------------------  **Mr. Pranaya Nakarmi**  Asian College of Higher Studies |
| **Signature of Internal Examiner**  --------------------------------------------  **Mr. Janak K Lal**  Internal Examiner | **Signature of External Examiner**  ---------------------------------------------  **Mr. Sushant Poudel**  External Examiner |

# ABSTRACT

In an era of information overload, effective text summarization plays a crucial role in providing users with concise and meaningful insights from voluminous content. The "Online Text Summarization Tool" is a web-based application designed to streamline this process. Leveraging natural language processing (NLP) algorithms, including TextRank and TF-IDF, the tool aims to condense lengthy text passages into coherent and condensed summaries. The project integrates a user-friendly interface, allowing users to input text through manual entry or file uploads. A selection of summarization algorithms provides flexibility in catering to diverse user preferences. The backend, developed using Flask, manages the summarization process and communicates seamlessly with the frontend built using React. Key features include algorithmic summarization, the ability to upload text files for analysis, and user-friendly options for copying or exporting summaries. The tool also supports multiple algorithms, enabling users to choose the summarization approach that best suits their needs. This report outlines the project's architecture, development process, and integration of summarization algorithms. It explores the challenges faced during implementation and provides insights into potential future enhancements. The "Online Text Summarization Tool" serves as an effective solution for individuals seeking to distill large volumes of text into concise and meaningful summaries, promoting efficient information consumption in an increasingly data-centric world.

# ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all those who contributed to the successful completion of the "Online Text Summarization Tool" project. This endeavor would not have been possible without the support, guidance, and encouragement from various individuals and resources. First and foremost, I extend my sincere appreciation to my project superviso for his invaluable guidance and mentorship throughout the development process. Their expertise and insights were instrumental in shaping the project and overcoming challenges. I would like to thank Asian College of Higher Studies for providing the necessary infrastructure and resources for the successful execution of the project. The learning environment and facilities have played a pivotal role in fostering creativity and innovation. I extend my gratitude to my peers and colleagues who offered their constructive feedback, suggestions, and encouragement, contributing significantly to the project's improvement. Collaborative efforts have been key to the project's success. Furthermore, I am grateful to the open-source community for the wealth of knowledge, tools, and frameworks that formed the foundation of this project. The collaborative spirit of the community has been a constant source of inspiration. Last but not least, I want to express my deepest appreciation to my family and friends for their unwavering support and understanding throughout the project's development. Their encouragement provided the motivation needed to overcome challenges and persevere. This project has been a rewarding experience, and the acknowledgment extends to all those who played a role, directly or indirectly, in making it a reality.

Thank you.

Gopal Dhakal

Asian College of Higher Studies

**TABLE OF CONTENTS**

ABSTRACT iii

ACKNOWLEDGEMENT iv

LIST OF ABBREVIATIONS vi

LIST OF FIGURES………………………………………………………… …………….viii

LIST OF TABLES……………………… ………………………………………………….ix

CHAPTER 1: INTRODUCTION 1

1.1. Introduction 1

1.2. Problem Statement 1

1.3. Objectives 1

1.4. Scope and Limitation 2

1.4.1. Scope 2

1.4.2. Limitation 3

1.5. Development Methodology 4

1.6. Report Organization 5

Chapter 2: Background study and Literature review 5

2.1. Background study 7

2.2. Literature Review 7

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN 8

3.1. System Analysis 8

3.1.1. Requirement Analysis 8

i. Functional Requirements 8

ii. Non Functional Requirements 9

3.1.2. Feasibility Analysis 10

i. Technical Analysis 10

ii. Operational Analysis 10

iii. Economic Analysis 11

iv. Schedule Analysis 11

3.1.3. Object Modelling using Class and Object Diagrams 12

3.1.4. Dynamic Modelling using State and Sequence Diagrams 14

3.1.5. Process Modelling using Activity Diagrams 15

3.2. System Design 16

3.2.2. Component Diagrams 16

3.2.3. Deployment Diagrams 17

3.3. Algorithm Details (if any) 18

CHAPTER 4: IMPLEMENTATION AND TESTING 21

4.1. Implementation 21

4.1.1. Tools Used 21

4.2. Testing 22

4.2.1 Test Cases for Unit Testing 22

4.2.2.1For Website 22

4.3.2 Test Cases for System Testing 23

CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS 26

5.1. Conclusion 26

5.2. Future Recommendations 26

**APPENDICES**

**REFERENCES**

# 

# LIST OF ABBREVIATIONS

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | GUI | Graphical User InterfaceOnline Text Summarization | | HTML5 | HyperText Markup Language 5 | | HTTPS | Hypertext Transfer Protocol Secure | | IDE | Integrated Development Environment | | JS | JavaScript | | NLP | Natural Language Processing | | OTS | Online Text Summarization | | PDF | Portable Document Format | | SPA | Single Page Application | | TF-IDF | Term Frequency-Inverse Document Frequency | | UI | User Interface | |  |  |
| **LIST OF FIGURES** |  |  |
| **Figure 1.1 Waterfall Model of SDLC 5**  **Figure 3.1 Use Case Diagram 8**  **Figure3.2 GanttChart………………………………………………………………………………..11**  **Figure 3.3 Class Diagram 12**  **Figure 3.4 Object Digram 13**  **Figure 3.5 Sqeuence Diagram 14**  **Figure 4.1 Activity Diagram 15**  **Figure 4. 2 Component Diagram 16**  **Figure 4.3 Deployment Diagram 17**  **Figure 5.1 loginpage 26**  **Figure 5.2 Homepage 27**  **Figure 5.3 Implementation of algorithm 28** |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF TABLES**

**[Table 4.3 Unit test case urls 24](#_Toc142591829)**

**[Table 4.4 Unit Test cases, Expected output and Actual Output 25](#_Toc142591830)**

**[Table 4.5 File Upload Button test case 26](#_Toc142591831)**

# CHAPTER 1: INTRODUCTION

## Introduction

In the fast-paced world of information, the ability to quickly distill meaningful insights from vast amounts of text is crucial. Our project, the "Online Text Summarization Tool," is designed to address this need by employing advanced natural language processing and machine learning algorithms. This tool is crafted to offer users a seamless experience in extracting key information from lengthy textual content, providing a valuable solution to combat information overload. With its intuitive interface and powerful summarization capabilities, our project aims to enhance user productivity, making it an indispensable tool for anyone dealing with substantial amounts of text across various domains. This document provides a comprehensive glimpse into the features, functionalities, and technologies that make our tool a reliable companion in navigating the vast landscape of digital information.

* 1. **Problem Statement**

The increasing volume of digital content across various domains has led to information overload, making it challenging for individuals to quickly extract relevant insights from extensive texts. Traditional methods of reading and comprehending large documents are time-consuming and often impractical. This poses a significant challenge for professionals, researchers, and students who need to process a vast amount of textual data regularly. Our project addresses this problem by developing an "Online Text Summarization Tool." The objective is to create a user-friendly platform that leverages advanced algorithms to automatically generate concise and informative summaries from lengthy texts. By doing so, the tool aims to provide a time-efficient solution for users to obtain key information swiftly, ultimately improving productivity in information consumption.

## 1.3. Objectives

## Automated Summarization: To develop algorithms for text summarization, including default summarization and additional algorithms such as TextRank and TF-IDF.

## User-Friendly Interface: To create an intuitive web interface that allows users to easily input text and choose summarization algorithms.

## File Upload Feature: To implement the functionality to upload text files for summarization, expanding the tool's usability beyond manual input.

## Export and Download: To enable users to download generated summaries in various formats, such as plain text, PDF, or Word document.

## User Authentication: To implement a user authentication system to enhance security and provide personalized features for registered users.

## Responsive Design: To ensure that the application is responsive and accessible across different devices to accommodate a diverse user base.

## Integration of Advanced Algorithms: To explore the integration of more advanced algorithms for summarization, such as BERT-based models, to enhance the tool's capabilities.

## Dark Mode: To integrate a dark mode option for users who prefer a different visual theme for the application.

## **1.4. Scope and Limitation**

### **1.4.1. Scope**

* Text Summarization: The primary scope of the project is to provide a reliable and efficient text summarization tool that can generate concise summaries from input text using various algorithms.
* User-Friendly Interface: The project aims to offer a user-friendly web interface accessible to users with varying levels of technical expertise, ensuring ease of use.
* Algorithm Flexibility: Users will have the flexibility to choose from multiple summarization algorithms, allowing them to tailor the tool to their specific needs.
* File Upload Capability: The inclusion of a file upload feature extends the tool's functionality, enabling users to upload text files for summarization.
* Downloadable Summaries: Users can download generated summaries in different formats, promoting convenience and the sharing of summarized content.
* User Authentication: The implementation of a user authentication system enhances the security of the tool, allowing for personalized features and user-specific interactions.

### **1.4.2. Limitation**

* Algorithm Limitations: The accuracy and effectiveness of summarization may vary based on the selected algorithm and the nature of the input text. Some algorithms may struggle with certain types of content.
* Dependency on Text Quality: The tool's performance is dependent on the quality and coherence of the input text. Poorly written or highly technical content may result in less accurate summaries.
* Processing Time: Summarizing lengthy documents may take a longer time, impacting the tool's responsiveness, especially for complex algorithms.
* Algorithm Integration: While the project incorporates multiple summarization algorithms, it may not cover all available algorithms, leaving room for future enhancements.
* Device and Browser Compatibility: The tool's performance may vary across different devices and browsers, and achieving consistent performance on all platforms may pose challenges.
* Scalability: The tool's scalability may be limited by factors such as server capacity and processing capabilities, impacting its performance under heavy usage
* Language Dependency: The summarization algorithms may be language-dependent, and their effectiveness could be influenced by the language of the input text.
* Feedback Processing: While user feedback is encouraged, processing and incorporating every user suggestion may be challenging within the scope of the project.
* Dark Mode Implementation: The dark mode feature may have some limitations in terms of its visual appeal and compatibility with all elements of the interface.

## 1.5. Development Methodology

The Waterfall Model is a traditional linear sequential life cycle model in software development, where progress is seen as flowing steadily downwards (like a waterfall) through several phases. Each phase must be completed before the next one begins, and there is little to no overlap between phases.[1] The model is well-structured and easy to understand, making it one of the oldest and most straightforward methodologies. Here's an overview of the key phases in the Waterfall Model:

* Requirements Gathering and Analysis: The project begins with a thorough understanding of customer requirements. Analysts communicate with stakeholders to gather and document all necessary information. Detailed documentation is created, including functional and non-functional requirements.
* System Design: Based on the gathered requirements, the system architecture and design are developed. This includes creating detailed technical specifications for the system components.BSystem design also involves planning for hardware and software specifications.
* Implementation (Coding): The actual coding of the software takes place in this phase. Developers write the code based on the design specifications. This phase focuses on translating design documents into executable code.
* Testing: The system undergoes testing to ensure that it meets the specified requirements. Testing can include unit testing, integration testing, system testing, and acceptance testing. Bugs and defects are identified, reported, and addressed during this phase.
* Deployment (Installation):Once testing is successful and the system is deemed ready, it is deployed to the production environment. Users can now access and use the software. Deployment may include data migration, user training, and transitioning to the new system.
* Maintenance and Support: After deployment, the system enters the maintenance phase. This involves fixing any issues that arise, addressing user feedback, and providing ongoing support.Changes or enhancements may be made to the system as needed.

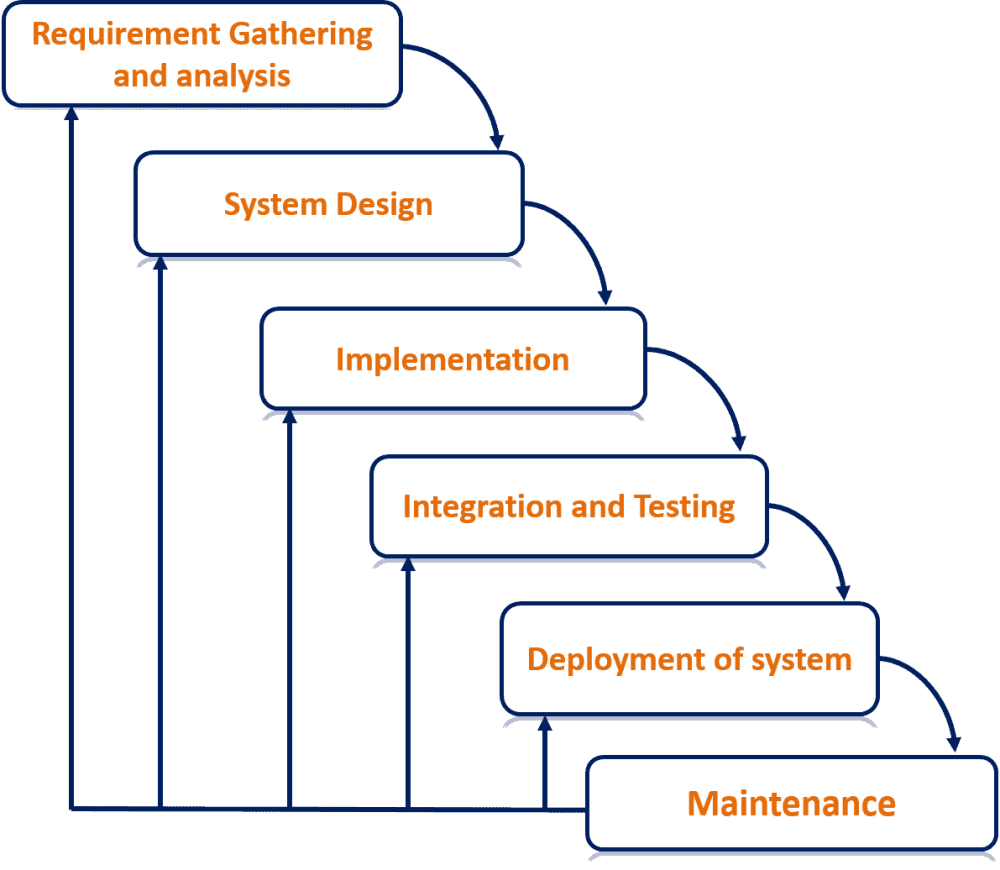


Figure 1.1 Waterfall Model of SDLC

## 1.6. Report Organization

**Chapter 1: Introduction**

This chapter describes Online Summarization Tool. And also explain the scope and objective of the project in addition it gives background information on why it is important and also gives a brief explanation of the methodology employed.

# Chapter 2: Background study and Literature review

This chapter includes the review of different research papers and talks about the different types of algorithms used for online summarization of text And also explains about the different kinds of research done on the topic text summarization and the study of different summarization algorithm.

**Chapter 3: System Analysis and Design**

This chapter encompasses the workings of the project including feasibility analysis, functional and non-functional analysis along with the schema and architectural design and the data and process modeling diagrams.

**Chapter 4: Implementation and Testing**

This chapter discusses the system's implementation tools, methods, and experiments as well as the results. The system's implementation tools, database, and programming languages are all fully explained in this document.

**Chapter 5: Conclusion and Future Recommendations**

An examination of the project's findings is presented in this chapter, along with a conclusion. To draw conclusions, the project's findings are examined and condensed. It also lists the project's successes in summary form and makes recommendations for further project-related work.

**CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW**

## 2.1. Background study

The project involves the development of an Online Text Summarization Tool, which aims to provide users with the ability to generate concise and coherent summaries of given text documents or input. Text summarization is a critical component in natural language processing, aiding in information retrieval and comprehension. The background study delves into the existing landscape of text summarization techniques, tools, and their applications.

## 2.2. Literature Review

The literature review underscores the significance of the Online Text Summarization Tool in addressing the need for efficient information processing. As information overload becomes a prevalent issue, tools that can distill and convey key information concisely are increasingly valuable. The project aims to contribute to this area by implementing and enhancing existing text summarization techniques.

* Text Summarization Techniques: The literature review explores various text summarization techniques, such as extractive and abstractive summarization. Extractive methods involve selecting and combining existing sentences, while abstractive methods involve generating new sentences to capture the essence of the text.
* Algorithms and Models: Reviewing state-of-the-art algorithms and models used in text summarization, including machine learning-based approaches like BERT, TextRank, and TF-IDF. Understanding the strengths and limitations of each algorithm is crucial for effective implementation.
* Applications of Text Summarization: Investigating real-world applications of text summarization in diverse fields, such as journalism, legal document analysis, and academic research. Understanding how summarization tools contribute to these domains provides insights into the tool's potential impact.
* User Experience and Feedback: Exploring user experiences and feedback on existing text summarization tools. Analyzing user preferences, challenges faced, and expectations helps in designing a user-friendly and effective tool.
* Challenges and Future Directions: Identifying challenges in text summarization, such as handling domain-specific language, addressing ambiguity, and improving the coherence ogenerated summaries.

**CHAPTER 3: SYSTEM ANALYSIS AND DESIGN**

## 3.1. System Analysis

System Analysis is the process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. It is done to study a system and its parts to identify the objectives and challenges.

### **3.1.1. Requirement Analysis**

The system requirement analysis is carried out in order to efficiently gather information about the system and the required functionality of the system, which includes hardware and software requirements for the application's satisfactory functioning and output. The requirements are divided into functional and nonfunctional requirements.

#### **i. Functional Requirements**

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks.

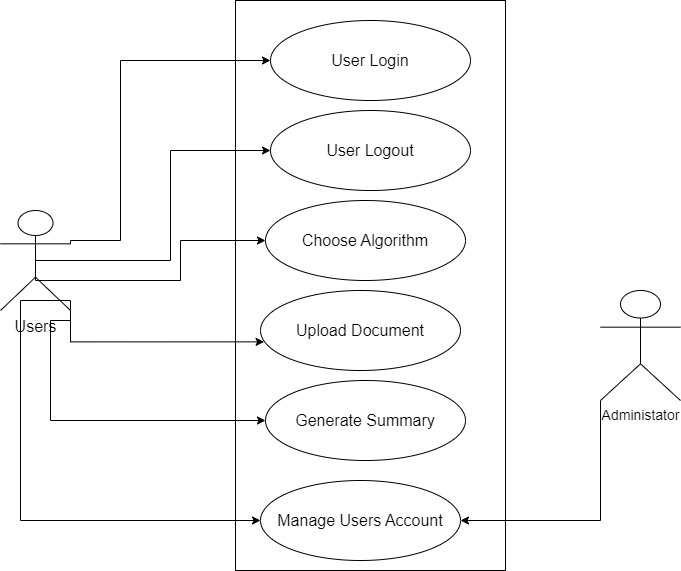


Figure 2. Use Case Diagram of Online Summarization Tool

In this system, the users and the admin are actors who interact with the system. The user has the option to sign up and log in. This use case diagram outlines the primary interactions between users and the summarization tool, capturing key functionalities such as text summarization, file upload, download options, and algorithm selection.

#### **ii. Non Functional Requirements**

Non-Functional Requirements determine criteria that can be used to judge the operation of a system, rather than specific behaviors. Non-Functional Requirements for the summarization tool project:

* Performance: The system should be capable of summarizing a moderate-length text (up to 1000 words) within a reasonable time frame (e.g., a few seconds). File upload and summarization should have acceptable response times even for large text files (up to 1 MB).
* Scalability: The system should handle multiple concurrent users without a significant decrease in performance. It should accommodate an increasing number of text summarization requests without a significant impact on response times.
* Reliability: The summarization tool should be available and reliable for use 99% of the time. It should have mechanisms to recover from failures gracefully and resume normal operation.
* Usability: The user interface should be intuitive and user-friendly, ensuring that users can easily interact with the tool without confusion. The summarization results should be presented in a clear and understandable format.
* Security: User data, especially input texts and generated summaries, should be handled securely and protected from unauthorized access. The system should be protected against common security threats, such as SQL injection and cross-site scripting.
* Compatibility:The tool should be compatible with commonly used web browsers (e.g., Chrome, Firefox, Safari). It should support multiple operating systems and devices to ensure broad accessibility.
* Maintainability: The system should be designed in a modular and maintainable way, allowing for future updates, improvements, and bug fixes.Code documentation and comments should be provided for ease of maintenance.
* Legal and Ethical Compliance: The tool should comply with relevant data protection and privacy laws. It should adhere to ethical guidelines regarding the use and handling of user data.

These non-functional requirements address various aspects such as system performance, reliability, usability, security, compatibility, maintainability, and legal/ethical considerations, ensuring a well-rounded and robust summarization tool.

### **3.1.2. Feasibility Analysis**

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There are different types of feasibility which includes:

#### **Technical Analysis**

Objective: Evaluate the technical viability of the project.

Considerations:

* Availability of required technologies and tools for development.
* Compatibility with web browsers and devices.
* Feasibility of integrating natural language processing libraries.

Outcome: The technical analysis shows that the project is feasible with readily available technologies and tools.

#### **ii**. **Operational Analysis**

Objective: Assess the operational aspects of the system.

Considerations:

* User-friendliness of the interface for both input and output.
* Scalability to handle multiple users simultaneously.
* User training and support requirements.

Outcome: Operational analysis indicates that the tool is user-friendly and scalable, with minimal training needs.

**iii. Economic Analysis**

Objective: Examine the financial feasibility of the project.

Considerations:

* Development costs, including software and hardware.
* Maintenance and support costs.
* Potential revenue generation or cost savings.

Outcome: Economic analysis suggests that the benefits of the tool, such as improved efficiency in information processing, outweigh the development and maintenance costs.

#### **iv. Schedule Analysis**

We have a proper schedule and enough time to complete this project in time.

Therefore, this project is schedule feasible.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity | Week (1-2) | Week (3-4) | Week (5-10) | Week (11-12) | Week (13-15) |
| Requirements Specification |  |  |  |  |  |
| Design |  | **X** |  |  |  |
| Development |  |  | **X** |  |  |
| Testing |  |  |  | **X** |  |
| Maintenance |  |  |  | **X** | **X** |

**Figure 3. 2: Gantt Chart**

### **3.1.3. Class and Object Diagrams**

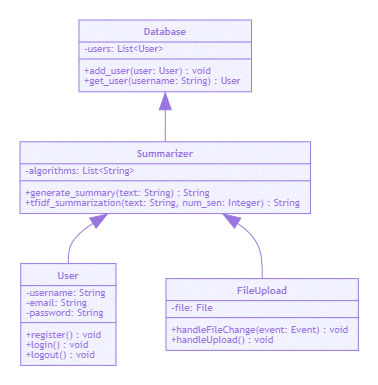
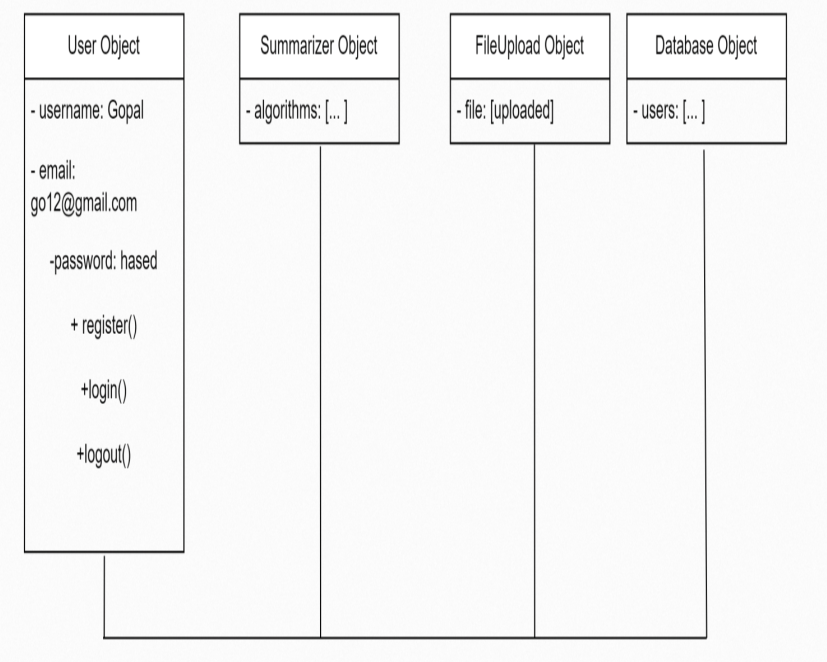


Figure 3.3 Class Diagram of a System

The uses relationship indicates that instances of the User class use the services of the Summarizer class. Similarly, instances of the FileUpload class use the services of the Summarizer class. The Summarizer class uses the services of the Database class to store and retrieve user information. This represents a simplified relationship.



**Figure 3.4 Object and Diagram of System**

The User Object interacts with the Summarizer Object to perform text summarization using selected algorithms. The User Object can also interact with the FileUpload Object to upload a file for summarization. The Database Object stores information about users, including their usernames, emails, and hashed passwords. The Summarizer Object and FileUpload Object may interact with the Database Object to retrieve or store relevant information. These relationships represent the interactions between different objects in the system.

### **3.1.4. Dynamic Modelling using State and Sequence Diagrams**

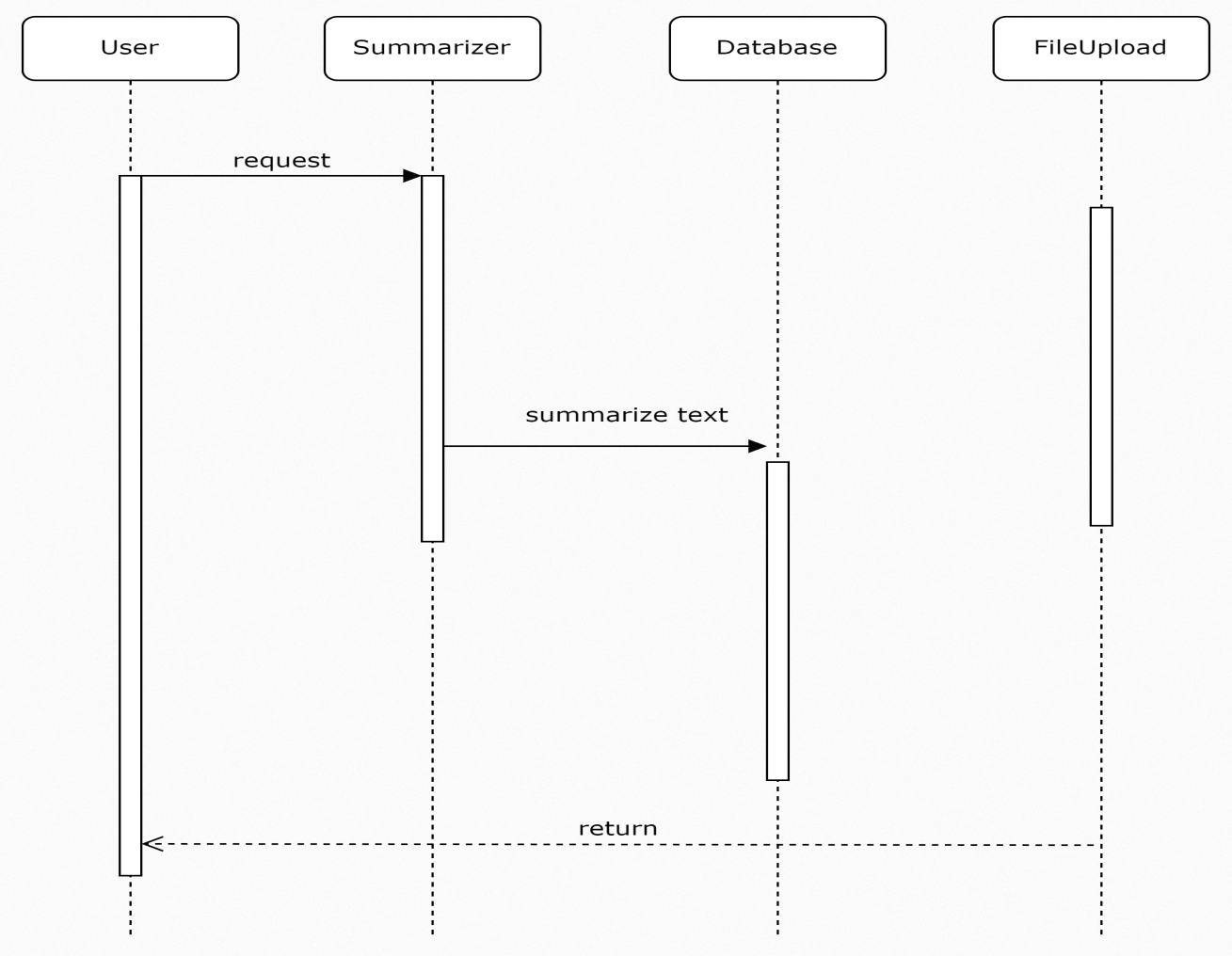


Figure 3.5 Sequence Diagram Of a System

This sequence diagram illustrates the flow of interactions between different components in your summarization tool. Here's a brief explanation:

* User: Initiates a request to the system by providing input text or uploading a file.
* Summarizer: Receives the request and performs text summarization using a specified algorithm (e.g., default, TF-IDF, TextRank).
* FileUpload: Receives the uploaded file, processes it, and communicates with the summarizer to generate a summary.
* Database: Stores or retrieves data as needed. In this case, it could store user information or other relevant data.

The interactions demonstrate how a user's request for summarization is processed by the Summarizer, whether through direct input or file upload. The summarization result is then potentially stored in a database or sent back to the user.

### **3.1.5. Process Modelling using Activity Diagrams**

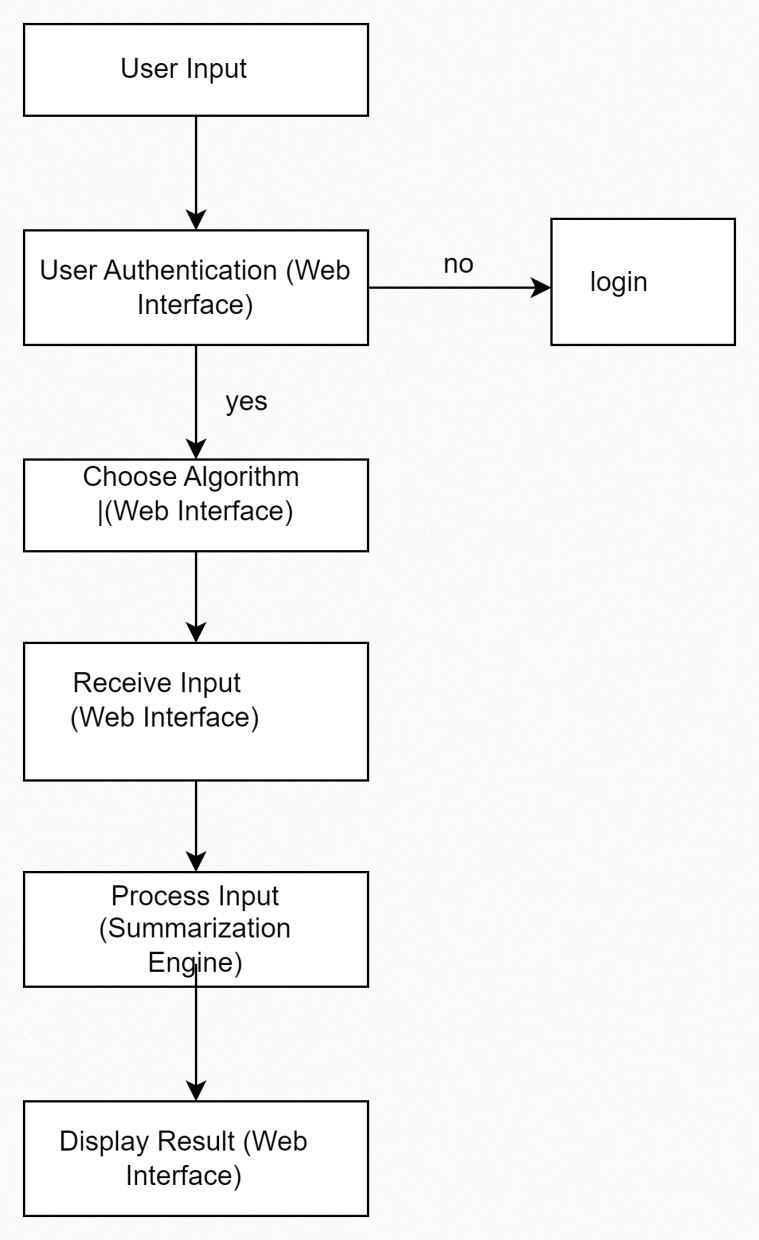
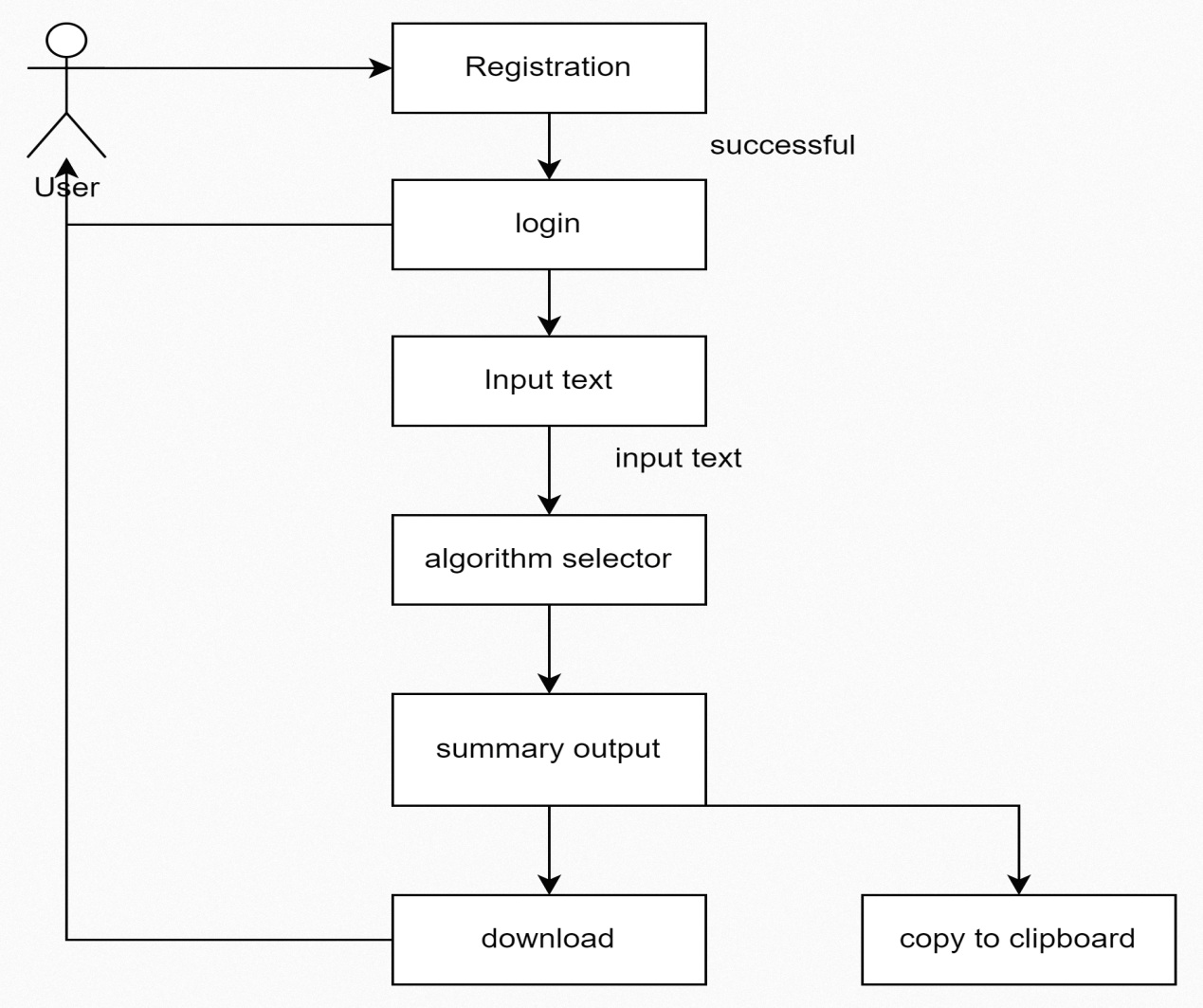


Figure 4.1 Online text Summarization tool Activity Diagram

The activity diagram illustrates the sequential steps involved in the user interaction within the summarization system. It begins with the user input, where individuals provide text for summarization. The process incorporates user authentication through the web interface to ensure access only for authenticated users. Subsequently, users have the option to choose the summarization algorithm via the web interface, offering flexibility in selecting the method that best suits their needs. The system then receives the user input and proceeds to the summarization engine, where the text processing occurs based on the chosen algorithm. Finally, the results are displayed back to the user through the web interface. This comprehensive representation outlines the sequential flow of activities, emphasizing key points such as user authentication and algorithm selection.

## 3.2. System Design

### **3.2.2. Component Diagrams**



**Figure 4.2 Component Diagram**

Creating Component Diagrams involves identifying the major components of your system and illustrating how they interact. In the context of your summarization tool project, above diagram gives the general outline of components.

### **3.2.3. Deployment Diagrams**

### deplo

Figure 4.3 Deployment Diagram of a System

Creating a Deployment Diagram involves illustrating the physical deployment of software components across different nodes (hardware or software entities). In your summarization tool project, nodes could represent servers, databases, or external services.

**3.3. Algorithm Details (if any)**

**i Description of Algorithm**

### **TF-IDF Summarization Algorithm:**

### TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents.

How it works:

In the context of summarization, TF-IDF is applied to sentences rather than words. It involves calculating the TF-IDF score for each sentence in the document.

The TF-IDF score considers the frequency of words in a sentence (Term Frequency) and inversely scales this frequency by the number of documents containing the word (Inverse Document Frequency). Sentences with higher TF-IDF scores are deemed more important and are included in the summary.

TF (term frequency) example

The TF (term frequency) of a word is the frequency of a word (i.e., number of times it appears) in a document. When you know TF, you’re able to see if you’re using a term too much or too little.

When a 100-word document contains the term “cat” 12 times, the TF for the word ‘cat’ is

TFcat = 12/100 i.e. 0.12

IDF (inverse document frequency) example

The IDF (inverse document frequency) of a word is the measure of how significant that term is in the whole corpus (a body of documents).

Let’s say the size of the corpus is 10,000,000 million documents. If we assume there are 0.3 million documents that contain the term “cat”, then the IDF (i.e. log {DF}) is given by the total number of documents (10,000,000) divided by the number of documents containing the term “cat” (300,000).

IDF (cat) = log (10,000,000/300,000) = 1.52

TF-IDF Calculation

Put the TF and IDF calculations together to get a TF IDF score.

∴ Wcat = (TF\*IDF) cat = 0.12 \* 1.52 = 0.182

A TF-IDF score of 0.182 is much closer to 0 than 1. This suggests that “cat” is a common term with less weight.

**II)TextRank Algorithm:**

TextRank is an extractive summarization algorithm, meaning it selects and extracts existing sentences from the input text to create a summary.

How it works:

The algorithm views sentences as nodes in a graph, and edges between nodes represent the similarity between sentences. Similarity can be measured using metrics like cosine similarity.

TextRank applies a graph-based ranking algorithm (similar to Google's PageRank) to identify the importance of each sentence based on its connections in the graph. Important sentences are considered representative of the document's key ideas and are included in the summary.

Example:

Document

Natural language processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and humans through natural language. [2]NLP aims to enable machines to understand, interpret, and generate human-like text. It involves various tasks such as language translation, sentiment analysis, and text summarization.

Text summarization is a crucial aspect of NLP. It allows the extraction of key information from a document, providing users with concise and meaningful summaries. Extractive summarization methods, like TextRank, identify important sentences that capture the essence of the document.[3]

TextRank Algorithm: TextRank views sentences as nodes in a graph, with edges representing the similarity between sentences. Similarity is often measured using cosine similarity based on the occurrence of words. The algorithm assigns a weight to each sentence based on its importance in the graph.

Steps:

1. Tokenize the document into sentences.

2. Create a similarity graph, where sentences are nodes and edges represent similarity.

3. Apply the PageRank algorithm to assign importance scores to sentences.

4. Select sentences with the highest importance scores for the summary.

Summary:

Natural language processing (NLP) is a vital part of artificial intelligence, focusing on human-computer interaction through language. Text summarization, an essential NLP task, uses methods like TextRank to extract key information from documents.

**III) Default Summarization Algorithm:**

The default summarization algorithm in your tool is likely an abstractive summarization model. Abstractive summarization involves generating new sentences that capture the essence of the input text.

How it works:

This type of algorithm typically utilizes advanced natural language processing (NLP) models, such as BERT (Bidirectional Encoder Representations from Transformers) or GPT (Generative Pre-trained Transformer). These models are trained on massive amounts of text data and learn to understand and generate human-like text. During summarization, the model analyzes the input text and generates a concise and coherent summary, attempting to capture the main ideas while producing human-readable sentences.

**CHAPTER 4: IMPLEMENTATION AND TESTING**

## 4.1. Implementation

During the research duration, various methodologies were used for data collection and system requirements. The dataset was collected from and Google. Different algorithms were studied and implemented and after analyzing them, the best performance and satisfying algorithm was chosen for deployment for our training session, Website development. The website was constructed using the Flask in the backend and React on the frontend .

### **4.1.1. Tools Used**

* **React.JS**

ReactJS is an open-source JavaScript library used to create user interfaces in a declarative and efficient way. It is a component-based front-end library responsible only for the view layer of a Model View Controller(MVC) architecture.[4] React is used to create modular user interfaces and promotes the development of reusable UI components that display dynamic data.

* **Plant UML**

Plant UML is a textual language used to create UML diagrams. It allows you to create diagrams such as class diagrams, sequence diagrams, activity diagrams, and more. Here's an example of how to use Plant UML to create a class diagram, State diagram, Component Diagram and so on.

* **Visual Studio Code**

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

* **Python**

Machine Learning is simply recognizing patterns in your data to be able to make improvements and intelligent decisions. Python is the most suitable programming language for its readability, non-complexity, and ability for fast prototyping with inbuilt libraries like Tensorflow, Pytorch, and Scikit-learn for machine learning. Hence python and its libraries numpy, panda, and Tensorflow are used to make auto encoder for Photo Pigmentation.

* **AUTH0**

Auth0 is a flexible, drop-in solution to add authentication and authorization services to your applications. Your team and organization can avoid the cost, time, and risk that come with building your own solution to authenticate and authorize users.[6]

* **FLASK**

Flask is a micro [web framework](https://en.wikipedia.org/wiki/Web_framework" \o "Web framework) written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)" \o "Python (programming language)). It is classified as a [microframework](https://en.wikipedia.org/wiki/Microframework" \o "Microframework) because it does not require particular tools or libraries. It has no [database](https://en.wikipedia.org/wiki/Database" \o "Database) abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.[7] Extensions exist for [object-relational mappers](https://en.wikipedia.org/wiki/Object%E2%80%93relational_mapping" \o "Object–relational mapping), form validation, upload handling, various open authentication technologies and several common framework related tools.

## 4.2. Testing

### **4.2.1 Test Cases for Unit Testing**

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. Unit testing is done to check whether the particular module works as expected or not, rework has been done for the module that does not meet the expectation.

#### **4.2.2.1For Website**

The Photo Pigmentation website began with a URL. This testing is done to find out whether the website responds the way it is made for. When we click on upload button, colorize button and any other action then it should respond to the valid page to the user .This kind of testing is called url testing and in our project we have done eight unit testing to test the different url for different module and which is run successfully.

Table 4.2 Unit test case urls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.  N: | Test Case | Expected  Output | Actual Output | Result |
| 1 | To check the url response of home page | 302 response | 302 response | Pass |
| 2 | To check the url response of About | 302 response | 302 response | Pass |
| 3 | To check the url response of login | 302 response | 302 response | Pass |
| 4 | To check the url response of logout | 302 response | 302 response | Pass |
| 5 | To check the url response of summary | 302 response | 302 response | Pass |

### **4.3.2 Test Cases for System Testing**

Table 4.3 Unit Test cases, Expected output and Actual Output

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.  N: | Test Cases | Expected output | Actual Output |  | Result |
| 1 | Default summarization algorithm | Running | Running |  | Pass |
| 2 | text\_rank summarization algorithm. | Running | Running |  | Pass |
| 3 | Tfidf summarization algorithm. | Running | Running |  | Pass |
| 4 | Download | Download summary.pdf | running |  | Pass |

System testing is the testing to ensure that by putting the software in different environments to check how well will the system perform in the different environment. System testing is done with full system implementation and environment. It falls under the class of black box testing.

###### 4.3.3.1For File upload:

In this testing we test the main functionality of the system that represents the overall system. Since our site is for the summarization of the long text we have used the different algorithms to get a summary.we have tested various component and we will check file upload component now.

Table 4.4 File UploadButton test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.  N: | Test Case | Expected Output | Actual Output | Result |
| 1 | File upload component’s rendering | to display field | to display field | Pass |
| 2 | File state on file input change | to show selected file | to show selected file | Pass |
| 3 | Button click for Upload and summarize | to display summary | Summary displayed | Pass |

# CHAPTER5: CONCLUSION AND FUTURE RECOMMENDATIONS

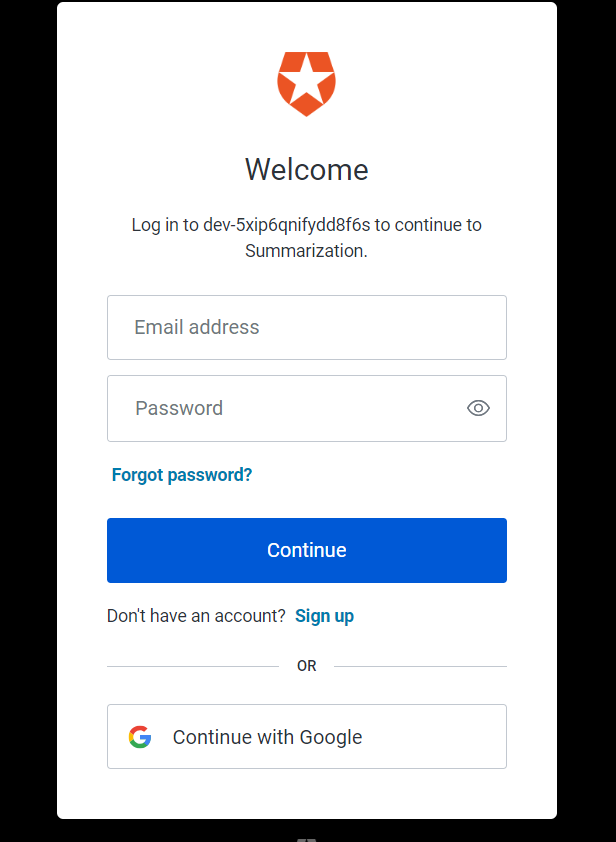
## 5.1. Conclusion

The development of the Online Text Summarization Tool has been a significant endeavor, aiming to provide users with a seamless and efficient means of condensing large textual content. Through the incorporation of diverse summarization algorithms, including default, TextRank, and TF-IDF, the tool caters to varying user preferences and requirements. The project successfully integrates both frontend and backend components, ensuring a user-friendly interface and robust functionality. The system has undergone extensive testing, encompassing unit and system testing to validate its reliability, security, and effectiveness. As a result, the Online Text Summarization Tool emerges as a valuable resource, empowering users to effortlessly distill essential information from extensive texts, promoting accessibility, and enhancing overall user experience.

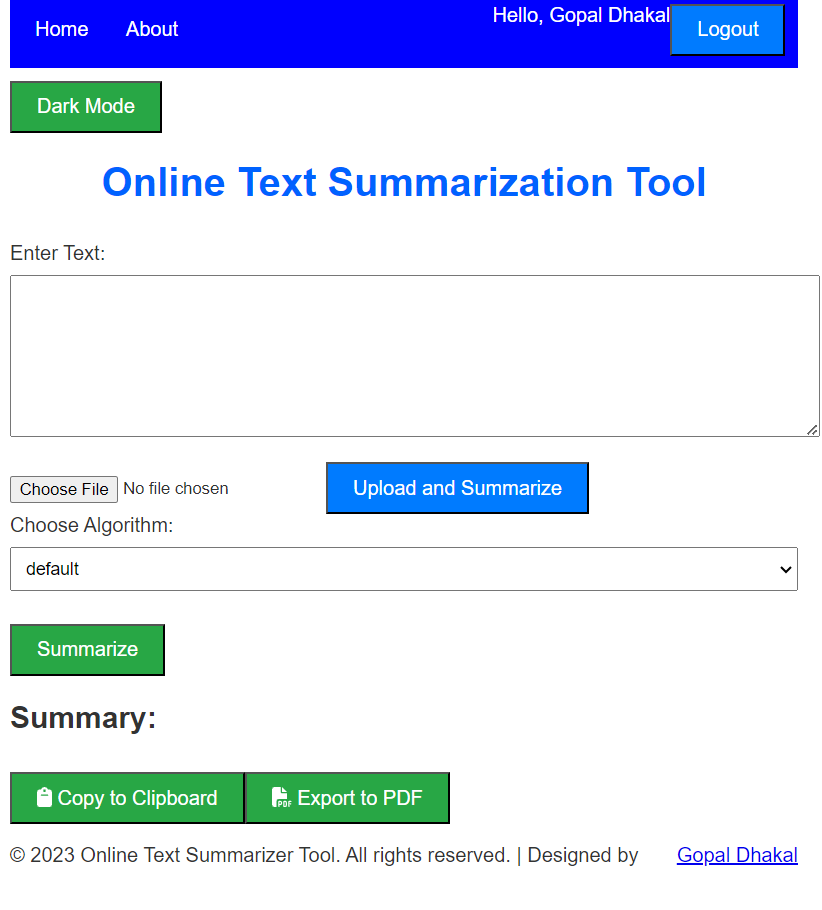
## 5.2. Future Recommendations

Looking ahead, there are several potential enhancements and features that could further elevate the capabilities and user experience of the Online Text Summarization Tool. First and foremost, the inclusion of additional summarization algorithms, such as extractive and abstractive methods, could provide users with a broader range of options to tailor the summarization process to their specific needs. Integration with external APIs for language processing and sentiment analysis could add another layer of sophistication to the tool, enabling users to gain deeper insights into the content they summarize. Additionally, considering the growing importance of multilingual support, incorporating language detection and summarization for diverse languages could enhance the tool's global utility.In conclusion, by incorporating these future recommendations, the Online Text Summarization Tool can evolve into a more comprehensive, user-centric, and feature-rich platform, meeting the diverse needs of users in the realm of text summarization.

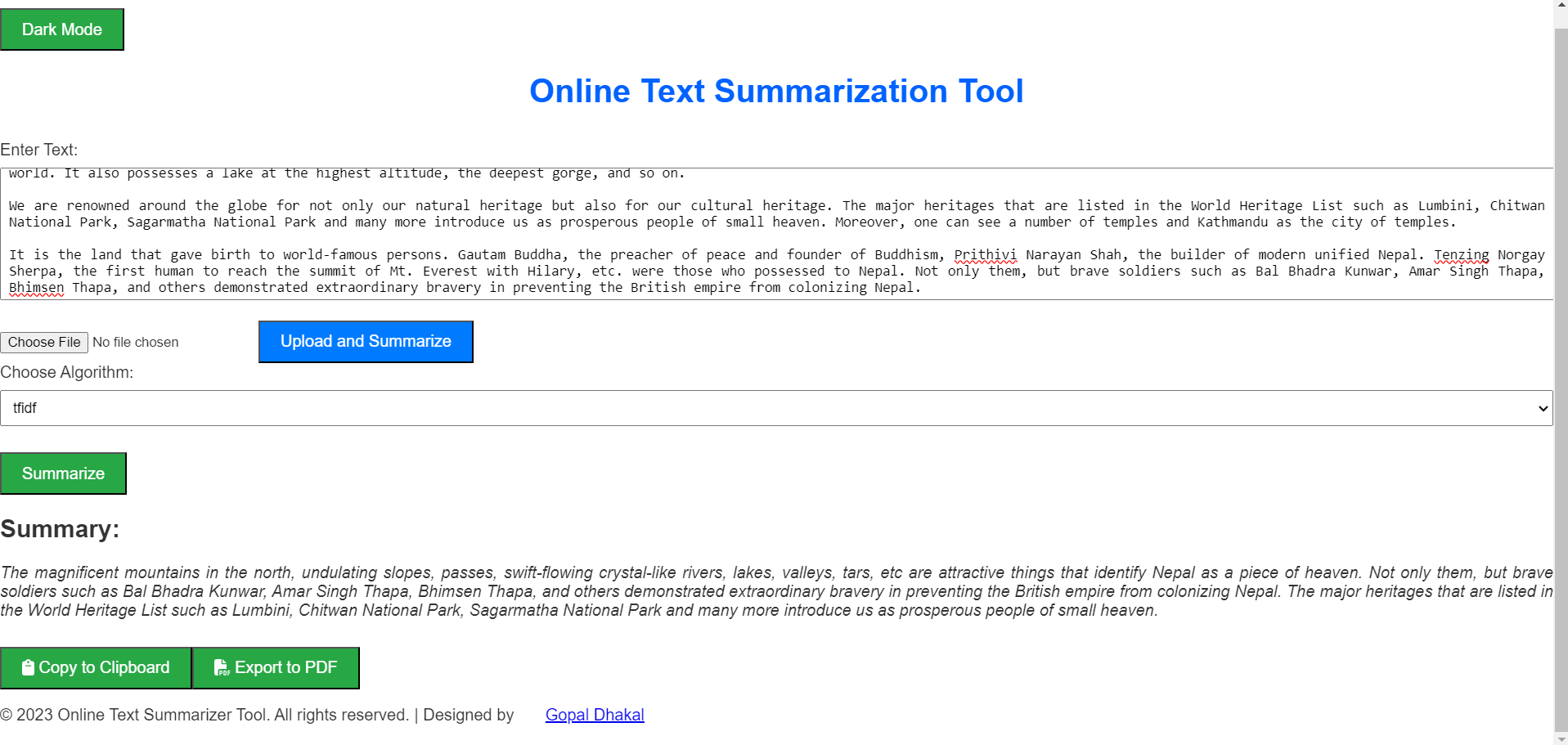
**APPENDICES**



**Figure 5.1 Snapshot of the login page**



**Fig 5.2 Homepage of online text summarization tool**



**Fig 5.3 Implementation of algorithm for summarization**

**REFERENCES**

1. Lutkevich,b (2022).definition and guide to waterfall model. Techtarget.
2. Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media.
3. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
4. Erkan, G., & Radev, D. R. (2004). "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization." Journal of Artificial Intelligence Research.
5. React Documentation. [https://reactjs.org/docs/getting-started.html](https://reactjs.org/docs/getting-started.html" \t "https://chat.openai.com/c/_new)
6. Erkan, G., & Radev, D. R. (2004). "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization." Journal of Artificial Intelligence Research.
7. Auth0 Documentation. [https://auth0.com/docs/](https://auth0.com/docs/" \t "https://chat.openai.com/c/_new)
8. Flask-SQLAlchemy Documentation. [https://flask-sqlalchemy.palletsprojects.com/](https://flask-sqlalchemy.palletsprojects.com/" \t "https://chat.openai.com/c/_new)