

ASTRAY

HARISANKAR K PRASAD

Reg No : 33218812011

NIKHIL PRAKASH

Reg No : 33218812017

SREYAS S

Reg No : 33218812021

Under the guidance of

Mr. BHAVIN B

Assistant Professor

Department of Computer Science and Applications



NAIPUNNYA SCHOOL OF MANGEMENT

CHERTHALA

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUEREMENTS

THE AWARD OF BACHELOR OF COMPUTER APPLICATIONS OF

UNIVERSITY OF
KERALA

2021

NAIPUNNYA SCHOOL OF MANAGEMENT

CHERTHALA



CERTIFICATE

Certified that this report titled **ASTRAY** is a bonafide record of the project work carried out by **Harisankar K Prasad, Nikhil Prakash** and **Sreyas S** under my supervision and guidance, towards partial fulfillment of the requirements for the award of the **Degree of Bachelors of Computer Applications** of the **University of Kerala** and no part thereof has been submitted for and other similar recognition.

Mr. BHAVIN B
Project guide

Mr. VINOD CHANDRAN
H.O.D

REV Fr. BAIJU GEORGE
Principal

Presented for the viva voice examination conducted by the University of Kerala held on
...../...../2021 at Naipunnya School of Management and verified by:

Internal Examiner

External Examiner

Place : Cherthala

Date :



CERTIFICATE

This is to certify that **Mr. HARISANKAR K PRASAD (Reg No: 33218812011), Mr. NIKHIL PRAKASH (Reg No: 33218812017), and Mr. SREYAS S (Reg No: 33218812021)** Bachelor of Computer Applications VI semester students of **NAIPUNNYA SCHOOL OF MANAGEMENT CHERTHALA** affiliated to University of Kerala, has done project work entitled **"ASTRAY THE COMPLETE SUMMARIZER"** in PYTHON under the guidance of our senior faculties towards the fulfillment of the award of "Bachelor of Computer Applications" during the period January 2021 to March 2021.

They successfully completed the project and during the period they were methodical and hardworking.

For RISS TECHNOLOGIES

Chief Executive Officer



DECLARATION

We hereby declare that this project work and the report entitled “**ASTRAY**” submitted to the **Department of Computer Science and Applications, Naipunnaya School of Management, Cherthala** in partial fulfillment of the award of degree of **BACHELOR OF COMPUTER APPLICATIONS** is an outcome of our own work.

We also declare that this report has not submitted to any other University or Institute for the award of any fellowship or degree or diploma.

To the best of our knowledge this project work or parts there, does not form a part of any other project work or thesis on the basis of which a degree or award was conferred on an earlier occasion.

Date :

Place : Cherthala

Harisankar K Prasad

Nikhil Prakash

Sreyas S

ACKNOWLEDGEMENT

First of all, we thank and praise God Almighty for being with us right from the day we started the project.

We express our sincere thanks to Rev. Fr. Baiju George, Principal, Naipunnya School of Management. We express our sincere thanks and deep sense of gratitude to Mr. Vinod Chandran, Head of the department of Computer Science for providing all sorts of support and necessary facilities throughout the course and for the inspiration to complete our project.

We would like to express our sincere gratitude for the valuable guidance and help provided by our project guide Mr. Bhavin B, Assistant Professor, Dept. of Computer Science and Applications, whose timely suggestions, advice and guidance led to successful completion of our project. We are extremely thankful to the staff of Naipunnya School of Management for all their immense support and help.

Last but not least we are also grateful to all our friends and families who gave suggestions, support and kind presence and had contributed a lot to the fulfillment of the project.

ABSTRACT

"**ASTRAY**" is a text summarization software which can be used for creating an abstract. In this project, a text summarization system is developed. The system works by assigning scores to sentences in the document to be summarized, and using the highest scoring sentences in the document. Score values are based on features extracted from the sentence. A linear combination of feature scores is used. Almost all of the mappings from feature to score and the coefficient values in the linear combination are derived from a training corpus. Some anaphor resolution is performed. In addition to basic summarization, some extra features are also added like, receiving the summarized output in audio format and also translating the output into some specified language. The intended user is considered to have little background knowledge or reading ability.

CONTENTS

SLNO	TOPIC	PAGE NUMBER
1.	INTRODUCTION	1
2.	SYSTEM ANALYSIS	2
	2.1 Introduction	2
	2.2 Existing System	4
	2.3 Proposed System	6
	2.4 Feasibility Study	7
	2.4.1 Technical Feasibility	7
	2.4.2 Operational Feasibility	7
	2.4.3 Economical Feasibility	8
3.	SYSTEM SPECIFICATION	9
	3.1 Software Requirements	10
	3.2 Hardware Requirements	11
	3.3 Technologies Used	12
	3.3.1 MySQL	12
	3.3.2 Python	12
	3.3.3 HTML	13
4.	BACKGROUND	15
5.	SYSTEM DESIGN	19

5.1 Input Design	21
5.2 Output Design	22
5.3 Data Flow Diagram	23
5.4 Database Design	35
6. MODULE DESCRIPTION	39
7. SYSTEM TESTING	41
7.1 Unit Testing	42
7.2 Integration Testing	42
7.3 System Testing	43
8. SYSTEM IMPLEMENTATION	44
9. SYSTEM MAINTENANCE	45
10. FUTURE ENHANCEMENT	46
11. CONCLUSION	47
12. APPENDIX	48
12.1 SAMPLE INPUT SCREEN	48
12.2 Process Description	61
13. BIBLLIOGRAPHY	102
14. WEBLIOGRAPHY	103

1. INTRODUCTION

"ASTRAY The Complete Summarizer" is a text summarization software which can be used for creating an abstract.

Text summarization is basically summarizing the given paragraph using natural language processing and machine learning. There has been an explosion in the amount of text data from a variety of sources. This volume of text is an invaluable source of information and knowledge which needs to be effectively summarized to be useful. In this review, the main approaches to automatic text summarization are described.

We review the different processes for summarization and describe the effectiveness and shortcomings of the different methods. Two types will be used i.e.-extractive approach and abstractive approach. The basic idea behind summarization is finding the subset of the data which contains the information of all the set. There is a great need to reduce unnecessary data. It is very difficult to summarize the document manually so there is the great need of automatic methods. Approaches have been proposed inspired by the application of deep learning methods for automatic machine translation, specifically by framing the problem of text summarization as a sequence-to-sequence learning problem.

In this project, a text summarization system is developed. The system works by assigning scores to sentences in the document to be summarized, and using the highest scoring sentences in the document. Score values are based on features extracted from the sentence. A linear combination of feature scores is used. Almost all of the mappings from feature to score and the coefficient values in the linear combination are derived from a training corpus. Some anaphor resolution is performed. In addition to basic summarization, some extra features are also added like, receiving the summarized output in audio format and also translating the output into some specified language. The intended user is considered to have little background knowledge or reading ability.

2. SYSTEM ANALYSIS

2.1 Introduction

System analysis refers to an orderly structured process for identifying and solving problems using computer. It is the most essential part of the project development. It is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements to the system.

System analysis is a step –by-step process used to identify and develop or acquire the software need to control the processing of specific application. System analysis is a continuing activity the stages of the systems development. Once the existing system is studied and user requirements are identified, a more detailed analysis is carried out .System analysis is the process of dividing the system into parts and establishing relationships among them. It involves the existing system, proposed system, software requirement specification (SRS), feasibility analysis and Data flow diagram (DFD).It is the detailed study of the various operations performed by the system and their relationships within and outside the system. The various tasks in the system analysis include the following.

- Understanding application.
- Planning.
- Scheduling.
- Developing candidate solution.
- Performing candidate studies.
- Performing cost benefit analysis.
- Recommending alternative solutions.
- Selling of the system.
- Supervising, installing and maintaining the system.

System analysis involves the study of an application area to fully understand the problem being posed. Activities are focused on developing a comprehensive knowledge of the existing system, its strengths and weaknesses and the reasons for the need to restructure, replace or automatic the existing system. Risk assessments and risk containment plan, cost estimation and plans for the remainder of the development are results usually prepared by the system analyst as a buy-product of system analysis.

In this section we will analyze the requirements needed for our auto share. The Existing System, Proposed System, System Specifications, Operating System, Software package, Hardware and Software Specifications.

2.2 EXISTING SYSTEM

Before the selection of our project a thorough study was made on the existing system. The existing software is a text summarizer software used to create summary of a given paragraph/content. It consumes more time for research and for compressing texts. The existing software's are paid software's and price are charged based on the length of the content to be summarized. As the length of the content increases the chances for occurring error will also be increased.

Traditional System

Manual Methods

One of the most commonly used traditional method is manual summarization. The most obvious and simple way to evaluate a summary is to have assessors evaluating its quality. For example for DUC, the judges had to evaluate the coverage of the summary, which means they had to give a global score assessing to what extent the candidate summary covers the text given as input. In more recent frameworks, and especially in TAC, query-oriented summaries have to be produced: judges then have to evaluate the responsiveness of the summary, that is to say to what extent a given summary answers the query given in input. Manual evaluation can also provide some indicators to assess the quality and readability of a text. A good summary is supposed to be:

- Syntactically accurate
- Semantically coherent
- Logically organized
- Without redundancy

These different points are too complex to be fully automatically calculated, especially semantic coherence and logical organization. In order to get a reliable evaluation of these different aspects, it is necessary to get human judgements. For the DUC and TAC campaigns, human experts had to give different scores to each candidate summary, using the following indicators:

- Grammaticality
- Non redundancy
- Focus (integration of the most important pieces of information of the original text)
- Structure and coherence.

Experts had to give a grade between 0 (void) and 10 (perfect) for each of these indicators. For TAC reference summaries written by human experts got an average grade of 8.8/10. Thus, this grade can be seen as the upper bound score reachable by candidate summaries.

Limitations of existing system

- Time consuming
- Paid Software (High cost)
- Number of existing software is comparatively less
- Low accuracy

2.3 PROPOSED SYSTEM

The proposed system is a text summarization software, which can be used to produce an abstract of a detailed paragraph/ web page/ blog/ e-book etc. The system works by assigning scores to sentences in the document to be summarized, and using the highest scoring sentences in the document. Score values are based on features extracted from the sentence. A linear combination of feature scores is used. Almost all of the mappings from feature to score and the coefficient values in the linear combination are derived from a training corpus. Some anaphor resolution is performed. In addition to basic summarization, some extra features are also added like, receiving the summarized output in audio format.

2.4 FEASIBILITY STUDY

Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation.

- Technical feasibility
- Operation feasibility
- Economic feasibility

2.4.1 Technical Feasibility

Earlier no system existed to cater to the needs of „Secure Infrastructure Implementation System“. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users of accuracy, reliability and security, would be granted based on the roles specified. Therefore, it provides the technical guarantee.

The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

2.4.2 Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation.

This system is targeted to be in accordance with the above mentioned issues. Beforehand, the management issues and the user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

Proposed project is beneficial only if it can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed.

2.4.3 Economic Feasibility

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. it does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, there is nominal expenditure and economic feasibility for certain.

Evaluating the technical feasibility is the trickiest part of a feasibility study, This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance ,costs on(on account of the kind of technology to be deployed)etc. A number of issues have to be considered while doing a technical analysis. Understand the different technologies involved in the proposed system before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the new system. Find out whether the organization currently processes the required technologies.

3. SYSTEM SPECIFICATION

A System Requirements Specification is a structured collection of information that embodies the requirements of a system. System specification specifies the hardware and software configuration of the new proposed system. To develop application software, we use different type of software. The software for the development has been selected based on several factors such as

- Support and stability
- Cost of effectiveness
- Development speed
- Ability to create robust application in least time.

The system requirements of the project includes

- Software specification
- Hardware specification

3.1 SOFTWARE REQUIREMENTS

A software requirement specification (SRS), a requirements specification for a software system, is a complete description of the behavior of the system to be developed and may include a set of use cases that describes interactions the users will have with the software. In addition it also contains non-functional requirements. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints) the software requirement specification document enlists all necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed.

Operating System	:	WINDOWS 7 & Above for better performance
Front end	:	PYTHON(web application)
Back end	:	MYSQL
Software	:	WAMP, Sublime 3
Web Browser	:	Internet Explorer/Google Chrome/Firefox
Web Server	:	Apache

3.2 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

A hardware requirements list is often accompanied by a Hardware Compatibility List (HCL), especially in case of operating systems. An HCL lists tested, compatible and sometimes incompatible hardware devices for a particular operating system or application.

Processor	:	Intel Pentium or above
Hard Disc	:	320 GB
Display Type	:	PC Display
Keyboard	:	PC/AT Enhanced PS/2Keyboard (110/10Key).
Mouse	:	First/Pilot Mouse Serial (c48)
Input Device	:	Mouse, keyboard
Output Device	:	Monitor, Mobile Display

3.3 TECHNOLOGIES USED

3.3.1 MySQL

MySQL is an open-source relational database management system(RDBMS).Its name is a combination of “My” ,the name of co-founder Michael Widener’s daughter ,and “SQL”, the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Cooperation).In 2010, when oracle acquired Sun; Widenius forked the open source MySQL Project to create Maria DB.

MySQL is component of the LAMB web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and Word Press.

The first version of MySQL appeared on 23 may 1995.It was initially created for personal usage from MySQL based on the low-level language ISAM, which the creators considered too slow and inflexible. They created a new SQL interface, while keeping the same API as MySQL. By keeping the API consistent with the MySQL system, many developers were able to use MySQL instead of the (proprietary licensed) MySQL antecedent.

3.3.2 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are

available in source or binary form without charge for all major platforms, and can be freely distributed.

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

3.3.3 HTML

HTML stands for Hypertext Mark-up Language. It allows the user to create and structure sections, paragraphs, headings, links, and block quotes for web pages and applications. HTML is not a programming language, meaning it doesn't have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word.

HTML documents are files that end with a .html or .htm extension. You can view them using any web browser (such as Google Chrome, Safari, or Mozilla Firefox). The browser reads the HTML file and renders its content so that internet users can view it. Usually, the average website includes several different HTML pages. For instance: home pages, about pages, contact pages would all have separate HTML documents.

Each HTML page consists of a set of tags (also called elements), which you can refer to as the building blocks of web pages. They create a hierarchy that structures the content into sections, paragraphs, headings, and other content blocks.

Different Versions of HTML

- HTML 1.0
- HTML 2.0
- HTML 3.2
- HTML 4.01
- HTML 5

4. BACKGROUND

This section explores the technologies which were used in this project. The section first discusses the key concepts for text summarization, followed by the metrics used to evaluate them along with the environments and the libraries used to complete this project.

Natural Language Processing

Natural Language Processing (NLP) is a field in Computer Science that focuses on the study of the interaction between human languages and computers. Text summarization is in this field because computers are required to understand what humans have written and produce human-readable outputs. NLP can also be seen as a study of Artificial Intelligence (AI). Therefore many existing AI algorithms and methods, including neural network models, are also used for solving NLP related problems.

Text Extraction

Extractive summarization means extracting keywords or key sentences from the original document without changing the sentences. Then, these extracted sentences can be used to form a summary of the document.

Text rank

Textrank is an algorithm inspired by Google's PageRank algorithm that helps identify key sentences from a passage. The idea behind this algorithm is that the sentence that is similar to most other sentences in the passage is probably the most important sentence in the passage. Using this idea, one can create a graph of sentences connected with all the similar sentences and run Google's PageRank algorithm on it to find the most important sentences. These sentences would then be used to create the summary.

TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is used to determine the relevance of a word in the document. The underlying algorithm calculates the frequency of the word in the document (term frequency) and multiplies it by the logarithmic function of the number of documents containing that word over the total number of documents in the dataset (inverse document frequency). Using the relevance of each word, one can compute the relevance of each sentence. Assuming that most relevant sentences are the most important sentences, these sentences can then be used to form a summary of the document.

Text Abstraction

Compared to extractive summarization, abstractive summarization is closer to what humans usually expect from text summarization. The process is to understand the original document and rephrase the document to a shorter text while capturing the key points. Text abstraction is primarily done using the concept of artificial neural networks. This section introduces the key concepts needed to understand the models developed for text abstraction.

Word Embedding

Word embedding is a set of feature learning techniques in NLP where words are mapped to vectors of real numbers. It allows similar words to have similar representation, so it builds a relationship between words and allows calculations among them. A typical example is that after representing words to vectors, the function “king - men + women” would ideally give the vector representation for the word “queen”. The benefit of using word embedding is that it captures more meaning of the word and often improves the task performance, primarily when working with natural language processing.

LIBRARIES

Keras

Keras is a Python library initially released in 2015, which is commonly used for machine learning. Keras contains many implemented activation functions, optimizers, layers, etc. So, it enables building neural networks conveniently and fast. Keras was developed and maintained by François Chollet, and it is compatible with Python 2.7-3.6.

NLTK

Natural Language Toolkit (NLTK) is a text processing library that is widely used in Natural Language Processing (NLP). It supports the high-performance functions of tokenization, parsing, classification, etc. The NLTK team initially released it in 2001 (Nltk.org, 2018).

Scikit-learn

Scikit-learn is a machine learning library in Python. It performs easy-to-use dimensional reduction methods such as Principal Component Analysis (PCA), clustering methods such as k 10 means, regression algorithms such as logistic regression, and classification algorithms such as random forests.

Flask

Flask, issued in mid-2010 and developed by Armin Ronacher, is a robust web framework for Python. Flask provides libraries and tools to build primarily simple and small web applications with one or two functions.

Bootstrap

Bootstrap is an open-source JavaScript and CSS framework that can be used as a basis to develop web applications. Bootstrap has a collection of CSS classes that can be directly used to create effects and actions for web elements. Twitter's team developed it in 2011.

LXML

The XML toolkit LXML, which is a Python API, is bound to the C libraries libXML2 and libxslt. LXML can parse XML files faster than the Element Tree API, and it also derives the completeness of XML features from libXML2 and libxslt libraries

5. SYSTEM DESIGN

System architecture or system's architecture is the conceptual mode that defines the structure, behavior and more views of a system. An architecture description is a formal description and representation of a system, organized in very that supports reasoning about the structure of the system.

System architecture can comprise system components, the externally visible properties of those components, the relationships (ex: the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called as architectural description languages (ADLs).

The system architecture can best be thought of as a set of representation of an existing (or to be created) system. It is used to convey the informational content of the elements comprising a system, the relationship among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that architecture describes may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organization or people. System architecture is primarily concerned with the internal interfaces among the system and its external environment, especially the user.

The structural design reduces complexity, flexibility change and result in easier implemented by encouraging parallel development of different parts of the system. The procedural design transforms structural elements of program architecture into a procedural description of software components. The architectural design considers architecture as the most important functional requirements. The system is based on three-tier architecture.

The first level is the user interface (presentation logic), which displays control, receives and validates user input. The second level is the business layer (business logic) where the application specific logic takes place. The third level is the data layer where the application information is stored in files or database. It contains logic about to retrieve and

update. The important feature about the three-tier design is that information only travels from one level to an adjacent level.

System design transforms a logical representation of what a given system is required to doing the physical specification. The specifications are converted into a physical reality during development. The design forms a blueprint of the system and the components related to each other. The design phrase proceeds according to an orderly sequence of steps beginning with review and assignments of tasks and ending with package design.

5.1 INPUT DESIGN

Input design is a part of the overall design. The input methods can be broadly classified into batch and online. Internal controls must be established for monitoring. Input design is the link that ties the information system into the world of its users. The input design involves determining what the input is, how the data should be performed, how to validate data, how to do data-entry, and how to provide multi-user facility. Inaccurate input data are the most common cause of errors in data processing. Input design is the process of converting user originated input to a computer-based format. Input data are collected and organized into groups of similar data. Once identified, appropriate input media are selected for processing.

5.2 OUTPUT DESIGN

A quality output design is which meets the requirements of end users and present the information clearly. In any system result of processing are communicated to the user and the other system through outputs. In the output design it is determined how the information is to be displayed for immediate need.

It is the most important and direct source information is to the user. Efficient and intelligent output design improves the system's relationships with the user and helps indecision making. The objective of the output design is to convey the information of all the past activities, current status and to emphasis important events. The output generally refers to the results and information that is generated from the system. Outputs from computers are required primarily to communicate the results of processing to the users.

Output also provides a means of storage by copying the results for later references in consultation. There is a chance that some of the end users will not actually operate the input data or information through work stations but will see the output from the system.

Two phases of the output design are: -

- Output Definition
- Output Specification

Output Definition takes into account the type of output contents. Its frequency and its volume, the appropriate output media is determined for output. Once the media is chosen, the detailed specifications of output documents are carried out. The nature of output required from the proposal system is determined during logical design stage. It takes the outline of the output from the logical design phase during. In a project, when designing the output, the system analyst must accomplish.

5.3 DATA FLOW DIAGRAMS

A Data Flow Diagram is a network that describes the flow of data and processes that change, or transform, data throughout the system. This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which a data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs.

There are various symbols used in a DFD. Bubbles represent the processes. Named arrows indicate the data flow. External entities are represented by the rectangles. Entities supplying data are known as sources and those that consume data are known as sinks. Data are stored in a data store by a process in the system. Each component in a DFD is labelled with a descriptive name. Processes named are further identified with a number. The Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system. For a candidate system, it describes the input (source), outputs(destination), database(files)and procedures (data flow), all in a format that meet the user's requirements. The main merit of DFD is that it can provide an overview of the system requirements, what data a system would process, what transformations of data are done, what files are used, and where the results flow.

Rules for Constructing a Data Flow Diagram

- Arrows should not cross each other.
- Squares, circles and files must bear names.
- Choose meaningful names for data flow.
- Draw all data flows around the outside of the diagram.
- Decomposed dataflow squares and circles can have same time.

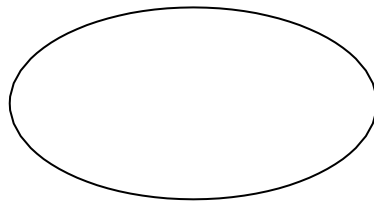
Basic Data Flow Symbols

1. Arrow:



A data flow is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a process and from data store or process. An arrow line depicts the flow, with arrow head pointing in the direction of the flow.

2. Ellipse



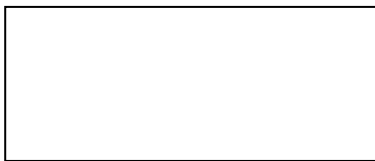
It stands for process that converts data into information. A process represents transformation where incoming data flows are changed into outgoing data flows.

3. Slide open rectangle:



A data store is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names. If a process merely uses the content of store and does not alter it, the arrowhead goes only from the store to the process. If a process alters the details in the store then a double-headed arrow is used.

4. Rectangle:

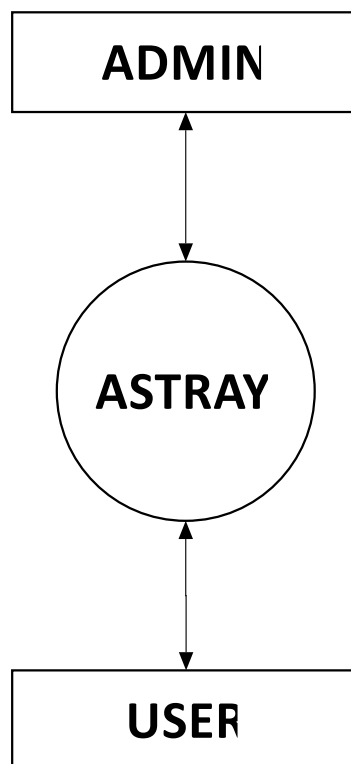


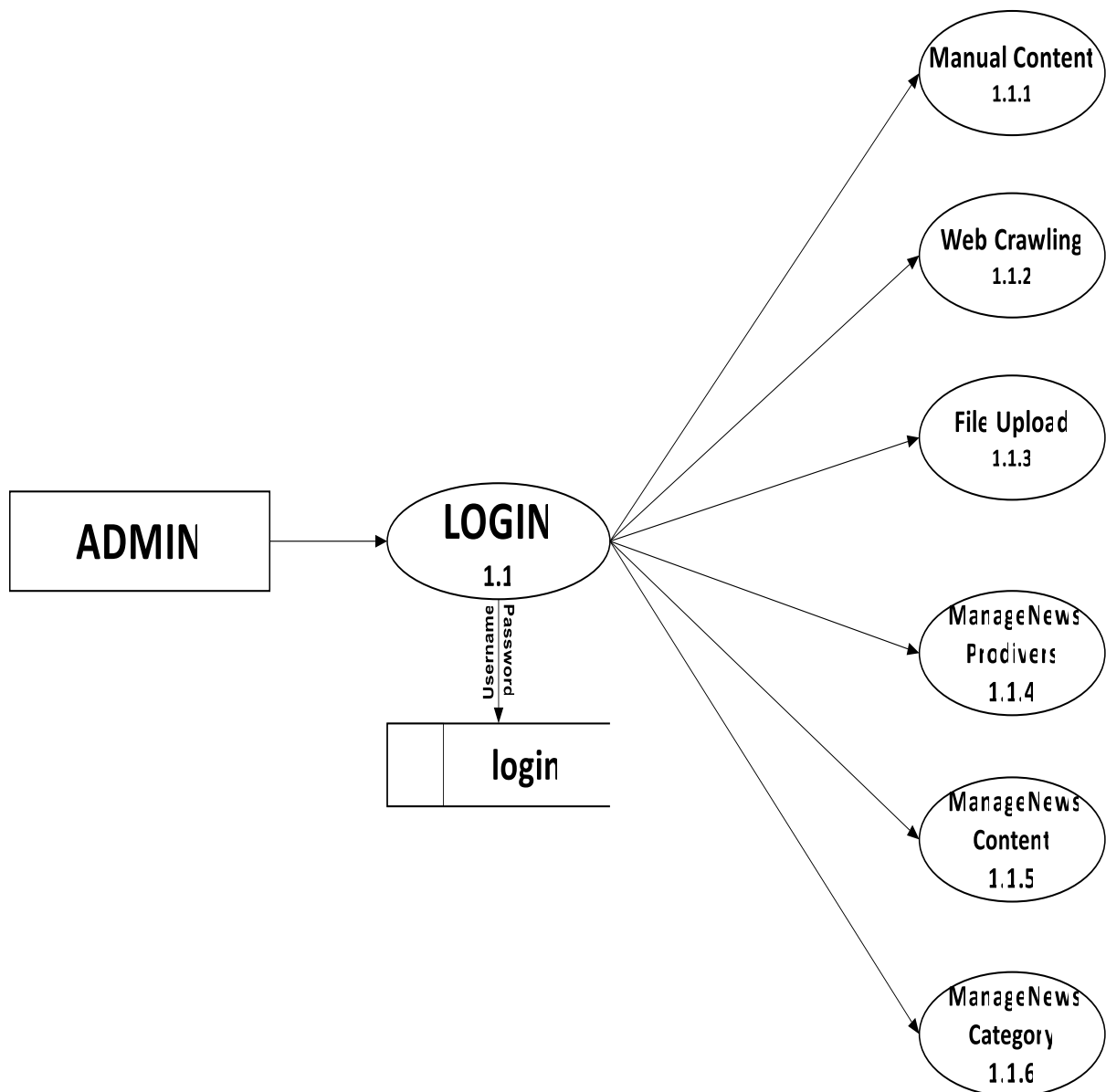
A source or sink is a person or part of an organization, which enters or receives information from the system, but is considered to be outside the context of data flow model.

Each component in a DFD is labelled with a descriptive name, Process name are further identified with number. Context level DFD is draw first. Then the process is decomposed into several elementary levels and is represented in the order of importance. A DFD describes what data flow (logical) rather than how they are processed, so it does not depend on hardware, software, and data structure or file organization.

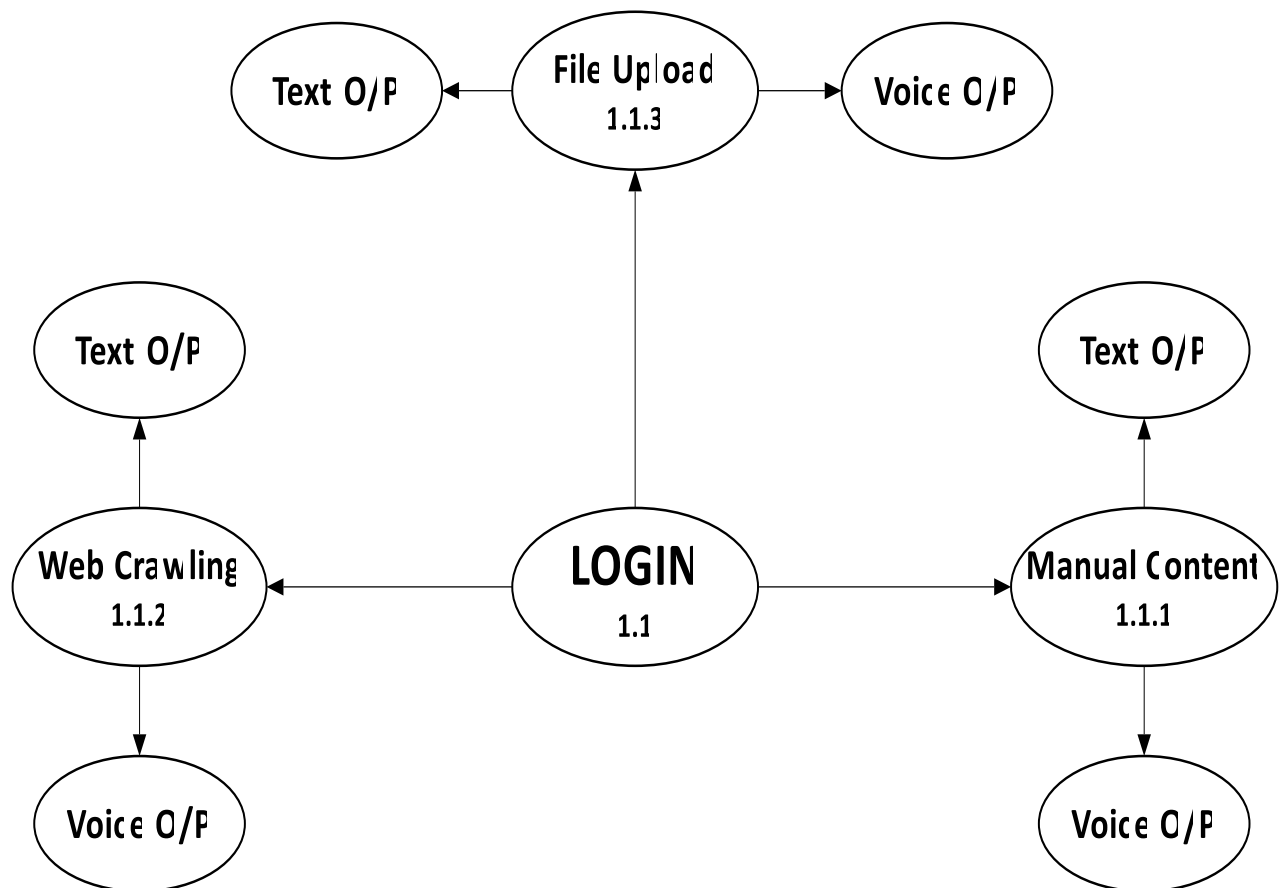
A DFD methodology is quite effective; especially when the required design is clear and the analyst need a notation language for communication. The DFD is easy to understand after a brief orientation.

DATA FLOW DIAGRAM

LEVEL 0

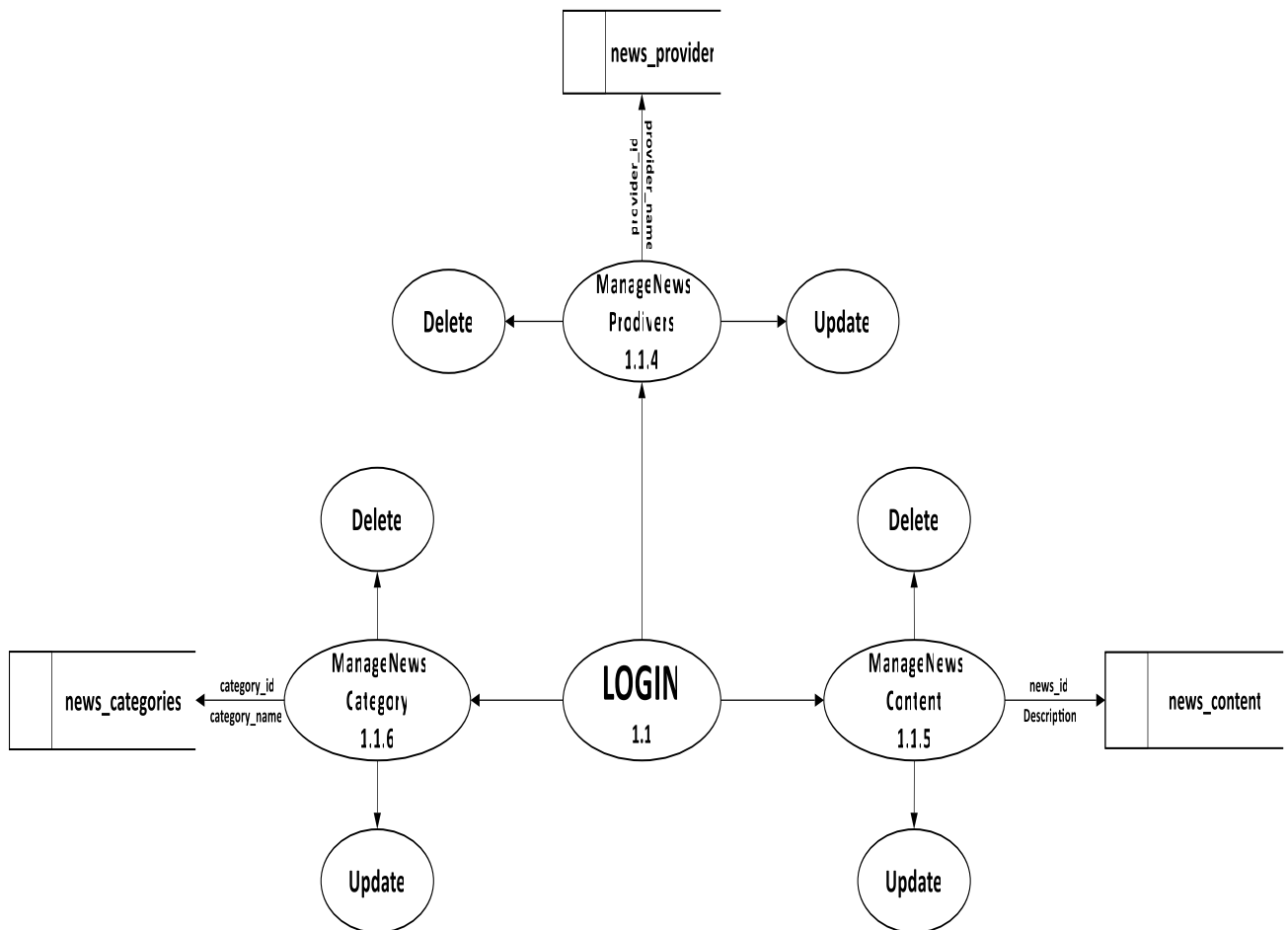
LEVEL 1 (ADMIN)

LEVEL 1.1
ADMIN (PART - 1)



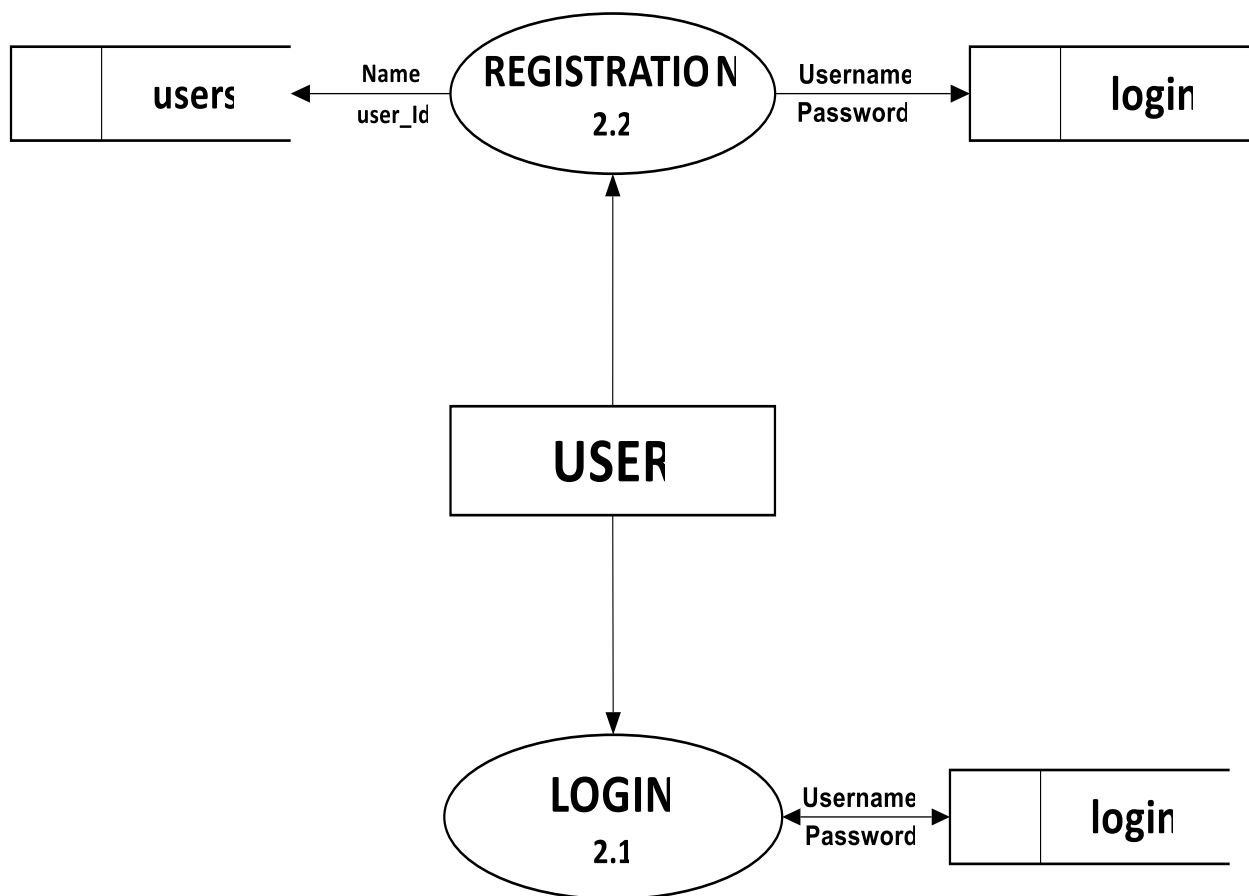
LEVEL 1.1

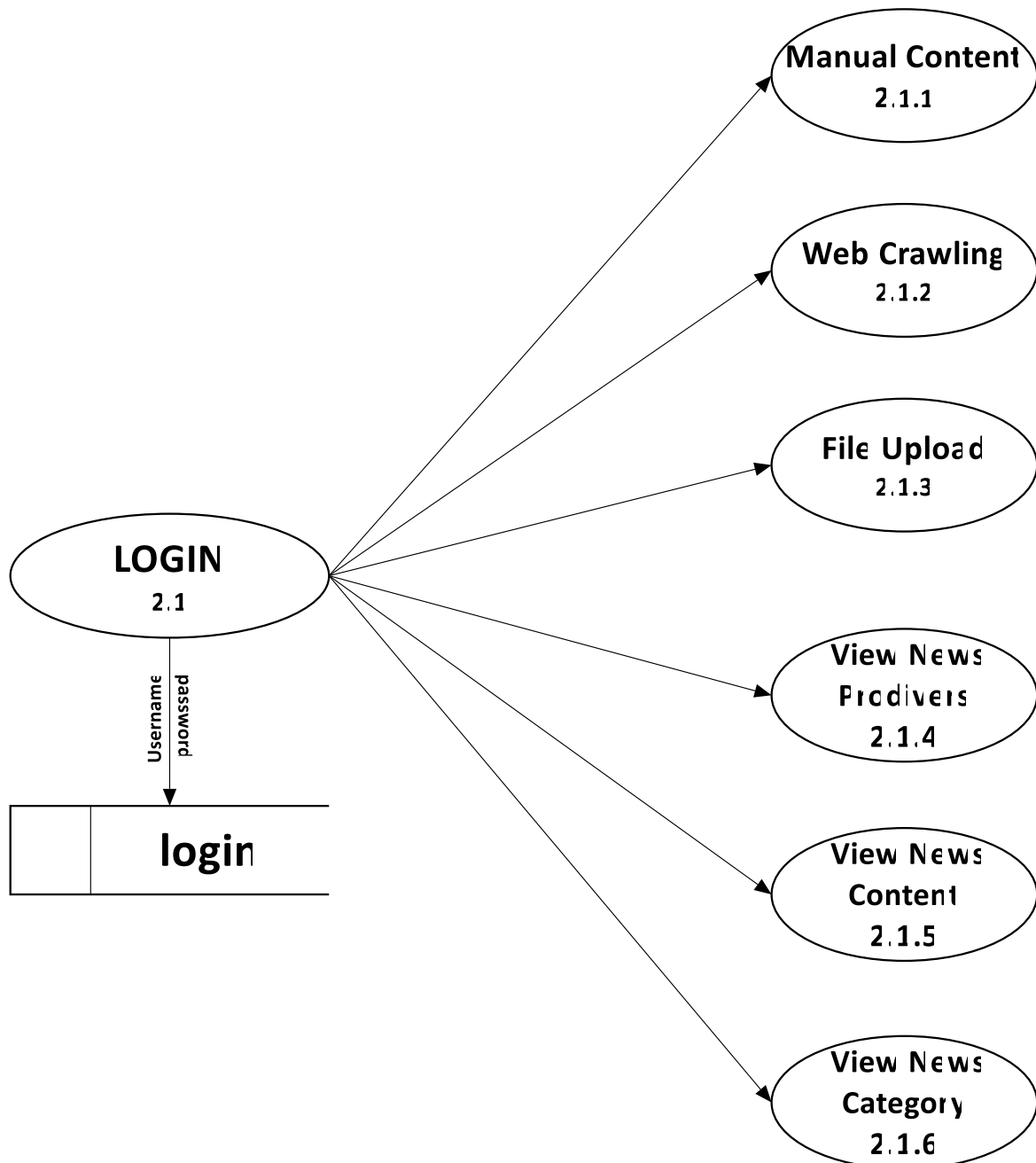
ADMIN (PART - 2)



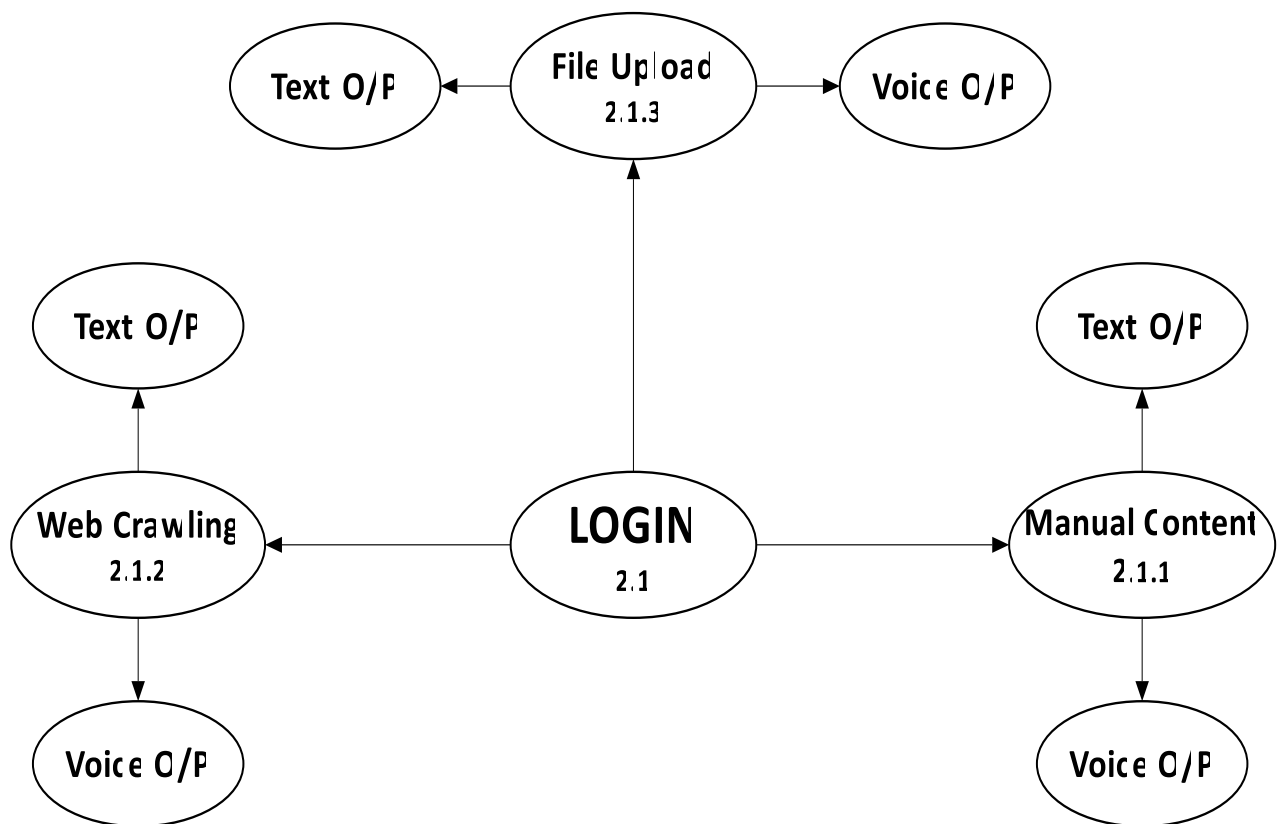
LEVEL 2

USER



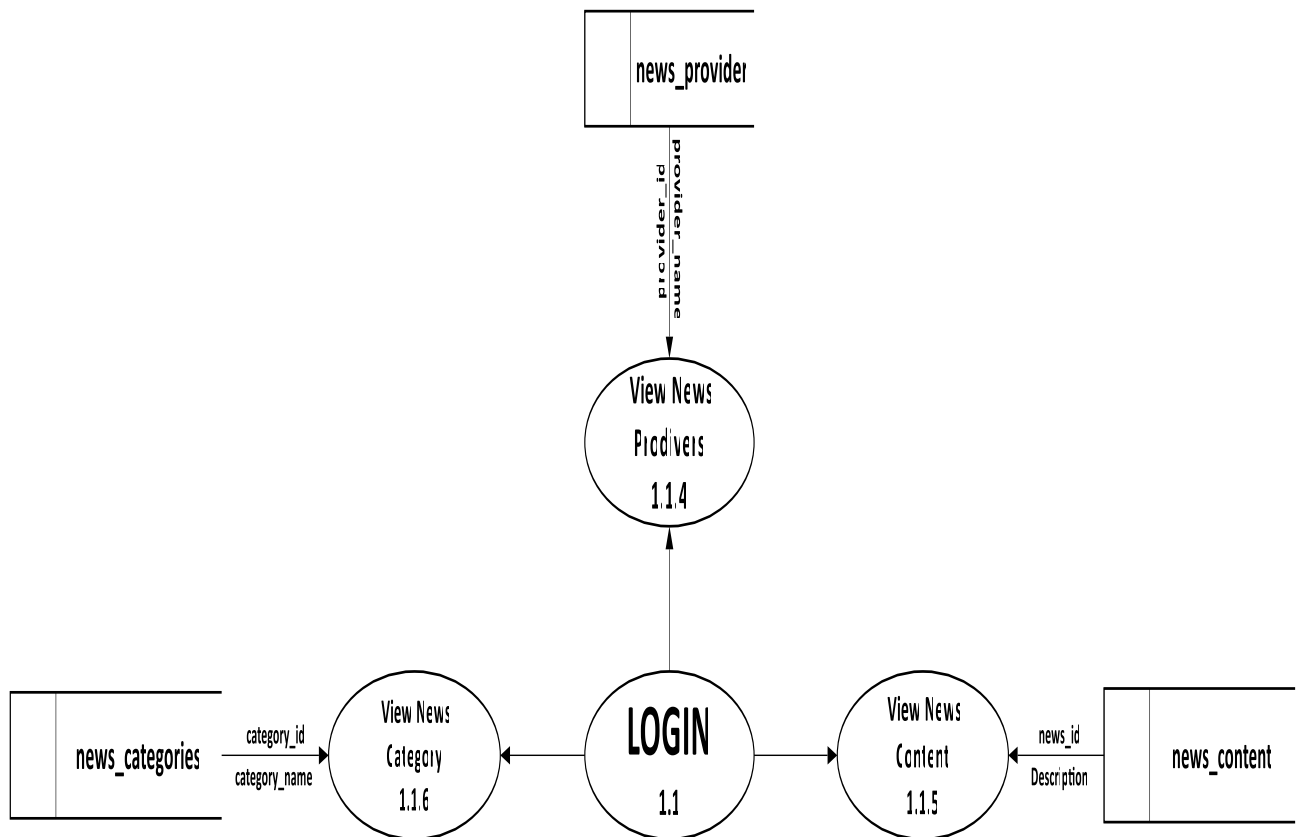
LEVEL 2.1**USER**

LEVEL 2.1
USER (PART - 1)



LEVEL 2.1

USER (PART - 2)



5.4 DATABASE DESIGN

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data.

A DBMS relieves users of framing programs for data maintenance. Fourth-generation query languages, such as SQL, are used along with the DBMS package to interact with a database. Some of features of DBMs are: -

- A DBMS uses various powerful functions to store and retrieve data efficiently.
- Offers Data Integrity and Security.
- DBMS offers a variety of techniques to store & retrieve data.
- Uniform administration procedures for data.
- Application programmers never exposed to details of data representation and storage.

TABLES

Table No : 1

Name : Login

Primary Key : Login_id

Field Name	Type	Constraint	Size
Login_id	Integer	Primary Key	11
Username	varchar	Not null	30
Password	varchar	Not null	30
User_type	varchar	Not null	30

Table No : 2

Name : Users

Primary Key : user_id

Field Name	Type	Constraint	Size
user_id	Integer	Primary key	11
login_id	Integer	Foreign Key	11
First_name	Varchar	Not null	50
last_name	Varchar	Not null	50
phone	Varchar	Not null	20
Email	Varchar	Not null	50
Place	Varchar	Not null	60

Table No : 3

Name : news_providers

Primary Key : provider_id

Field Name	Type	Constraint	Size
provider_id	Integer	Primary key	11
provider_name	Varchar	Not null	100
provider_contact	Varchar	Not null	100
Description	Varchar	Not null	100

Table No : 4

Name : news_categories

Primary Key : category_id

Field Name	Type	Constraint	Size
category_id	Integer	Primary key	11
category_name	Varchar	Not null	100
category_description	Varchar	Not null	100

Table No : 5

Name : news_content

Primary Key : news_id

Field Name	Type	Constraint	Size
news_id	Integer	Primary key	11
category_id	Integer	Not null	11
Title	Varchar	Not null	30
Date	Varchar	Not null	20
Description	Varchar	Not null	60

Table No : 6

Name : feedback

Primary Key : feedback_id

Field Name	Type	Constraint	Size
feedback_id	Integer	Primary Key	11
user_id	Integer	Not Null	11
feedback	Varchar	Not Null	50
date	Varchar	Not Null	50

6. MODULE DESCRIPTION

There are 2 modules under this application and they are: -

6.1 ADMIN

- Login
- Manage news providers
- Manage news categories
- Manage news content
- Manual news content uploading
- Using web crawling by providing URL
- Upload Document File

6.2 USER

- Register
- Login
- View news providers
- View news categories

ADMIN

Login

The authority can login to this app using their password and user type.

Manage news providers

Here the admin can manage user details

Manage news categories

Here the admin can manage news categories

Manage news content

This phase includes the core features of this software. It includes the following:-

Manual news content uploading

Input the content to be summarized in the form of text.

Using web crawling by providing URL

Input the URL of a blog / web page to be summarized

Upload Document File

Input the content to be summarizer in the form of document (docx , pdf).

USER**Register**

User can register by entering their details

Login

The authority can login to this app using their password and user type.

View news providers

Here the user can view the details of news providers.

View news categories

Here the user can view news category and other details.

Manual news content uploading

Input the content to be summarized in the form of text.

Using web crawling by providing URL

Input the URL of a blog / web page to be summarized

Upload Document File

Input the content to be summarizer in the form of document (docx , pdf).

7. SYSTEM TESTING

System testing is actually a series of different testes whose primary purpose is to fully exercise the computer-based system. All though each test has a different purpose, all work to verify that all system elements have been properly integrated and perform all allocated functions. The following are the main objectives of testing.

Testing is the final verification and validation activity within the organization itself. Testing is vital to success of the system. System testing makes logical assumption that if all parts of the system are correct, the goal will be successfully achieved. Another reason for system testing is utility as a user-oriented vehicle before implementation.

1. Testing is process of executing a program with the intent of finding errors.
2. A new test case is one that has a high probability of finding errors.
3. It is a set of activities that can be planned in advance and conducted automatically.

Testing Process

For any software that is newly developed, first and foremost difference is given to the testing of the system. It is developer's last chance to detect and correct the errors. That may occur possibly in the software. The programmers will generate a set of test data, which will give the maximum possibility of finding all most all types of errors that can occur in the system. There are 3 types of testing- unit testing, integration testing and system testing. A series of tests are performed for the proposed system before the system is ready for user acceptance testing.

The steps involved in the testing are:

7.1 UNIT TESTING

This is the first level of testing. In these different modules are tested against the specification produces during the design of the modules. Unit testing is done during the coding phase and to test the internal logic of the modules. It refers to the modules. It refers to the verification of single program module in an isolated environment. Unit testing first focuses on the modules independently of one another to locate errors.

After coding each dialogue is tested and run individually, all necessary coding where removed and it was ensured that all the modules are worked, as the programmer would expect. Logical errors found where corrected. So, by working all the modules independently and verifying the outputs o each module in the presence of staff we conducted that the programs was functioning as expected.

7.2 INTEGRATION TESTING

Data can be lost across an interface, one module can be adverse effort on another, sub functions when combined may not produce the desired major functions. Integration testing is a systematic testing for constructing the program structure. Conducting the tests is to uncover errors associated within the interface. The objective is to take unit tested to modules and build a program structure. All the modules are combined and tested as whole. Here correction is difficult because the vast expense of entire program complicates the isolation causes. Thus, in integration testing steps, all the errors uncovered are corrected for the next testing steps.

In integration testing all the modules that had complete unit test, is combined using menus and hyperlinks and test for whole.in Portfolio Management, all the modules like administrator module and user module are combined.in the first step user module is tested, then user module and administrator module was combined.

7.3 SYSTEM TESTING

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behaviour of a component or a system when it is tested.

System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behaviour of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS).

System testing is the testing to ensure that by putting the software in different environment (e.g. Operating system) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing. System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartially. It has both functional and non-functional testing.

8. SYSTEM IMPLEMENTATION

Implementation is an activity that is contained throughout the development phase it is the process of bringing a developed system into operational use and turning it over to the user. The new system and its components are to be tested in a structured and planned manner. A successfully system should be delivered and users should have the confidence that the system would work efficiently and effectively. The more complex the system being implemented the more involved will be the system analysis and design effort required for implementation. implementation is a stage of the system when the theoretical design is turned into working system. The implementation involves careful planning investigation of the current system and its constraints on implementation design of methods to achieve the changeover method. There are three types of implementation: -

- Implementation of a computer system to replace manual system. The problems involved are converting files, training users creating accurate files and verifying printouts for integrity.
- Implementation of a new computer system to replace existing one. This is usually a difficult conversion. If not properly planned, there can be many problems some large system has taken as long as year to convert.
- Implementation of modified application to replace existing one using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in files.

9. SYSTEM MAINTENANCE

Maintenance covers a wide range of activities including correcting coding and design errors, updating documentation and test data and upgrading user support. After a system is successfully implemented, it should be maintained in proper manner. The need for system maintenance is to make the system adaptable to the changes in the system environment. There may be social, economic and technical changes, which affect system being implemented. through proper system maintenance procedures, the system can be adapted to cope with these changes. Maintenance can be classified as corrective, adaptive, perfective. Corrective implementation means repairing processing or performance failure or making changes because previously uncorrected problem false assumption adaptive maintenance means changing the problem function. Perfective maintenance or modifying the program is to respond to the user's additional or changing needs.

Any system developed should be secure and protected against possible hazards. Security measures are provided to prevent unauthorized access of the database at various levels.an interpreted power supply should be that of power failures or voltage fluctuation will not erase the data in files.

Password protection and simple procedures to prevent the unauthorized access are provided to the user. The system allows entering the system only through proper username and password.

10. FUTURE ENHANCEMENT

The project has a very vast scope in future. In future work, new metrics can be investigated which can be used in automatic evaluation environment to measure - the overall quality such as grammar, readability, prominence and relativeness.

The state of the art summarization systems are all extractive in nature, but the community is gradually progressing towards abstractive summarization. Although a complete abstractive summarization would require deeper natural language understanding and processing a hybrid or shallow abstractive summarization can be achieved through sentence compression and textual entailment techniques. Textual entailment helps in detecting shorter versions of text that entail with same meaning as original text. With textual entailment we can produce more concise and shorter summaries.

The Implemented system in this thesis can work as framework for the research community to understand and extend the applicability of cognitive and symbolic approach in various domains of business needs. Research in summarization continues to enhance the diversity and information richness, and strive to produce coherent and focused answers to users information need.

11. CONCLUSION

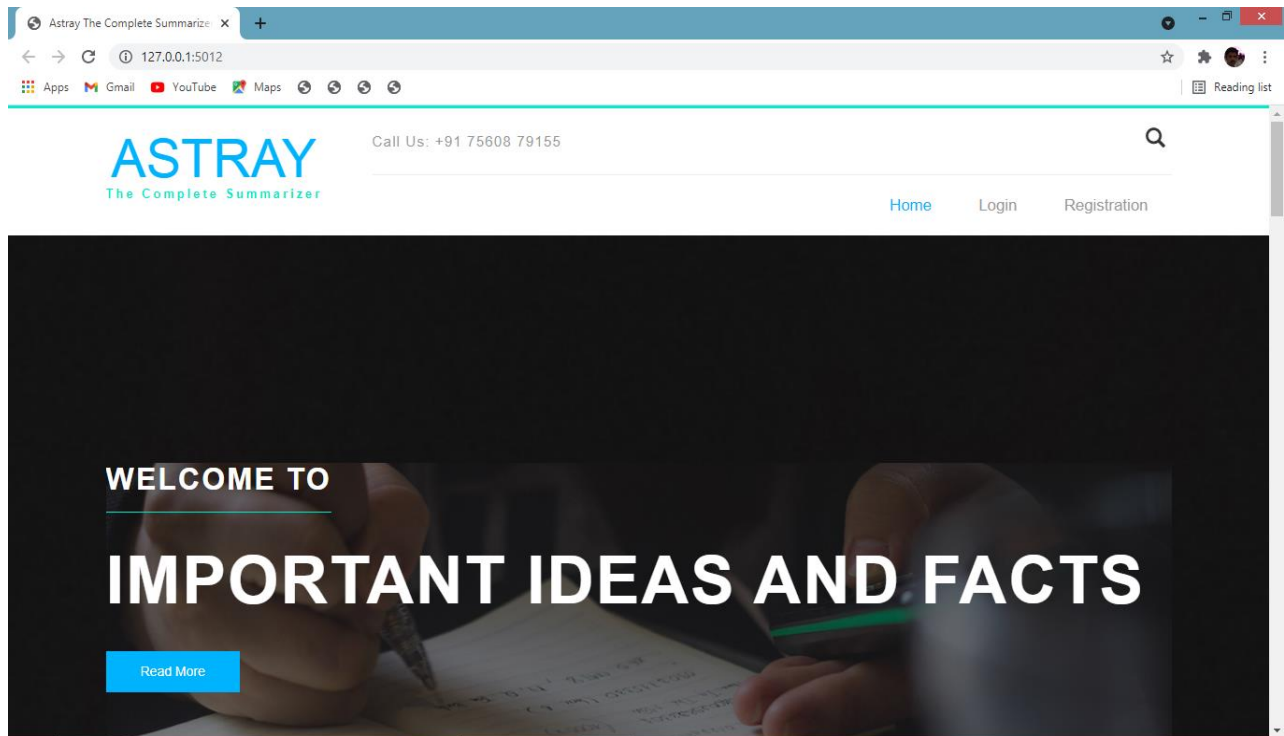
As with time internet is growing at a very fast rate and with it data and information is also increasing. It will going to be difficult for human to summarize large amount of data. Thus there is a need of automatic text summarization because of this huge amount of data. Until now, we have read multiple papers regarding text summarization, natural language processing and lesk algorithms.

There are multiple automatic text summarizers with great capabilities and giving good results. We have learned all the basics of Extractive and Abstractive Method of automatic text summarization and tried to implement extractive one. We have made a basic automatic text summarizer using nltk library using python and it is working on small documents. The system has developed an automatic text summarization system using natural language processing technique i.e. sematic chaining in combination with feature extraction in using fuzzy logic and handling of correlation of sentences.

12. APPENDIX

12.1 Sample Input Screen

12.1.1. Home



12.1.2. Registration

A screenshot of a web browser displaying the registration form for 'Astray The Complete Summarizer'. The browser's address bar shows the URL '127.0.0.1:5012/registration'. The form is overlaid on a background image of hands writing on a notepad. The form fields are as follows:

Field Label	Value
First Name	Alex
Last Name	Zack
Phone Number	9847088225
Email ID	alexza@hotmail.com
Place	Bangalore
Username	alexroz
Password	****

A blue 'SUBMIT' button is located at the bottom of the form.

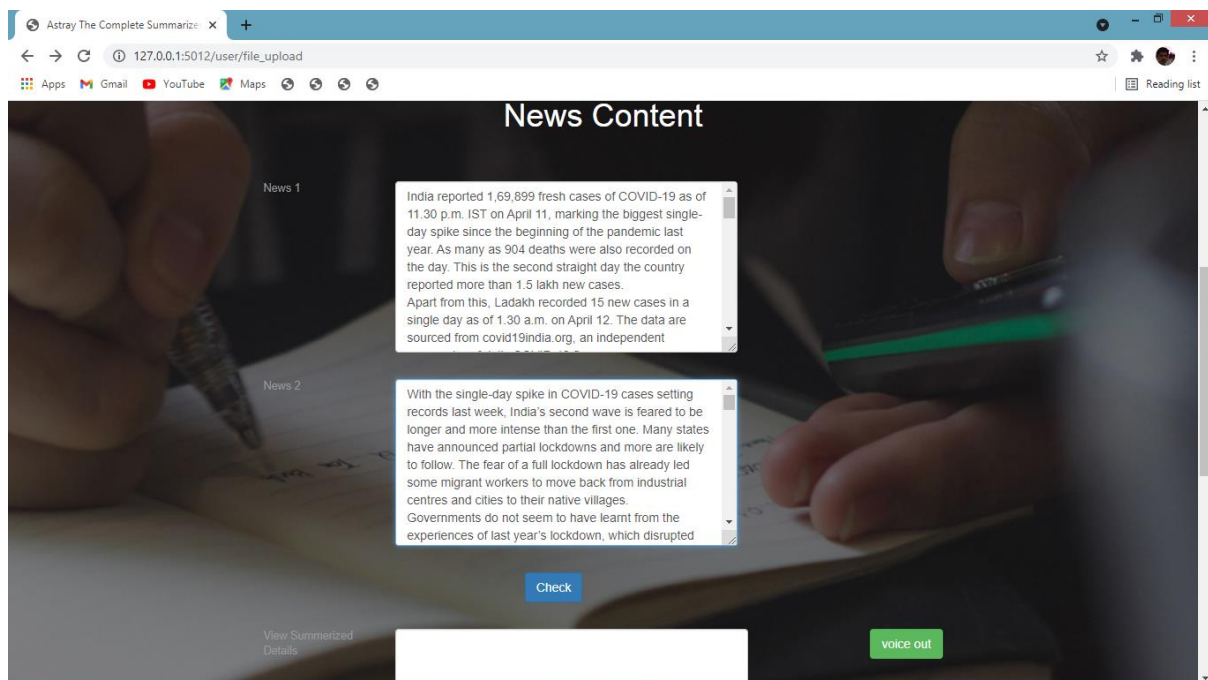
12.1.3. Login (User)

A screenshot of the 'Login (User)' page on the 'Astray The Complete Summarizer' website. The browser's address bar shows the URL '127.0.0.1:5012/login'. The page features the 'ASTRAY The Complete Summarizer' logo and a contact number 'Call Us: +91 75608 79155'. Navigation links for 'Home', 'Login', and 'Registration' are visible. The login form is overlaid on a background image of hands writing on a notepad. The form fields are as follows:

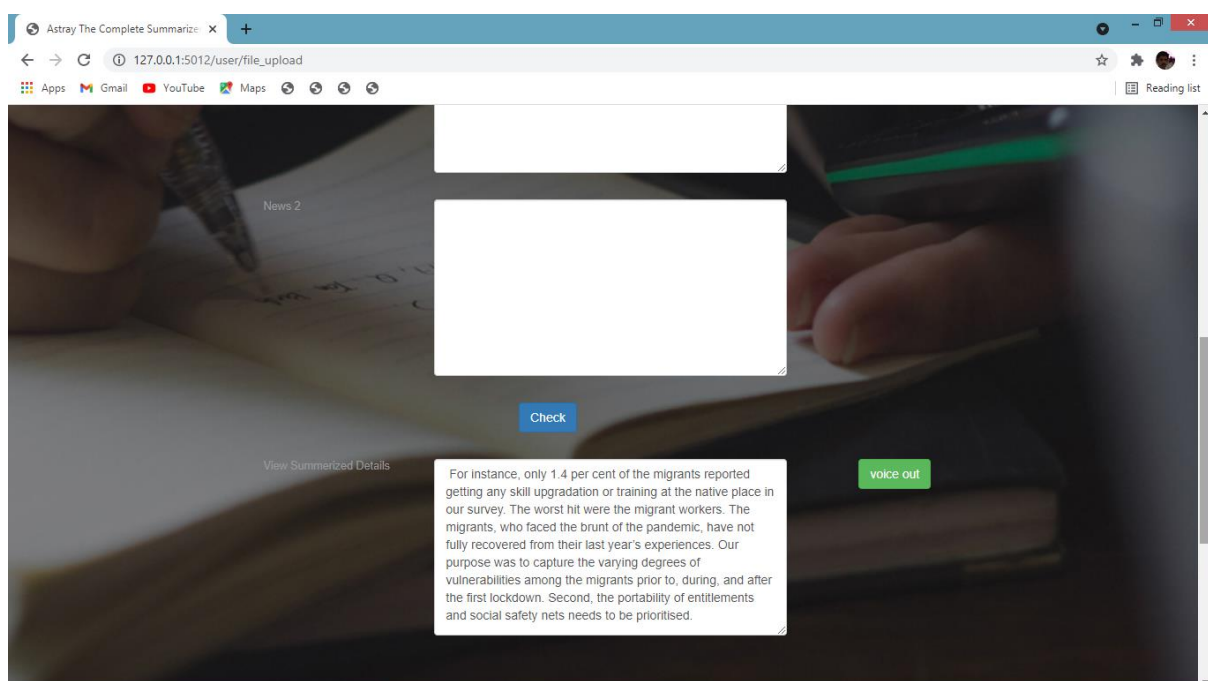
Field Label	Value
Username	user
Password	****

A blue 'LOGIN' button is located at the bottom of the form.

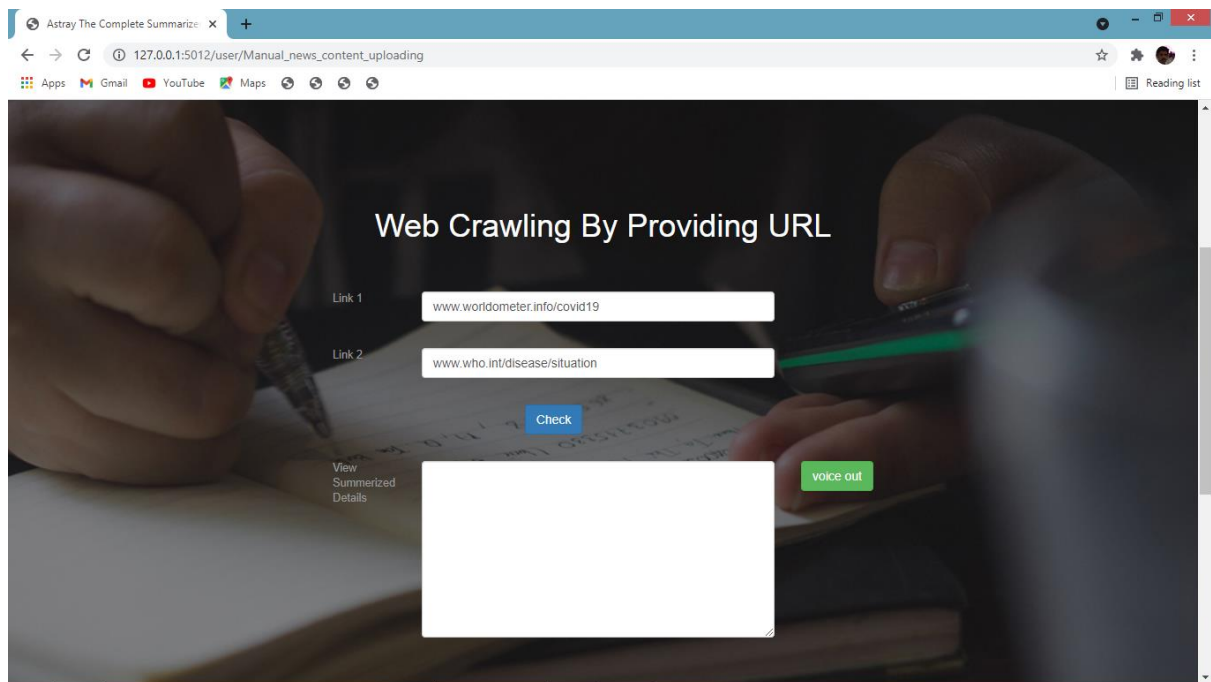
12.1.3.1 Manual Summarization (User)



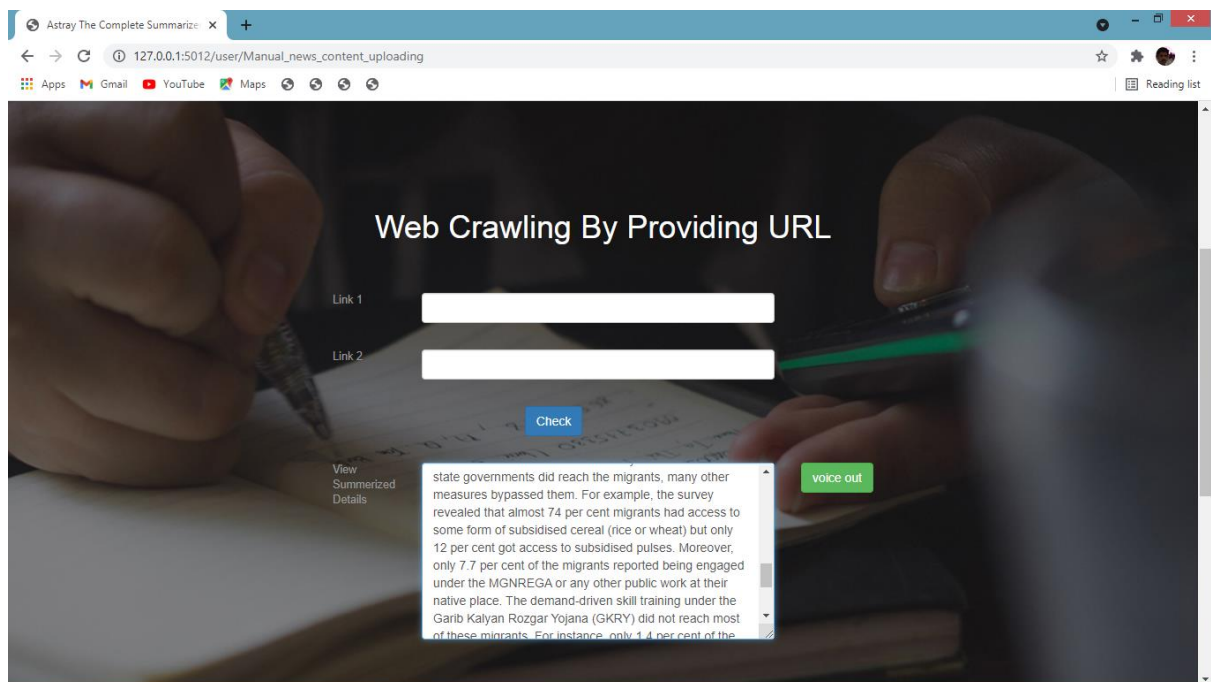
12.1.3.1.1 Manual Summarization (User)



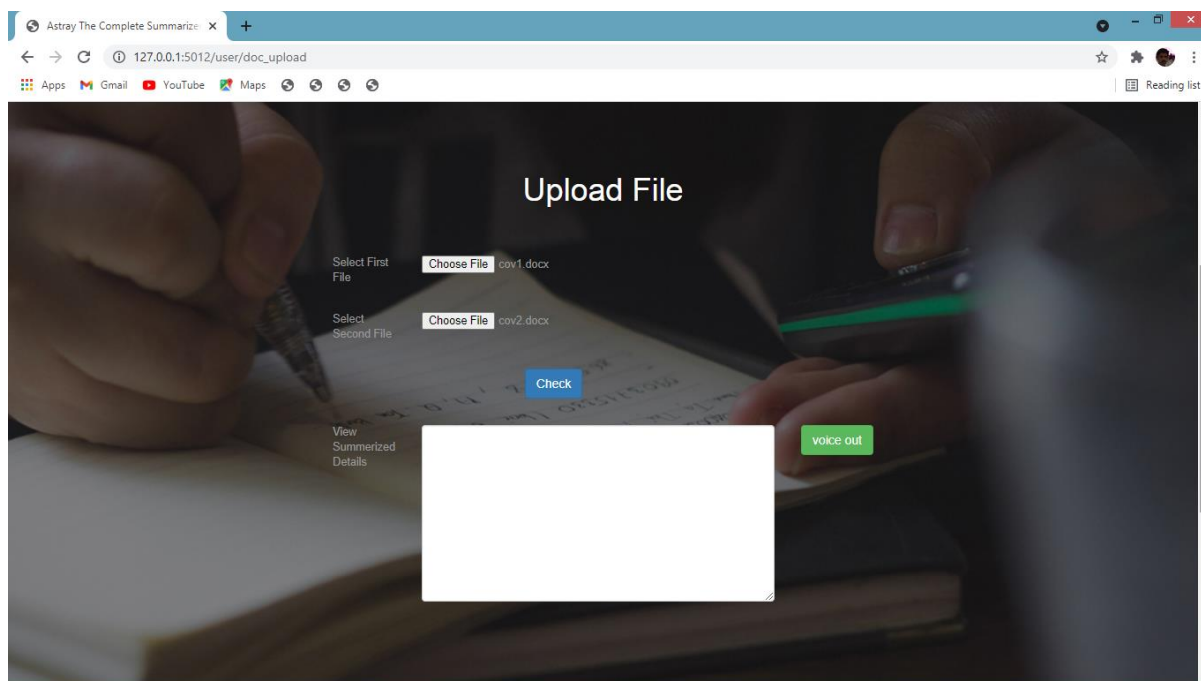
12.1.3.2 Web crawling (User)



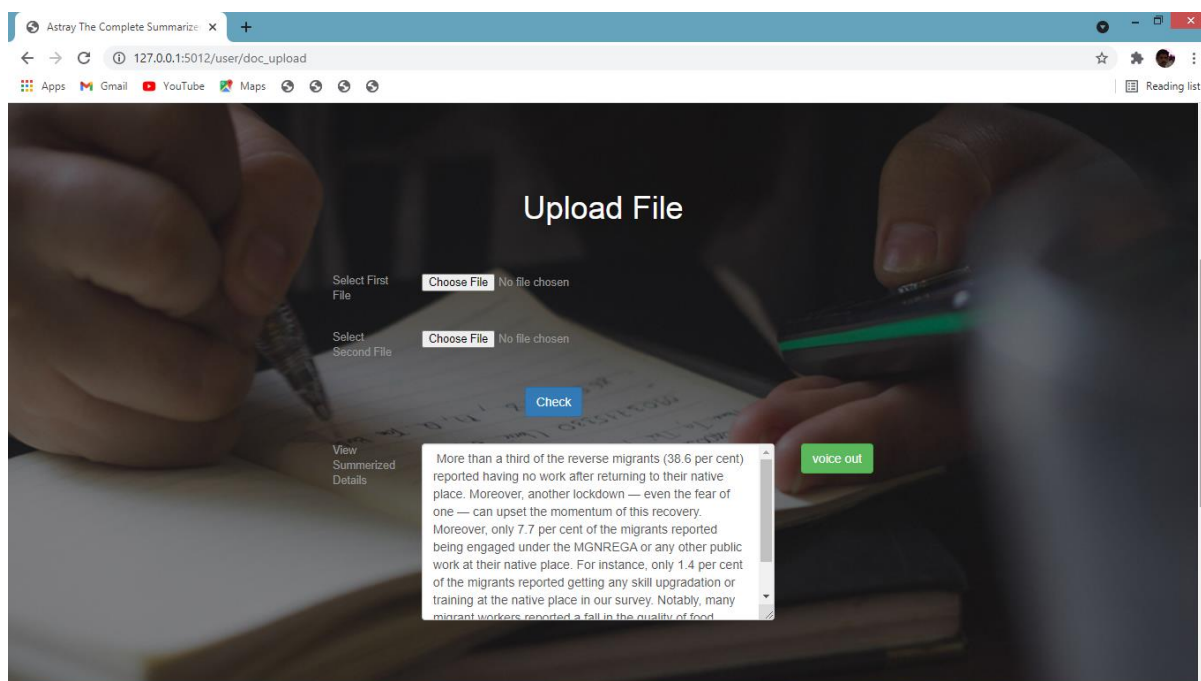
12.1.3.2.1 Web crawling (User)



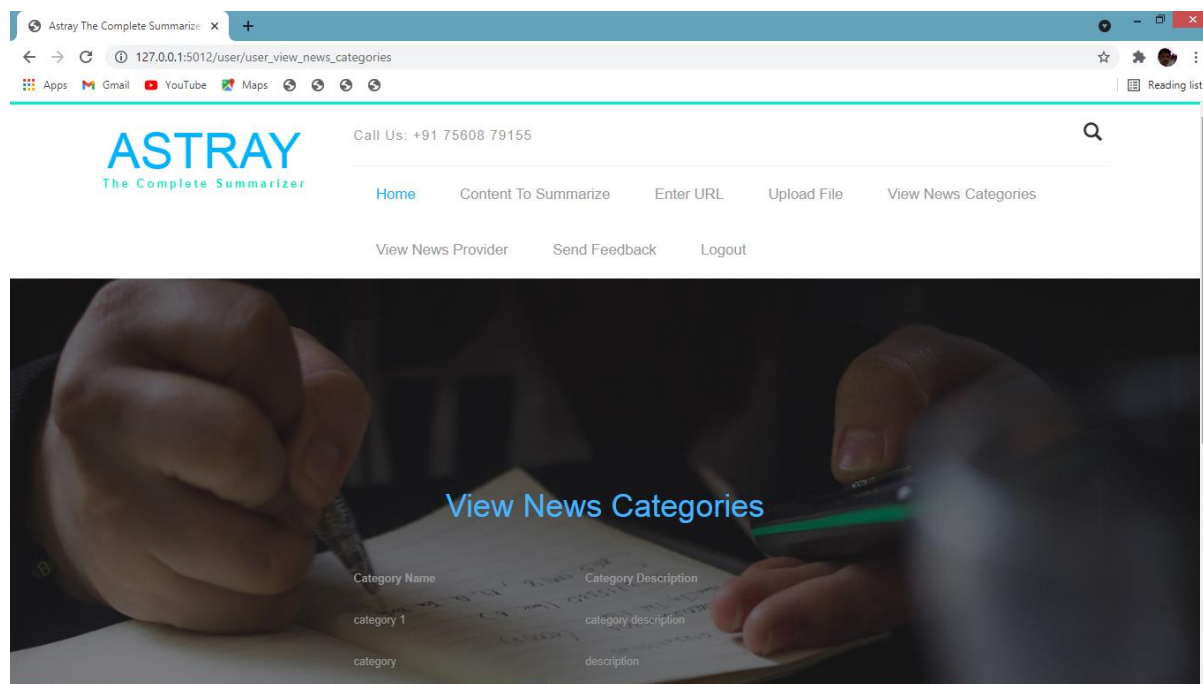
12.1.3.3 File Upload(User)



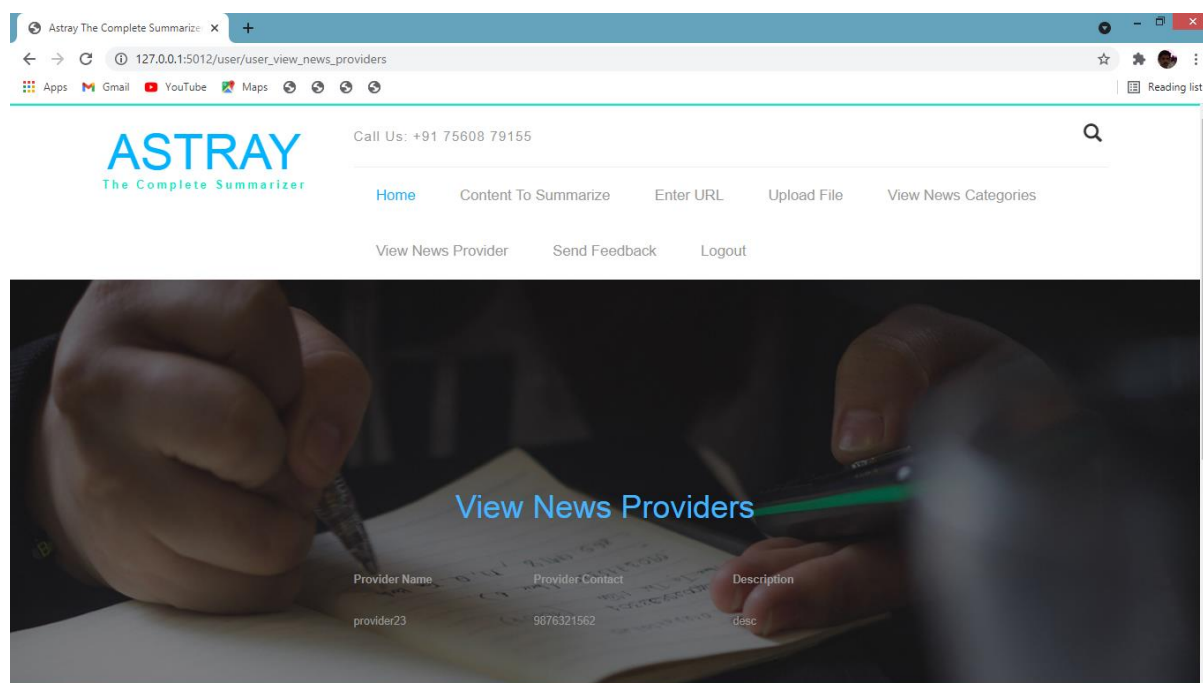
12.1.3.3.1 File Upload(User)



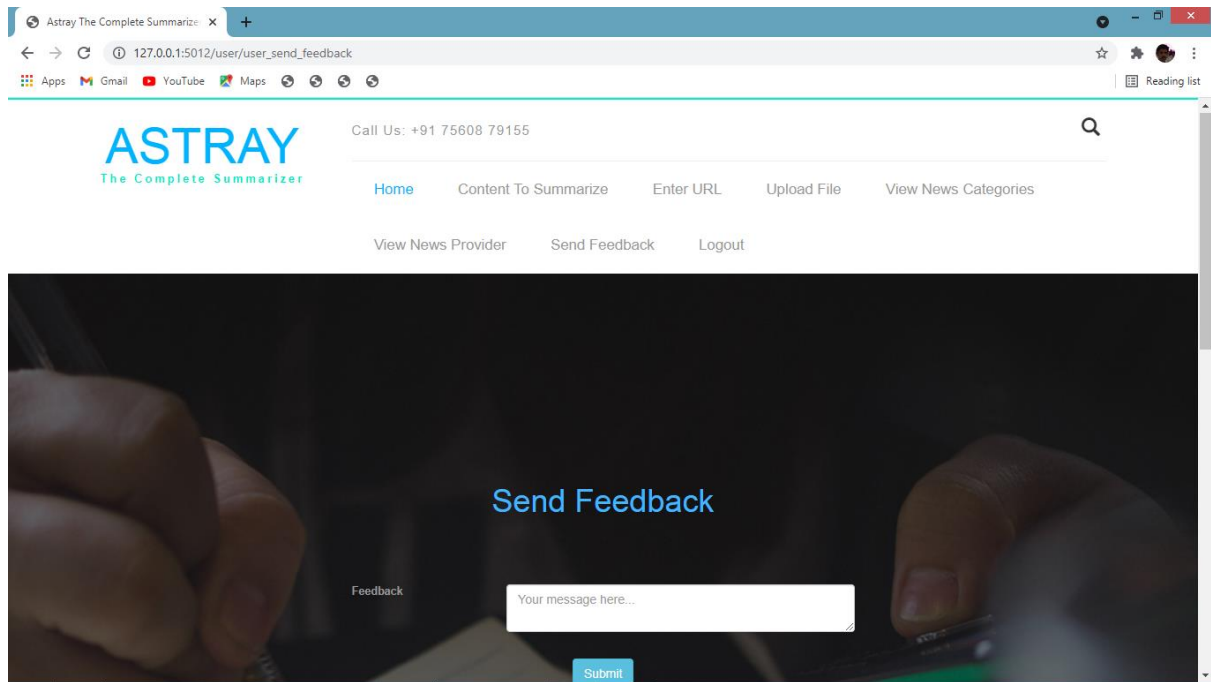
12.1.3.4 View News Categories



12.1.3.5 View News Provider

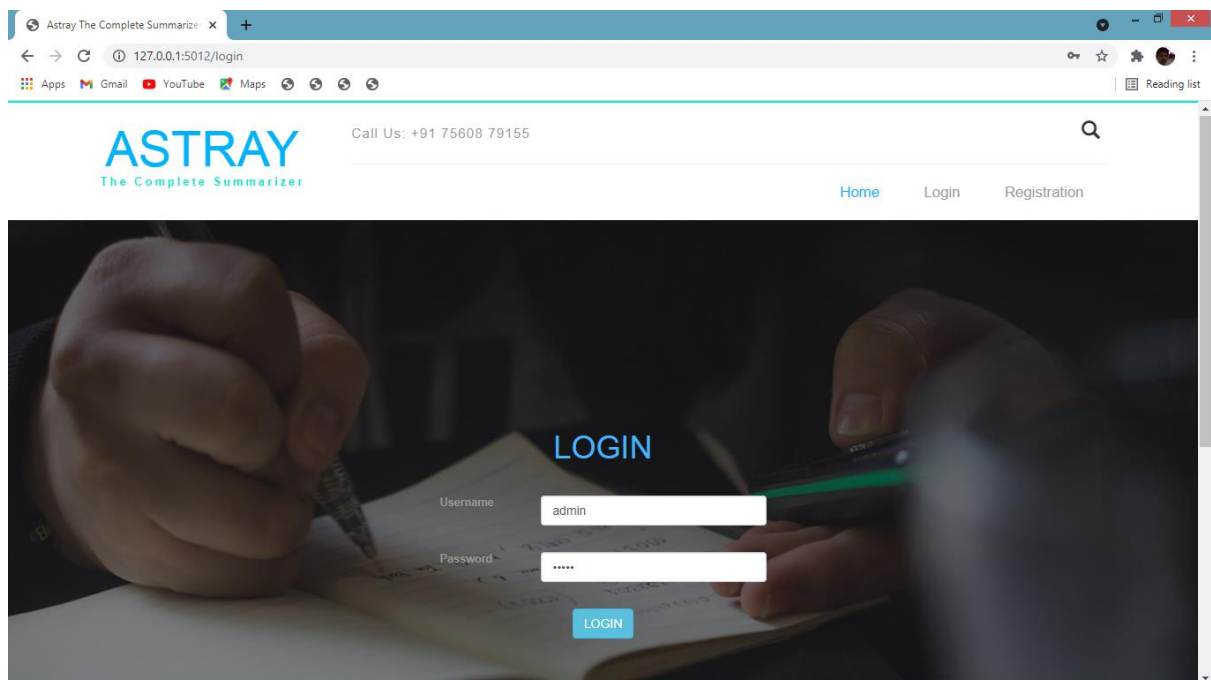


12.1.3.6 Feedback (User)



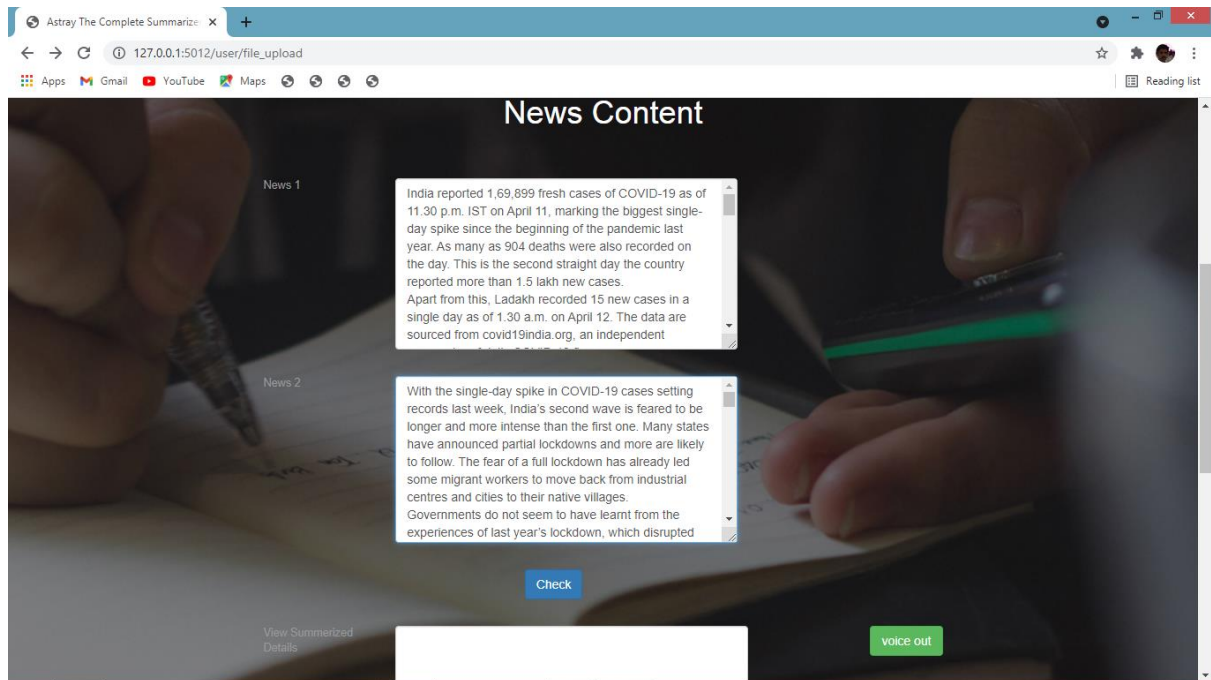
The screenshot shows a web browser window with the URL `127.0.0.1:5012/user/user_send_feedback`. The page header includes the ASTRAY logo, a phone number (+91 75608 79155), and a search icon. The navigation menu contains links for Home, Content To Summarize, Enter URL, Upload File, View News Categories, View News Provider, Send Feedback, and Logout. The main content area features a dark background with a hand writing on a notepad. The text "Send Feedback" is prominently displayed. Below it, there is a text input field labeled "Your message here..." and a blue "Submit" button.

12.1.4. Login (Admin)

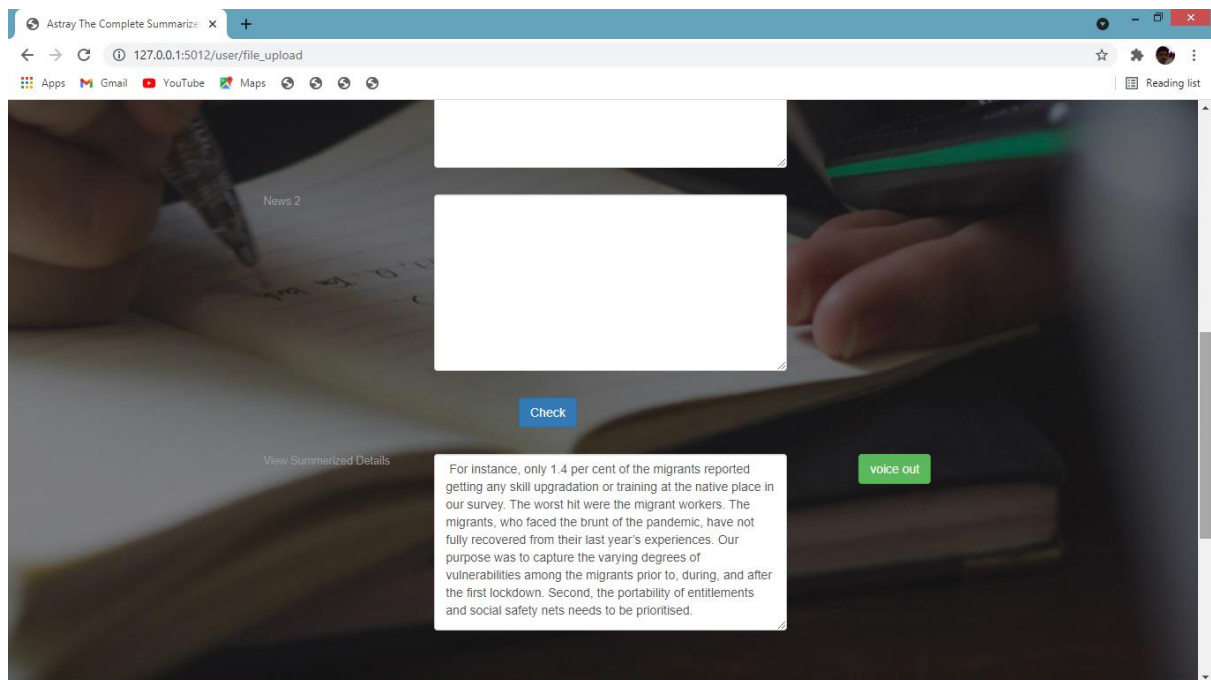


The screenshot shows a web browser window with the URL `127.0.0.1:5012/login`. The page header includes the ASTRAY logo, a phone number (+91 75608 79155), and a search icon. The navigation menu contains links for Home, Login, and Registration. The main content area features a dark background with a hand writing on a notepad. The text "LOGIN" is prominently displayed. Below it, there are two input fields: "Username" with the value "admin" and "Password" with masked characters (*****). A blue "LOGIN" button is positioned below the password field.

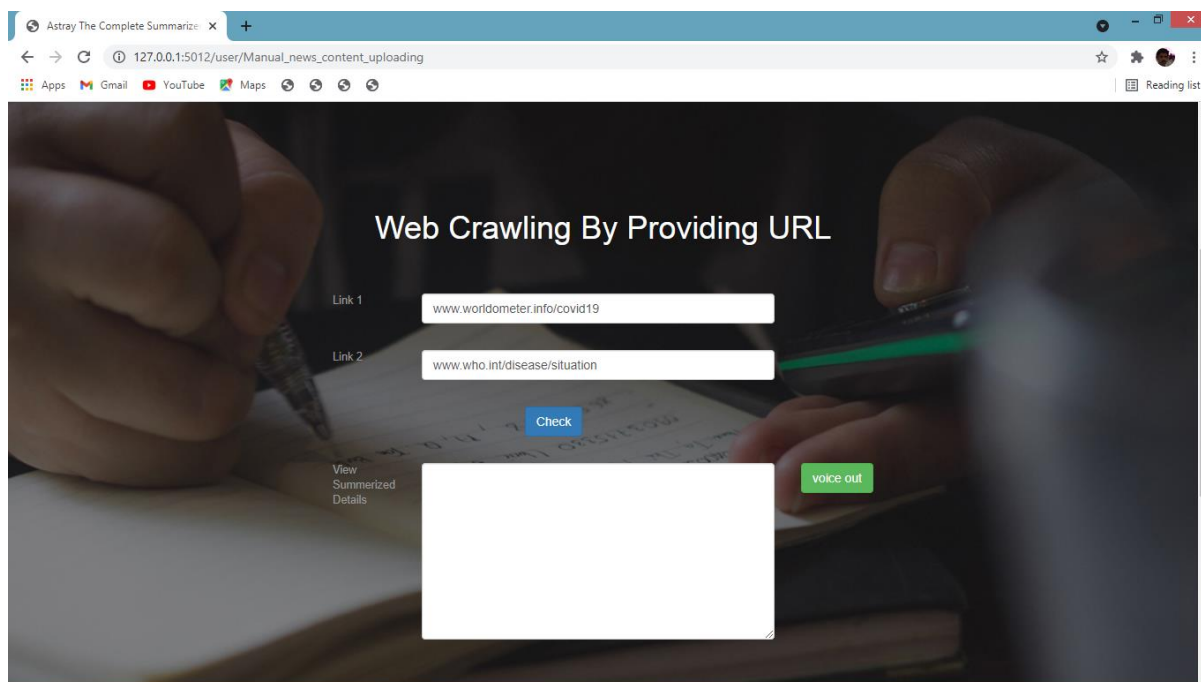
12.1.4.1 Manual Summarization (Admin)



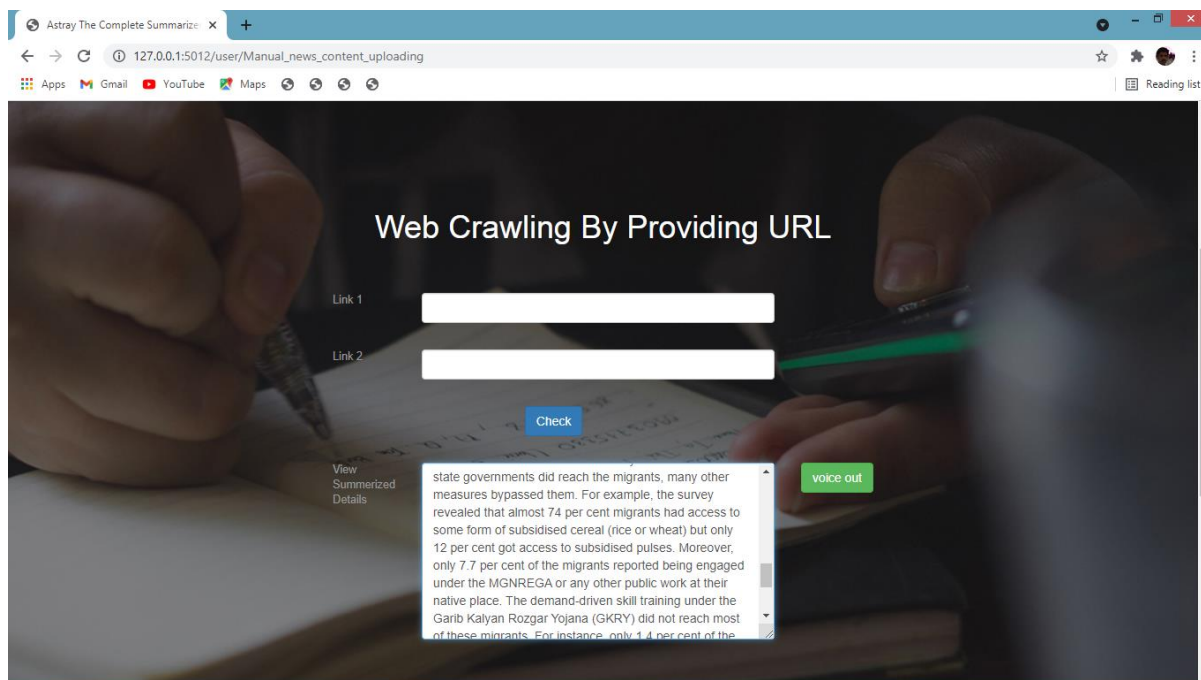
12.1.4.1.1 Manual Summarization (Admin)



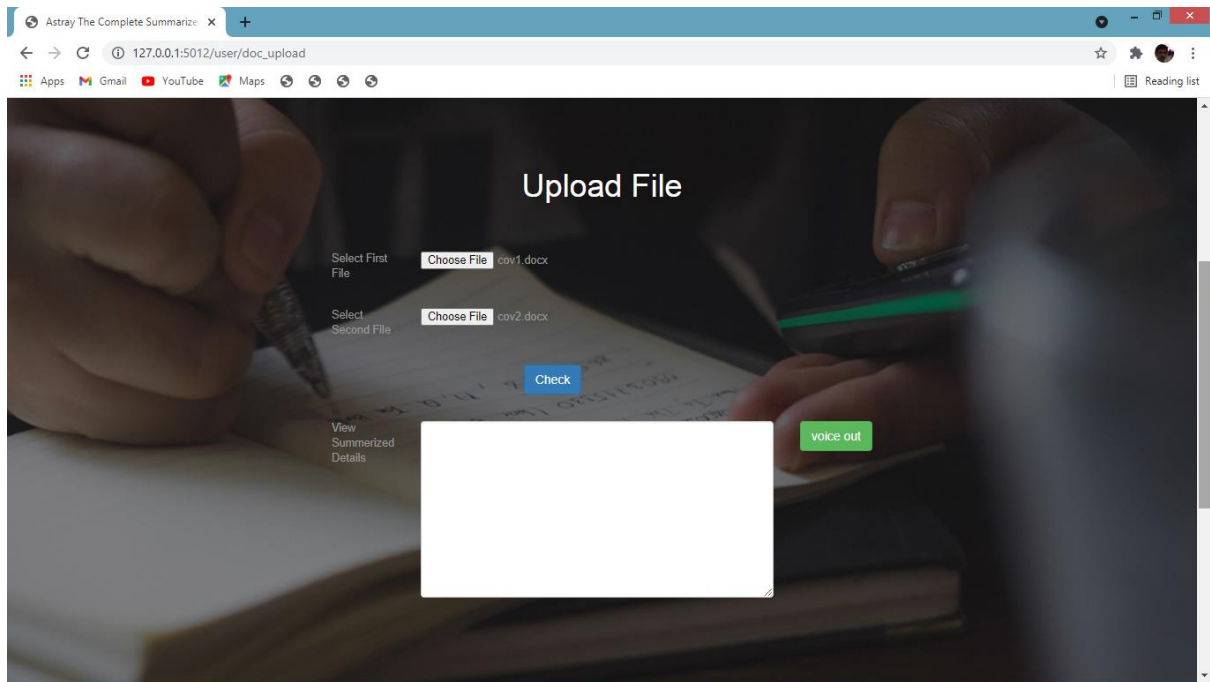
12.1.4.2 Web crawling (Admin)



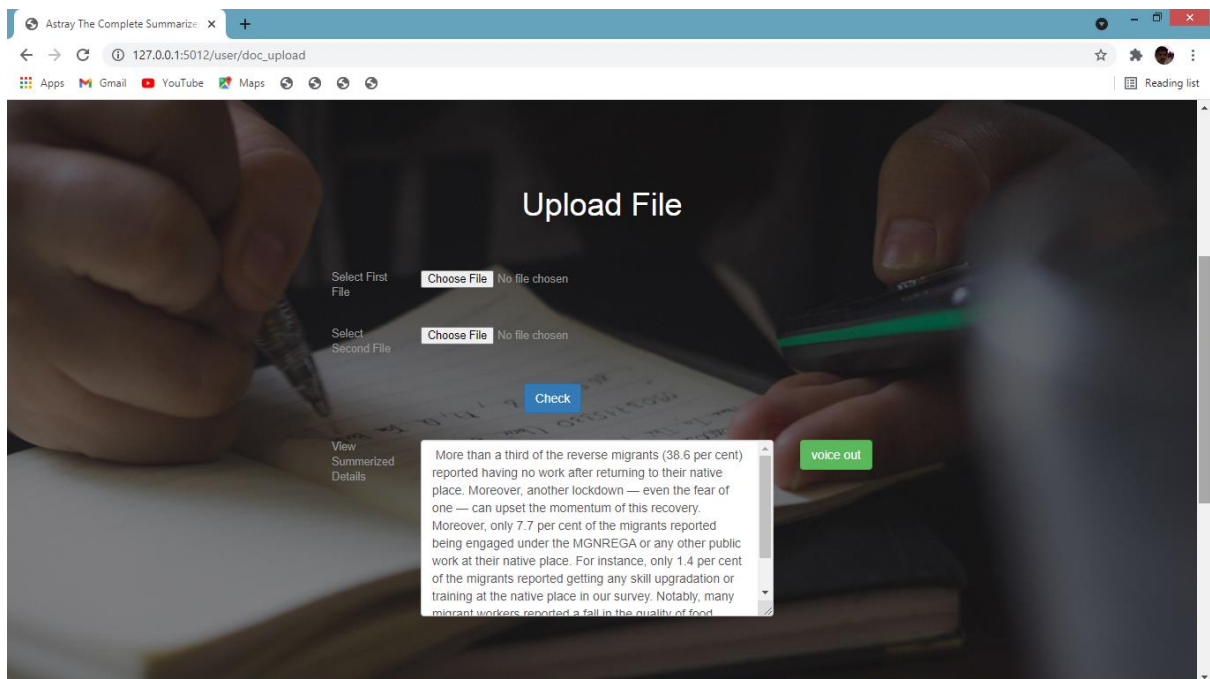
12.1.4.2.1 Web crawling (Admin)



12.1.4.3 File Upload(Admin)



12.1.4.3.1 File Upload(Admin)



12.1.4.4 Manage News Providers

The screenshot shows a web browser window with the URL `127.0.0.1:5012/admin/admin_manage_news_providers`. The page has a dark background with a light blue header. The main content area is titled "Manage News Providers" in blue text. Below the title, there are three input fields: "Provider name", "Provider Contact", and "Description". A blue "REGISTER" button is positioned below the "Description" field. Below the registration form, there is a table titled "News Providers". The table has three columns: "Provider Name", "Provider Contact", and "Description". The first row of data shows "provider23", "9876321562", and "desc". To the right of the "desc" cell, there are two buttons: a red "delete" button and a green "update" button.

Provider Name	Provider Contact	Description
provider23	9876321562	desc

12.1.4.5 Manage News Categories

The screenshot shows a web browser window with the URL `127.0.0.1:5012/admin/admin_manage_news_categories`. The page has a dark background with a light blue header. The main content area is titled "Manage News Categories" in blue text. Below the title, there are two input fields: "Category Name" and "Category Description". A blue "REGISTER" button is positioned below the "Category Description" field. Below the registration form, there is a table titled "News Categories". The table has two columns: "Category Name" and "Category Description". The first row of data shows "category 1" and "category description". To the right of the "category description" cell, there are two buttons: a red "delete" button and a green "update" button.

Category Name	Category Description
category 1	category description

12.1.4.6 Manage News Content

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5012/admin/manage_news_content'. The page features a dark background with a blurred image of hands writing on a notepad. The main heading is 'Manage News Content' in blue. Below it, there is a form with the following fields:

- News Categories:** A dropdown menu with 'select' as the current value.
- Title:** A text input field.
- Description:** A text input field.
- REGISTER:** A blue button.

Below the form, there is a section titled 'News Content' which contains a table with the following data:

Category Name	Title	Date	Description	
category 1	title1	2021-02-12	description_1	<div>delete</div> <div>update</div>

12.1.4.7 Feedback (Admin)

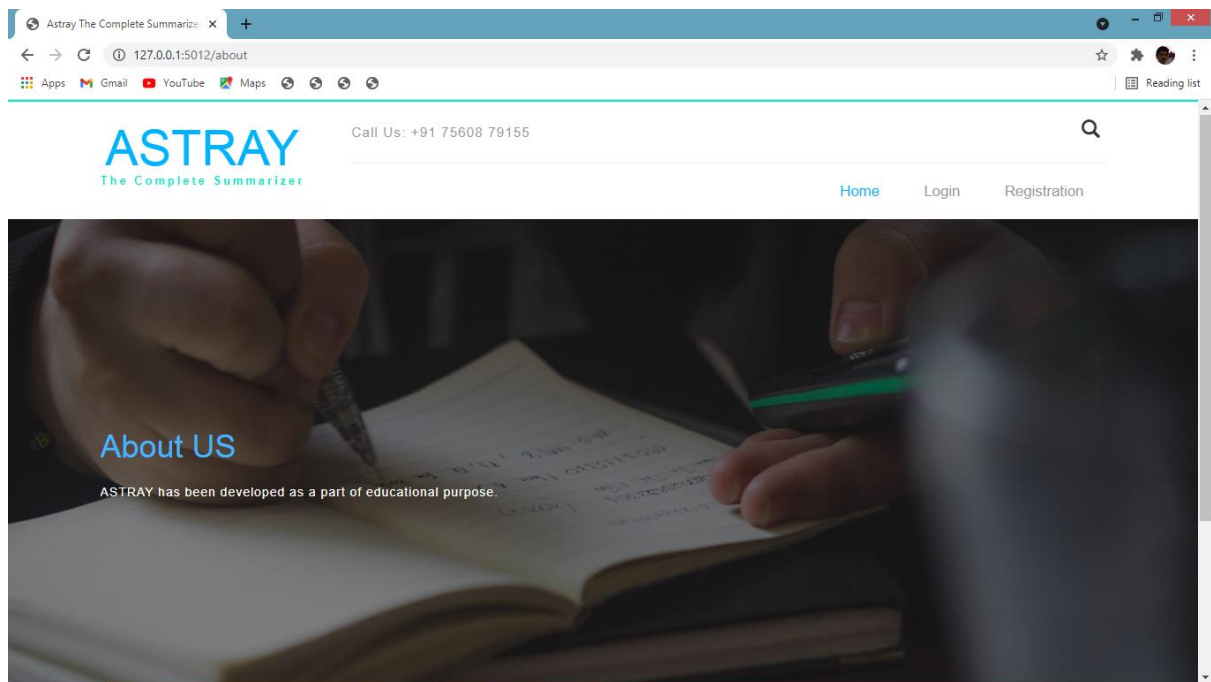
The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5012/user/user_send_feedback'. The page features a dark background with a blurred image of hands writing on a notepad. The main heading is 'Send Feedback' in blue. Below it, there is a form with the following fields:

- Feedback:** A text input field with the placeholder text 'Your message here...'. Below this field is a blue 'Submit' button.

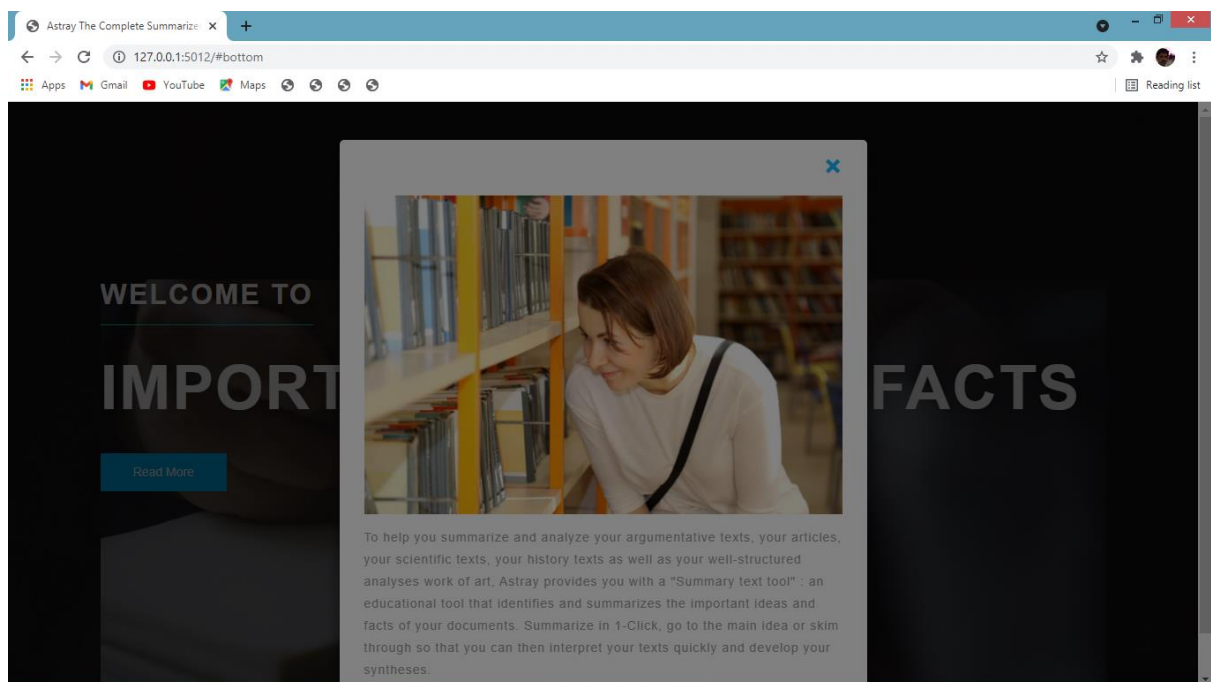
At the top of the page, there is a navigation bar with the following elements:

- ASTRAY The Complete Summarizer** logo.
- Call Us: +91 75608 79155
- Search icon.
- Navigation links: Home, Content To Summarize, Enter URL, Upload File, View News Categories, View News Provider, Send Feedback, Logout.

12.1.5. About



Objective



12.2 PROCESS DESCRIPTION (CODE)

HTML

Index.html

```
{% include 'heder.html' %}

<!-- banner -->
<div class="banner">
    <div class="container">
        <div class="banner-text agileits-w3layouts">
            <div id="top" class="callbacks_container">
                <ul class="rslides" id="slider3">
                    <li>
                        <div class="banner-text agileinfo">
                            <h6>WELCOME TO</h6>
                            <h3>IMPORTANT IDEAS AND
FACTS</h3>
                            <div class="more">
                                <a href="#" data-toggle="modal"
data-target="#myModal"> Read More</a>
                            </div>
                        </div>
                    </li>
                    <li>
                        <div class="banner-text agileinfo">
                            <h6>WELCOME TO</h6>
                            <h3>dolor sit amet </h3>
                            <div class="more">
                                <a href="#" data-
toggle="modal" data-target="#myModal"> Read More</a>
```

```

        </div>
    </div>
</li>
<li>
    <div class="banner-textagileinfo">
        <h6>WELCOME TO</h6>
        <h3>adipiscing elit</h3>

        <div class="more">
            <a href="#" data-
toggle="modal" data-target="#myModal"> Read More</a>
        </div>
    </div>
</li>
</ul>
</div>
</div>
</div>
</div>
<!-- modal-sign -->
<div class="modal bnr-modal fade" id="myModal" tabindex="-1" role="dialog">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-
dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
            </div>
            <div class="modal-body modal-spa">
                
                <p>To help you summarize and analyze your
argumentative texts, your articles, your scientific texts, your history texts as well as your

```

well-structured analyses work of art, Astray provides you with a "Summary text tool" : an educational tool that identifies and summarizes the important ideas and facts of your documents. Summarize in 1-Click, go to the main idea or skim through so that you can then interpret your texts quickly and develop your syntheses. </p>

```

        </div>
    </div>
</div>
<!-- //modal-sign -->
<!-- //banner -->
    <!-- welcome -->
<div class="welcome">
    <div class="container">
        <div class="col-md-6 welcome-w3lright">
            <div class="welcome-grids">
                <div class="service-box">
                    <div class="agileits-wicon">
                        <i class="fa fa-clone" aria-
hidden="true"></i>

                    </div>
                    <h4>WRITERS</h4>
                    <h5>Need to summarize your chapters ? With
Astray, go to the heart of your ideas..</h5>
                </div>
            <div class="clearfix"> </div>
        </div>
    <div class="welcome-grids">
        <div class="col-md-6 col-sm-6 col-xs-6 service-box">
            <div class="agileits-wicon">
                <i class="fa fa-heart-o" aria-
hidden="true"></i>

            </div>
        </div>
    </div>

```

```

        <h4>EDITORS</h4>
        <h5>Identify and understand very fast the facts
and the ideas of your texts that are part of the current news and events.</h5>
    </div>
    <div class="col-md-6 col-sm-6 col-xs-6 service-box">
        <div class="agileits-wicon">
            <i class="fa fa-tv" aria-
hidden="true"></i>
        </div>
        <h4> READERS </h4>
        <h5>Save time, summarize your digital
documents for a relevant and fast uptake of information.</h5>
    </div>
    <div class="clearfix"> </div>
</div>
</div>
<div class="col-md-6 welcome-w3lleft">
    <h3 class="agileits-title">Welcome !</h3>
</div>
<div class="clearfix"> </div>
</div>
</div>
<!-- //welcome -->
<!-- testimonials -->
<div class="testimonials">
    <div class="container">
        <div class="w3ls-title">
            <h3 class="w3ls-title">Testimonials</h3>
        </div>
        <div class="slidw3-agileits">
            <div class="col-md-4 slid-w3text">
                </div>
            </div>
        </div>
    </div>
</div>

```



```

        <div class="col-md-4 slid-w3text">
        </div>
        <div class="col-md-4 slid-w3text">
        </div>
        <div class="clearfix"> </div>
    </div>
</div>
<!-- //testimonials -->
<!-- features -->
<div class="features">
    <div class="container">
        <div class="w3ls-title">
            <h3 class="agileits-title">Features</h3>
        </div>
        <div class="features-agilerow">
            <div class="col-sm-6 features-left">
                <div class="features-w3grid">
                    <div class="col-xs-3 features-w3grid-left">
                        <i class="fa fa-heart-o" aria-
hidden="true"></i>
                    </div>
                    <div class="col-xs-9 features-w3grid-right">
                    </div>
                    <div class="clearfix"> </div>
                </div>
                <div class="features-w3grid features-w3grid-mdl">
                    <div class="col-xs-3 features-w3grid-left">
                        <i class="fa fa-cogs" aria-
hidden="true"></i>
                    </div>
                    <div class="col-xs-9 features-w3grid-right">

```

```

        </div>
        <div class="clearfix"> </div>
    </div>
    <div class="features-w3grid">
        <div class="col-xs-3 features-w3grid-left">
            <i class="fa fa-line-chart" aria-
hidden="true"></i>

        </div>
        <div class="col-xs-9 features-w3grid-right">
        </div>
        <div class="clearfix"> </div>
    </div>
</div>
<div class="col-sm-6 features-left">
    <div class="features-w3grid">
        <div class="col-xs-3 features-w3grid-left">
            <i class="fa fa-cogs" aria-
hidden="true"></i>

        </div>

        <div class="col-xs-9 features-w3grid-right">
        </div>
        <div class="clearfix"> </div>
    </div>
    <div class="features-w3grid features-w3grid-mdl">
        <div class="col-xs-3 features-w3grid-left">
            <i class="fa fa-line-chart" aria-
hidden="true"></i>

        </div>
        <div class="col-xs-9 features-w3grid-right">
        </div>
        <div class="clearfix"> </div>

```

```

        </div>
        <div class="features-w3grid">
            <div class="col-xs-3 features-w3grid-left">
                <i class="fa fa-heart-o" aria-
hidden="true"></i>
            </div>
            <div class="col-xs-9 features-w3grid-right">
                <div class="clearfix"> </div>
            </div>
        </div>
        <div class="clearfix"> </div>
    </div>
</div>
<!-- //features -->
<!-- work start here -->
    <div class="work jarallax">
        <div class="container">
            <div class="work-agileinfo">
                <h3>Are You Impressed By Our Work?</h3>
                <p>Nor again is there anyone who loves or pursues or desires
to obtain pain.</p>
                <div class="w3btns-agile">
                    <a href="contact.html">Get In Touch</a>
                </div> </div>
            </div>
            <!-- work end here -->
            <!-- news -->
            <div class="news">
                <div class="container">
                    <div class="agileits_w3layouts_news_grids">
                        <div class="col-md-3 col-xs-6 agileits_w3layouts_news_grid">
                            <div class="w3_agileits_news_grid">

```

```

class="img-responsive" />

    <h4>10 Aug 2017</h4>
</div>
<div class="col-md-3 col-xs-6 agileits_w3layouts_news_grid">
    <div class="w3_agileits_news_grid">
        
        <div class="w3_agileits_news_grid_pos">
            <h4>15 Aug 2017</h4>
        </div>
    </div>
<div class="col-md-3 col-xs-6 agileits_w3layouts_news_grid">
    <div class="w3_agileits_news_grid">
        
        <div class="w3_agileits_news_grid_pos">
            <h4>20 Sep 2017</h4>
        </div>
    </div>
</div>
<div class="col-md-3 col-xs-6 agileits_w3layouts_news_grid">
    <div class="w3_agileits_news_grid">
        
        <div class="w3_agileits_news_grid_pos">
            <h4>14 Oct 2017</h4>
        </div>
    </div>
<div class="clearfix"> </div>
</div> </div>
<!-- //news -->
{% include 'footer.html' %}

```

registration.html

```
{% include ['heder.html']%}

<center>
<h1 style="color: #45b6fe">REGISTRATION FORM</h1>
<br>
<form method="POST">
<table class="table" style="width: 400px">
  <tr>
    <th>First Name</th>
    <td><input type="text" required class="form-control"
name="fname" pattern="[a-zA-Z]{0,10}"></td>
  </tr>
  <tr>
    <th>Last Name</th>
    <td><input type="text" required class="form-control"
name="lname" pattern="[a-zA-Z]{0,10}"></td>
  </tr>
  <tr>
    <th align="left" >Phone Number</th>
    <td><input type="text" required class="form-control"
name="phno" pattern="[0-9]{10}"></td>
  </tr>
  <tr>
    <th align="left" >Email ID</th>
    <td><input type="Email" required class="form-control"
name="mail"></td>
  </tr>
  <tr>
    <th>Place</th>
```

```

        <td><input type="text" required class="form-control"
name="place"></td>
    </tr>
    <tr>
        <th>Username</th>
        <td><input type="text" required class="form-control"
name="uname"></td>
    </tr>
    <tr>
        <th>Password</th>
        <td><input type="password" required class="form-control"
name="passw"></td>
    </tr>
    <tr>
        <td align="center" colspan="2"><input type="submit"
class="btn btn-info" name="submit" value="SUBMIT "></td>
    </tr>
</table>
</form>
</center>

```

```
{% include 'footer.html' %}
```

login.html

```
{% include ['heder.html'] %}

<center>

    <form method="post">

        <h1 style="color: #45b6fe">LOGIN </h1>
    <br>

    <table class="table" style="width: 400px">
        <tr>
            <th>Username</th>
            <td><input type="text" required class="form-control"
name="uname"></td>
        </tr>
        <tr>
            <th>Password</th>
            <td><input type="password" required class="form-control"
name="passw"></td>
        </tr>
        <tr>
            <td align="center" colspan="2"><input type="submit" class="btn btn-
info" name="login" value="LOGIN"></td>
        </tr>
    </table><br>
    </form>
</center>

{% include 'footer.html' %}
```

file_upload.html (Manual)

```
{% include ['userheader.html'] %}

<center>
<br><br>
<h1> News Content</h1><br><br>
<form method="post">
<table class="table" style="width: 800px">
<tr>
    <td>News 1</td>
    <td><textarea style="width:400px; height:200px;" name="news1" class="form-
control"></textarea></td>
</tr> <tr>
    <td>News 2</td>
    <td><textarea style="width:400px; height:200px;" name="news2" class="form-
control"></textarea></td>
</tr> <tr>
    <td colspan="2" align="center"><input type="submit" class="btn btn-primary"
name="news" value="Check"></td>
</tr> <tr>
    <td>View Summerized Details</td>
    <td><textarea style="width:400px; height:200px;" name="summerize" class="form-
control">{{data['details']}}</textarea></td>
    <td><input type="submit" name="submit" value="voice out" class="btn btn-
success"></td>
</tr>
</table>
</form>
</center>
{% include 'footer.html' %}
```


doc_upload.html (Document)

```

{% include ['userheader.html'] %}

<center>

<br><br>

<h1> Upload File </h1><br><br>

<form method="post" enctype="multipart/form-data">

<table class="table" style="width: 500px;">

<tr>

        <td>Select First File</td>

        <td><input type="file" name="File1" ></td>

</tr>

<tr>

        <td>Select Second File</td>

        <td><input type="file" name="File2" ></td>

</tr>

<tr>

        <td colspan="2" align="center"><input type="submit" class="btn btn-primary"

name="upld" value="Check"></td>

</tr>

<tr>

        <td>View Summerized Details</td>

        <td><textarea style="width:400px; height:200px;" name="summerize" class="form-

control">{{data['details']}}</textarea></td>

        <td><input type="submit" name="submit" value="voice out" class="btn btn-

success"></td>

</tr></table>

</form>

</center>

<br><br>

{% include 'footer.html' %}

```

manual_news_content_uploading.html (URL)

```
{% include ['userheader.html'] %}

<center>

<br><br>

<h1> Web Crawling By Providing URL</h1><br><br>

<form method="post">

<table class="table" style="width: 500px;">

<tr>

    <td>Link 1</td>

    <td><input type="text" class="form-control" name="link1"></td>

</tr>

<tr>

    <td>Link 2</td>

    <td><input type="text" class="form-control" name="link2"></td>

</tr>

<tr>

    <td colspan="2" align="center"><input type="submit" class="btn btn-primary"

name="link" value="Check"></td>

</tr><tr>

    <td>View Summerized Details</td>

    <td><textarea style="width:400px; height:200px;" name="summerize" class="form-

control">{{data['details']}}</textarea></td>

    <td><input type="submit" name="submit" value="voice out" class="btn btn-

success"></td>    </tr>

</table></form>

</center>

<br><br>

{% include 'footer.html' %}
```

PYTHON

main.py

```
from flask import Flask
from public import public
from admin import admin
from user import user

app=Flask(__name__)
app.secret_key="abc"
app.register_blueprint(public)
app.register_blueprint(admin,url_prefix='/admin')
app.register_blueprint(user,url_prefix='/user')

app.run(debug=True,port=5012)
```

admin.py

```

from flask import Blueprint,render_template,request,redirect,url_for,session
from database import *
import uuid
from summerize import *
from functions import *
import pyttsx3
admin=Blueprint('admin',__name__)
@admin.route('/adminhome')
def adminhome():
    return render_template('adminhome.html')
@admin.route('/Manual_news_content_uploading',methods=['get','post'])
def Manual_news_content_uploading():
    data={ }
    if 'link' in request.form:
        content1=request.form['link1']
        content2=request.form['link2']
        if content2=="":
            val=trade_spider(content1)
            print("checked",val)
            #vals = trade_spiders(content2)
            # print("checkedval", vals)
            #valmain=val+vals
            result = run_summarization(val)
            print(result)
            data['details']=result
            session['result']=result
        else:
            print("sfgkj")
            val=trade_spider(content1)
            print("checked",val)

```

```

        vals = trade_spiders(content2)
    # print("checkedval", vals)
    valmain=val+vals
    result = run_summarization(valmain)
    print(result)
    data['details']=result
    session['result']=result
if 'submit' in request.form:
    speech(session['result'])
return render_template('admin_manual_file_upload.html',data=data)
@admin.route('/doc_upload',methods=['get','post'])
def doc_upload():
    data={ }
    data={ }
if 'upld' in request.form:
    file1=request.files['File1']
    file2=request.files['File2']
    path1='static/files/'+str(uuid.uuid4())+str(file1.filename)
    file1.save(path1)
    splitpath=path1.split('.')
    types=splitpath[1]
    print(types,'.....')
    if types=='txt':
        value=txtreader(path1)
    if types=='docx':
        value=docreader(path1)
    if types=='pdf':
        value=pdfreader(path1)
    # upfl=trade_spider(file1)
    path2='static/files/'+str(uuid.uuid4())+str(file2.filename)
    file2.save(path2)
    splitpath2=path2.split('.')
    types1=splitpath2[1]

```

```

print(types1)
if types=='txt':
    value1=txtreader(path2)
if types=='docx':
    value1=docreader(path2)
if types=='pdf':
    value1=pdfreader(path2)
print(value1)
vals=value+value1
print(".....")
result = run_summarization(vals)
print(result)
print("////////////////////////////////////")
# speech(result)
data['details']=result
session['result']=result
if 'submit' in request.form:
    speech(session['result'])
    data['s']=session['result']
return render_template('doc_upload.html',data=data)
@admin.route('/file_upload',methods=['get','post'])
def file_upload():
    data={ }
    if 'news' in request.form:
        news1=request.form['news1']
        news2=request.form['news2']
        if 'news2'=="":
            valmain = news1
            result = run_summarization(valmain)
            print(result)
            data['details']=result
            session['result']=result
        else:

```

```

        valmain = news1 + news2
        result = run_summarization(valmain)
        print(result)
        data['details']=result
        session['result']=result
    if 'submit' in request.form:
        speech(session['result'])
        data['s']=session['result']
    return render_template('admin_file_upload.html',data=data)
@admin.route('/admin_manage_news_providers',methods=['get','post'])
def admin_manage_news_providers():
    data={ }
    if 'submit' in request.form:
        pname=request.form['pname']
        pcontact=request.form['pcontact']
        description=request.form['description']
        q="INSERT INTO
`news_providers`(`provider_name`,`provider_contact`,`Description`) VALUES
('%s','%s','%s')"%(pname,pcontact,description)
        insert(q)
        return redirect(url_for('admin.admin_manage_news_providers'))
    q="SELECT * FROM `news_providers` "
    select(q)
    res=select(q)
    data['pro']=res
    if 'action' in request.args:
        action=request.args['action']
        proid=request.args['id']
    else:
        action=None
    if action=='delete':
        q="DELETE FROM `news_providers` WHERE `provider_id`='%s'"%(proid)
        delete(q)

```

```

        return redirect(url_for('admin.admin_manage_news_providers'))
    if action=='update':
        q="SELECT * FROM `news_providers` WHERE `provider_id`='%s'"%(proid)
        res5=select(q)
        data['uppro']=res5
    if 'update' in request.form:
        pname=request.form['pname']
        pcontact=request.form['pcontact']
        description=request.form['description']
        q="update `news_providers` set
`provider_name`='%s',`provider_contact`='%s',`description`='%s' WHERE
`provider_id`='%s'"%(pname,pcontact,description,proid)
        update(q)
        return redirect(url_for('admin.admin_manage_news_providers'))
    return render_template('admin_manage_news_providers.html',data=data)
@admin.route('/admin_manage_news_categories',methods=['get','post'])
def admin_manage_news_categories():
    data={}
    if 'submit' in request.form:
        cname=request.form['cname']
        categorydescription=request.form['categorydescription']
        q="INSERT INTO`news_categories`(`category_name`,`category_description`)
VALUES('%s','%s')"%(cname,categorydescription)
        insert(q)
        return redirect(url_for('admin.admin_manage_news_categories'))
    q="SELECT * FROM `news_categories`"
    select(q)
    res=select(q)
    data['category']=res
    if 'action' in request.args:
        action=request.args['action']
        catid=request.args['id']
    else:

```



```

        action=None
    if action=='delete':
        q="DELETE FROM `news_categories` WHERE `category_id`='%s'"%(catid)
        delete(q)
        return redirect(url_for('admin.admin_manage_news_categories'))
    if action=='update':
        q="SELECT * FROM `news_categories` WHERE `category_id`='%s'"%(catid)
        res1=select(q)
        data['up']=res1
    if 'update' in request.form:
        cname=request.form['cname']
        categorydescription=request.form['categorydescription']
        catid=request.args['id']
        q="UPDATE `news_categories` SET
`category_name`='%s',`category_description`='%s' WHERE
`category_id`='%s'"%(cname,categorydescription,catid)
        update(q)
        return redirect(url_for('admin.admin_manage_news_categories'))
    return render_template('admin_manage_news_categories.html',data=data)
@admin.route('/admin_manage_news_content',methods=['get','post'])
def admin_manage_news_content():
    data={}
    q="SELECT * FROM `news_categories`"
    res=select(q)
    data['cat']=res
    if 'submit' in request.form:
        cid=request.form['cname']
        title=request.form['title']
        desc=request.form['desc']
        q="INSERT INTO `news_content`(category_id`,`Title`,`Date`,`Description`)
VALUES('%s','%s',curdate(),'%s')"%(cid,title,desc)
        insert(q)
        return redirect(url_for('admin.admin_manage_news_content'))

```

```

q="SELECT * FROM `news_categories` INNER JOIN `news_content`
USING(`category_id`)"
select(q)
res2=select(q)
data['content']=res2
if 'action' in request.args:
    action=request.args['action']
    contid=request.args['id']
else:
    action=None
if action=='delete':
    q="DELETE FROM `news_content` WHERE `news_id`='%s'"%(contid)
    delete(q)
    return redirect(url_for('admin.admin_manage_news_content'))
if action=='update':
    q="SELECT * FROM `news_content` INNER JOIN `news_categories` USING
(`category_id`) WHERE `news_id`='%s'"%(contid)
    res1=select(q)
    data['upnews']=res1
if 'update' in request.form:
    cname=request.form['cname']
    Title=request.form['title']
    descr=request.form['desc']
    q="UPDATE `news_content` SET
`category_id`='%s',`Title`='%s',`Description`='%s' WHERE
news_id='%s'"%(cname,Title,descr,contid)
    update(q)
    return redirect(url_for('admin.admin_manage_news_content'))
return render_template('admin_manage_news_content.html',data=data)
def speech(text):
    print("sg")
    engine = pyttsx3.init() # object creation
    """ RATE"""

```

```

rate = engine.getProperty('rate') # getting details of current speaking rate
print (rate)                      #printing current voice rate
engine.setProperty('rate', 125)   # setting up new voice rate
""""VOLUME""""

volume = engine.getProperty('volume') #getting to know current volume level (min=0 and
max=1)
print (volume)                    #printing current volume level
engine.setProperty('volume',1.0)  # setting up volume level between 0 and 1
""""VOICE""""

voices = engine.getProperty('voices') #getting details of current voice
#engine.setProperty('voice', voices[0].id) #changing index, changes voices. 0 for male
engine.setProperty('voice', voices[1].id) #changing index, changes voices. 1 for female
engine.say(text)
# engine.say('My current speaking rate is ' + str(rate))
engine.runAndWait()
engine.stop()

```

user.py

```

from flask import *
from database import *
import uuid
from summerize import *
from functions import *
import pyttsx3
user=Blueprint('user',__name__)
@user.route('/userhome')
def userhome():
    return render_template('userhome.html')
@user.route('/Manual_news_content_uploading',methods=['get','post'])
def Manual_news_content_uploading():
    data={}
    if 'link' in request.form:
        content1=request.form['link1']
        content2=request.form['link2']
        if content2=="":
            val=trade_spider(content1)
            print("checked",val)
            #vals = trade_spiders(content2)
            # print("checkedval", vals)
            #valmain=val+vals
            result = run_summarization(val)
            print(result)
            # speech(result)

```

```

        data['details']=result
        session['result']=result
    else:
        print("sfgkj")
        val=trade_spider(content1)
        print("checked",val)
        vals = trade_spiders(content2)
        # print("checkedval", vals)
        valmain=val+vals
        result = run_summarization(valmain)
        print(result)
        # speech(result)
        data['details']=result
        session['result']=result

    if 'submit' in request.form:
        speech(session['result'])

    return render_template('Manual_news_content_uploading.html',data=data)

@user.route('/doc_upload',methods=['get','post'])
def doc_upload():
    data={}

    if 'upld' in request.form:
        file1=request.files['File1']
        file2=request.files['File2']
        path1='static/files/'+str(uuid.uuid4())+str(file1.filename)
        file1.save(path1)
        splitpath=path1.split('.')
        types=splitpath[1]
        print(types,'.....')

```

```

if types=='txt':
    value=txtreader(path1)
if types=='docx':
    value=docreader(path1)
if types=='pdf':
    value=pdfreader(path1)
# upfl=trade_spider(file1)
path2='static/files/'+str(uuid.uuid4())+str(file2.filename)
file2.save(path2)
splitpath2=path2.split('.')
types1=splitpath2[1]
print(types1)
if types=='txt':
    value1=txtreader(path2)
if types=='docx':
    value1=docreader(path2)
if types=='pdf':
    value1=pdfreader(path2)
print(value1)
vals=value+value1
print(".....")
result = run_summarization(vals)
print(result)
print("////////////////////////////////////")
# speech(result)
data['details']=result
session['result']=result

if 'submit' in request.form:

```

```

        speech(session['result'])

        data['s']=session['result']

    return render_template('doc_upload.html',data=data)

@user.route('/file_upload',methods=['get','post'])
def file_upload():
    data={ }

    if 'news' in request.form:

        news1=request.form['news1']

        news2=request.form['news2']

        if 'news2'=="":

            valmain = news1

            result = run_summarization(valmain)

            print(result)

            # speech(result)

            data['details']=result

            session['result']=result

        else:

            valmain = news1 + news2

            result = run_summarization(valmain)

            print(result)

            # speech(result)

            data['details']=result

            session['result']=result

    if 'submit' in request.form:

        speech(session['result'])

    return render_template('file_upload.html',data=data)

@user.route('/user_view_news_providers')
def user_view_news_providers():

```

```

data={ }

q="SELECT * FROM `news_providers` "

select(q)

res=select(q)

data['pro']=res

return render_template('user_view_news_providers.html',data=data)

@user.route('/user_view_news_categories')

def user_view_news_categories():

    data={ }

    q="SELECT * FROM `news_categories`"

    select(q)

    res=select(q)

    data['category']=res

    return render_template('user_view_news_categories.html',data=data)

def speech(text):

    print("sg")

    engine = pyttsx3.init() # object creation

    """ RATE"""

    rate = engine.getProperty('rate') # getting details of current speaking rate

    print (rate)                #printing current voice rate

    engine.setProperty('rate', 125) # setting up new voice rate

    """VOLUME"""

    volume = engine.getProperty('volume') #getting to know current volume level
(min=0 and max=1)

    print (volume)                #printing current volume level

    engine.setProperty('volume',1.0) # setting up volume level  between 0 and 1

    """VOICE"""

    voices = engine.getProperty('voices') #getting details of current voice

```



```
#engine.setProperty('voice', voices[0].id) #changing index, changes voices. 0 for
male
engine.setProperty('voice', voices[1].id) #changing index, changes voices. 1 for
female

engine.say(text)

# engine.say('My current speaking rate is ' + str(rate))

engine.runAndWait()

engine.stop()

# speech("hai")
```

database.py

```
import mysql.connector

password=""

database="text_summarization"

def select(q):

    cnx=mysql.connector.connect(password=password,database=database,host="localhost",user="root",port=3306)

    cur=cnx.cursor(dictionary=True)

    cur.execute(q)

    result=cur.fetchall()

    cnx.close()

    cur.close()

    return result

def delete(q):

    cnx=mysql.connector.connect(password=password,database=database,host="localhost",user="root",port=3306)

    cur=cnx.cursor(dictionary=True)

    cur.execute(q)

    cnx.commit()

    result=cur.rowcount

    cur.close()

    cnx.close()

    return result

def update(q):

    cnx=mysql.connector.connect(password=password,database=database,host="localhost",user="root",port=3306)
```

```
cur=cnx.cursor(dictionary=True)

cur.execute(q)

cnx.commit()

result=cur.rowcount

cur.close()

cnx.close()

return result
```

```
def insert(q):
```

```
    cnx=mysql.connector.connect(password=password,database=database,host="localhost",user="root",port=3306)

    cur=cnx.cursor(dictionary=True)

    cur.execute(q)

    cnx.commit()

    result=cur.lastrowid

    cur.close()

    cnx.close()

    return result
```

public.py

```

from flask import Blueprint,render_template,request,redirect,url_for,session
from database import *

public=Blueprint('public',__name__)

@public.route('/')

def home():

    return render_template('index.html')

@public.route('/registration',methods=['get','post'])

def registration():

    if 'submit' in request.form:

        fname=request.form['fname']

        lname=request.form['lname']

        pno=request.form['phno']

        mail=request.form['mail']

        place=request.form['place']

        uname=request.form['uname']

        pas=request.form['passw']

        q="INSERT INTO `login`(`username`,`password`,`user_type`) VALUES

('%s','%s','user')"%(uname,pas,)

        id=insert(q)

        q1="INSERT INTO

`users`(login_id,`first_name`,`last_name`,`Phone`,`Email`,`Place`) VALUES

('%s','%s','%s','%s','%s','%s')"%(id,fname,lname,pno,mail,place)

        insert(q1)

        return redirect(url_for("public.registration"))

    return render_template('registration.html')

@public.route('/login',methods=['get','post'])

```

```

def login():
    if 'login' in request.form:
        uname=request.form['uname']
        passwd=request.form['passwd']
        q="SELECT * FROM `login` WHERE `Username`='%s' AND
`Password`='%s'"%(uname,passwd)
        res=select(q)
        if res:
            if res[0]['user_type']=="admin":
                return redirect(url_for('admin.adminhome'))
            elif res[0]['user_type']=="user":
                #q="select * from reg_form where
log_id='%s'"%(session['login_id'])
                #res=select(q)
                #session['reg_id']=res[0]['reg_id']
                return redirect(url_for('user.userhome'))
        return render_template('login.html')

```

functions.py

```
import PyPDF2

import spacy

def pdfreader(path):

    pdfFileObj = open(path, 'rb')

    pdfReader = PyPDF2.PdfFileReader(pdfFileObj)

    pdfReader.numPages

    pageObj = pdfReader.getPage(0)

    pageObj.extractText()

    return pageObj.extractText()

def docreader(path):

    from docx import Document

    document = Document(path)

    for para in document.paragraphs:

        return para.text

def txtreader(path):

    filepath =path

    data = ""

    with open(filepath) as fp:

        line = fp.readline()

        cnt = 1

        while line:

            data = data + line

            line = fp.readline()

            cnt += 1

    return data
```

summarize.py

```

from nltk.corpus import stopwords

from nltk.stem import PorterStemmer

from nltk.tokenize import word_tokenize, sent_tokenize

from flask import *

import nltk

nltk.download

import requests

from bs4 import BeautifulSoup

def test_function():

    print("Hello World")

def trade_spider(url):

    source_code = requests.get(url)

    plain_text = source_code.text

    soup = BeautifulSoup(plain_text)

    # print(soup)

    val=""

    # for link in soup.find_all('p', {'itemprop', 'articleBody'}):

    for link in soup.find_all('p'):

        # href = "https://en.wikipedia.org/" + link.get('href')

        title =link.string

        # print(title)

        val=val+convert_string(title)

    # print("sgedgdg",val)

    return val

source_code = requests.get(urls)

```

```

plain_text = source_code.text

soup = BeautifulSoup(plain_text)

# print(soup)

val=""

# for link in soup.find_all('p', {'itemprop', 'articleBody'}):
for link in soup.find_all('p'):

    # href = "https://en.wikipedia.org/" + link.get('href')

    title =link.string

    # print(title)

    val=val+convert_string(title)

return val

def convert_string(value):

    new_value = str(value)

    return new_value

def _create_frequency_table(text_string) -> dict:

    stopWords = set(stopwords.words("english"))

    words = word_tokenize(text_string)

    ps = PorterStemmer()

    freqTable = dict()

    for word in words:

        word = ps.stem(word)

        if word in stopWords:

            continue

        if word in freqTable:

            freqTable[word] += 1

        else:

            freqTable[word] = 1

    return freqTable

```



```

def _score_sentences(sentences, freqTable) -> dict:
    sentenceValue = dict()
    for sentence in sentences:
        word_count_in_sentence = (len(word_tokenize(sentence)))
        word_count_in_sentence_except_stop_words = 0
        for wordValue in freqTable:
            if wordValue in sentence.lower():
                word_count_in_sentence_except_stop_words += 1
            if sentence[:10] in sentenceValue:
                sentenceValue[sentence[:10]] += freqTable[wordValue]
            else:
                sentenceValue[sentence[:10]] = freqTable[wordValue]
            if sentence[:10] in sentenceValue:
                sentenceValue[sentence[:10]] = sentenceValue[sentence[:10]] /
word_count_in_sentence_except_stop_words
        return sentenceValue

def _find_average_score(sentenceValue) -> int:
    sumValues = 0
    for entry in sentenceValue:
        sumValues += sentenceValue[entry]
    average = (sumValues / len(sentenceValue))
    return average

def _generate_summary(sentences, sentenceValue, threshold):
    sentence_count = 0
    summary = ""
    for sentence in sentences:
        if sentence[:10] in sentenceValue and sentenceValue[sentence[:10]] >= (threshold):
            summary += " " + sentence

```

```
        sentence_count += 1

    return summary

def run_summarization(text):

    # 1 Create the word frequency table
    freq_table = _create_frequency_table(text)

    # 2 Tokenize the sentences
    sentences = sent_tokenize(text)

    # 3 Important Algorithm: score the sentences
    sentence_scores = _score_sentences(sentences, freq_table)

    # 4 Find the threshold
    threshold = _find_average_score(sentence_scores)

    # 5 Important Algorithm: Generate the summary
    summary = _generate_summary(sentences, sentence_scores, 1.3 * threshold)

    return summary
```

SQL

db.sql

```
USE `text_summarization`;

DROP TABLE IF EXISTS `login`;

CREATE TABLE `login` (
  `login_id` int(11) NOT NULL AUTO_INCREMENT,
  `Username` varchar(30) DEFAULT NULL,
  `Password` varchar(30) DEFAULT NULL,
  `user_type` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`login_id`)
) ENGINE=MyISAM AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;

/*Data for the table `login` */

insert into `login`(`login_id`,`Username`,`Password`,`user_type`) values
(1,'admin','admin','admin'),(2,'user','user','user');

/*Table structure for table `news_categories` */

DROP TABLE IF EXISTS `news_categories`;

CREATE TABLE `news_categories` (
  `category_id` int(11) NOT NULL AUTO_INCREMENT,
  `category_name` varchar(100) DEFAULT NULL,
  `category_description` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`category_id`)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;

/*Data for the table `news_categories` */

insert into `news_categories`(`category_id`,`category_name`,`category_description`) values
(2,'category 1','category description'),(3,'category','description');
```

```
/*Table structure for table `news_content` */
```

```
DROP TABLE IF EXISTS `news_content`;
```

```
CREATE TABLE `news_content` (
```

```
  `news_id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `category_id` int(11) DEFAULT NULL,
```

```
  `Title` varchar(30) DEFAULT NULL,
```

```
  `Date` varchar(20) DEFAULT NULL,
```

```
  `Description` varchar(60) DEFAULT NULL,
```

```
  PRIMARY KEY (`news_id`)
```

```
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
```

```
insert into `news_content`(`news_id`,`category_id`,`Title`,`Date`,`Description`) values  
(2,2,'title1','2021-02-12','description_1');
```

```
DROP TABLE IF EXISTS `news_providers`;
```

```
CREATE TABLE `news_providers` (
```

```
  `provider_id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `provider_name` varchar(100) DEFAULT NULL,
```

```
  `provider_contact` varchar(100) DEFAULT NULL,
```

```
  `description` varchar(100) DEFAULT NULL,
```

```
  PRIMARY KEY (`provider_id`)
```

```
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
```

```
insert into `news_providers`(`provider_id`,`provider_name`,`provider_contact`,`description`)  
values (3,'provider23','9876321562','desc');
```

```
DROP TABLE IF EXISTS `users`;
```

```
CREATE TABLE `users` (
```

```
  `user_id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `login_id` int(11) DEFAULT NULL,
```

```
  `first_name` varchar(50) DEFAULT NULL,
```

```
  `last_name` varchar(50) DEFAULT NULL,
```

```

`Phone` varchar(20) DEFAULT NULL,
`Email` varchar(50) DEFAULT NULL,
`Place` varchar(60) DEFAULT NULL,
PRIMARY KEY (`user_id`)
) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;

insert into `users`(`user_id`,`login_id`,`first_name`,`last_name`,`Phone`,`Email`,`Place`)
values
(1,2,'user','name','1234567890','user@gmail.com','userhome'),(2,3,'1',' ',' ',' '),(3,4,'5',' ',' ',' '),(4,
5,'m',' ',' ',' '),(5,6,'kk',' ','1111111111',' ');

```

13. BIBILOGRAPHY

- | | |
|------------------------------------------------------------|-------------------------------------|
| 1. Automatic text summarization | Juan Manuel and Torres Moreno |
| 2. From Extractive to Abstractive Summarization: A Journey | Parth Mehta |
| 3. Automatic summarization | Inderjeet Mani |
| 4. Speech And Language Processing | Daniel Jurafsky and James H. Martin |
| 5. Handbook of Natural Language Processing | Nitin Indurkha and Fred J. Damerau |
| 6. Big Data Analytics Method | Peter Ghavami |
| 7. Natural Language Understanding | James Allen |

14. WEBIOGRAPHY

1. <https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25>
2. https://en.wikipedia.org/wiki/Automatic_summarization
3. <https://stackabuse.com/text-summarization-with-nltk-in-python/>
4. https://ieeexplore.ieee.org/abstract/document/7860024?source=post_page-----

5. <https://ieeexplore.ieee.org/document/5234971>
6. <https://blog.floydhub.com/gentle-introduction-to-text-summarization-in-machine-learning/>