# Swastik College

## (Affiliated to Tribhuvan University)

**Chardobato, Bhaktapur**

**A Project Report**

**On**

**"News Guardian-Fake News Detection Portal"**

**In partial fulfillment of the requirements for the Bachelor of Science in Computer Science and Information Technology**

**Submitted to:**

**Department of Computer Science and Information Technology**

**Swastik College**


**Submitted by:**

**Ayush Tripathi (26848/077)**

**Kebal Khadka (26861/077)**

**Pranav Thapa (26864/077)**


**Under the Supervision of**

**Sristi Khatiwada**

**January 2025**

# SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by Ayush Tripathi, Kebal Khadka and Pranav Thapa entitled "NEWS GUARDIAN–FAKE NEWS DETECTION SYSTEM" in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Information Technology be processed for the evaluation. They possess excellent communication skills and is always willing to go above and beyond in fulfilling their responsibilities. They consistently met or exceeded expectations on every task assigned to them.

.....

Ms. Sristi Khatiwada

Swastik College

# CERTIFICATE OF APPROVAL

This is to certify that, this project entitled "NEWS GUARDIAN–FAKE NEWS DETECTION SYSTEM" submitted by Ayush Tripathi, Kebal Khadka and Pranav Thapa is partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Information Technology has been Studied. In our opinion, it is satisfactory in scope and quality as a project for the required degree.

.....

Ms. Sristi Khatiwada

Supervisor

……

External Examiner

……

Ms. Sristi Khatiwada

Coordinator

Swastik college

Chardobato, Bhaktapur

# ABSTRACT

The proliferation of fake news has emerged as a significant global challenge, undermining trust and spreading misinformation. News Guardian addresses this issue by offering a tech-driven solution for detecting fake news, recommending personalized news content, and classifying news as child-safe or mature. Utilizing machine learning techniques, the system integrates Naive Bayes and Random Forest classifiers for accurate fake news detection, while content-based filtering algorithms deliver tailored news recommendations based on user preferences. Furthermore, K-means clustering is employed to classify news content into child-safe or mature categories, enhancing user safety.

The platform's core innovation lies in its ability to merge advanced algorithms with intuitive user interfaces, empowering individuals to access credible, relevant, and safe news. Designed with a Python Django backend and React frontend, News Guardian ensures seamless functionality, supported by SQLite for efficient data storage. Initial evaluations highlight its effectiveness in improving news credibility and user satisfaction. In conclusion, News Guardian stands as a comprehensive solution to combat fake news, enrich personalized news experiences, and ensure content safety, with potential for continual enhancement to meet the dynamic needs of the digital information landscape.

*Keywords: Fake News Detection, Content-Based Filtering, Naive Bayes, Random Forest, K-means Clustering, React, Python Django.*

# ACKNOWLEDGEMENT

The report on project entitled “NEWS GUARDIAN–FAKE NEWS DETECTION SYSTEM” is made as a partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Information Technology. We are hoping that this project will be beneficial to the concerning bodies.

We would like to express our sincere gratitude to Ms. Sristi Khatiwada, who is our program coordinator as well as our project supervisor for her invaluable guidance, unwavering support and mentorship throughout the duration of project. Her valuable and timely suggestions at crucial stages, along with constant encouragement have made it possible for us to accomplish this work. Lastly, we extend our heartfelt thanks to everyone who has contributed to this project, directly or indirectly, and helped in its completion.

Thanking you,

Ayush Tripathi

Kebal Khadka

Pranav Thapa

# Table of Contents

# List of Abbreviation

| | |
|---|---|
| AI | ARTIFICIAL INTELIGENCE |
| CNN | CONVOLUTIONAL NEURAL NETWORK |
| CSS | CASCADING STYLE SHEET |
| CSV | COMMA SEPARATED VALUE |
| DFD | DATA FLOW DIAGRAM |
| ER | ENTITY RELATIONSHIP |
| GRU | GATED RECURRENT UNIT |
| LSTM | LONG SHORT-TERM MEMORY |
| TF-IDF | TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY |
| UI | USER INTERFACE |
| UML | UNIFIED MODELING LANGUAGE |

# List of Figures

# List of Tables

# Chapter1 Introduction

## 1.1 Introduction

In today's digital age, online news has transformed how people stay informed, offering instant access to updates on global events, diverse perspectives, and a wide array of topics. It has become an integral part of daily life, allowing users to stay connected and aware of the world around them. However, the convenience of online news also comes with challenges, such as information overload and the growing threat of misinformation, making it difficult for users to find credible and relevant content.

News Guardian aims to redefine how people access and interact with news in the digital age by offering a platform that delivers the latest updates while recommending personalized content tailored to individual preferences. It focuses on ensuring users stay informed with news that is not only relevant but also engaging, enhancing the overall experience through dynamic personalization and classification. By prioritizing user needs, News Guardian aspires to create a platform that helps users easily navigate the vast amount of information available today, ensuring they receive timely and meaningful updates.

In addition to delivering personalized content, News Guardian will address the growing issue of misinformation by incorporating advanced algorithms to predict and identify fake news. It aims to empower users to make informed decisions by distinguishing between credible and misleading information, fostering a sense of trust and reliability. With its commitment to accessibility, authenticity, and user engagement, News Guardian seeks to bridge the gap between information delivery and credibility, creating a trusted space for modern readers to explore and connect with the world of news.

## 1.2 Problem Statement

In the current digital landscape, the vast availability of online news has made it increasingly difficult for users to find reliable and relevant information amidst an overwhelming influx of content. The rise of misinformation and fake news has further eroded trust in online platforms, leaving readers vulnerable to manipulation and confusion. Existing news platforms often fail to provide personalized recommendations, resulting in a disconnected and impersonal user experience that doesn't cater to individual interests.

Furthermore, many platforms lack robust tools to detect and combat fake news effectively, contributing to the spread of misinformation. Inadequate user-friendly interfaces and

outdated designs across devices also hinder accessibility, limiting users' ability to engage meaningfully with credible content. News Guardian aims to address these challenges by offering a platform that delivers personalized news recommendations, predicts and prevents fake news, and enhances user engagement through a responsive and intuitive design.

## 1.3 Objectives

The main objective of this project is:

- To create a platform that delivers real time news, predicts fake news for credibility, classify news and personalizes recommendations to enhance user experience.

## 1.4 Scope and Limitation

### 1.4.1 Scope

The News Guardian project focuses on the detection of fake news by analyzing news articles and classifying them as true or false. The system aims to provide real-time identification of misleading or false information, ensuring that users are presented with accurate and reliable news. Additionally, the platform offers personalized news recommendations based on user preferences, and it categorizes news content as child-safe or mature to provide a tailored experience for users of all ages.

### 1.4.2 Limitation

Some of the limitation of the project are:

- Accuracy of prediction depends on the quality and diversity of the training data.
- Scaling to handle large volumes of articles could impact performance.
- News verification is limited to text-based content analysis, with no support for verifying images, videos, or other multimedia content.

## 1.5 Methodology

The Agile methodology guides the project by enabling flexibility, iterative development, and responsiveness to change. Through sprints, each lasting one to two weeks, the project has been broken down into manageable tasks that deliver functional pieces of the application. This process allows for continuous improvement, ensuring alignment with user needs and project goals. Regular sprint reviews and retrospectives help the team assess progress, address challenges, and make necessary adjustments. Agile emphasis on collaboration and communication keeps the project on track, facilitating the rapid

incorporation of complex features like content filtering recommendations, clustering and fake news detection, ensuring that the final product is both high-quality and user-centric. [1]

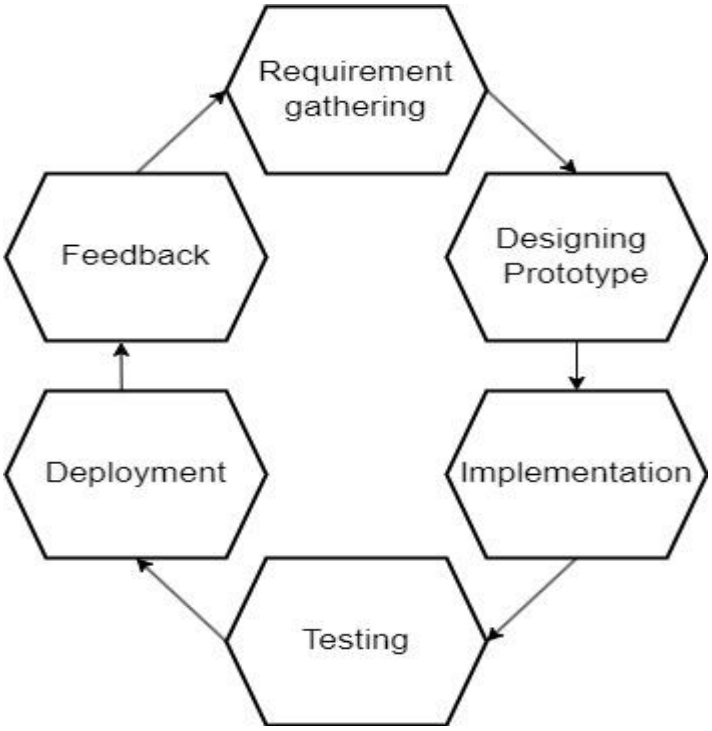



**Figure 1.1 Agile Development Methodology**

## 1.6 Report Organization

The project is broken up into six isolated chapters, each of which describes a different stage of development. The following is a summary of the chapters:

**Chapter 1** is the overview of the project including the introduction, problem statement, objectives, scope, limitations, and development methodology.

**Chapter 2** focuses on the background study and literature review related to the project for reference in project development.

**Chapter 3** involves system analysis, including requirement analysis and feasibility analysis.

**Chapter 4** includes the project design and modeling.

**Chapter 5** focuses on the implementation and testing of the system.

**Chapter 6** is the conclusion of the report and includes future recommendations. Furthermore, the reference section includes a list of all the sources cited in the report.

# Chapter2 Background and Literature Review

## 2.1 Background Study

The rapid spread of fake news has become a significant challenge in the digital age, driven by the growth of social media and online platforms. Misinformation disrupts the flow of credible information, influences public opinion, and creates societal mistrust. It exists in various forms, such as false news, satirical content, and poorly written articles, making its detection complex and demanding advanced solutions.

Several online systems have emerged to address this issue. Factmata employs AI-driven tools to detect misinformation, hate speech, and bias, while Hoaxy visualizes the spread of fake news and fact-checking efforts on social media. Check4Spam verifies viral content using fact-checked databases, and Botometer identifies social bots responsible for spreading fake news. Platforms like Full Fact provide real-time fact-checking tools, NewsGuard offers trust ratings for news websites, and browser extensions like SurfSafe and B.S. Detector flag suspicious sites. [2]

News Guardian aims to build upon success of these systems by integrating advanced technologies, including Naive Bayes and Random Forest for fake news detection, content-based filtering for personalized recommendations, and K-means clustering for classifying news as child-safe or mature. By providing a user-friendly, reliable platform, News Guardian seeks to enhance user trust, ensure access to credible information, and tackle misinformation effectively in today's digital landscape.

## 2.2 Literature Review

The literature on fake news detection explores various machine learning models, natural language processing techniques, and their applications to tackle the challenges posed by misinformation. Key themes and findings from the literature are summarized below:

The 2018 Science article by D. M. J. Lazer, M. A. Baum, Y. Benkler [3] offers a succinct summary of the research on the fake news phenomena. The writers examine fake news's many facets, such as its prevalence, consequences, and methods of distribution. Along with the psychological and cultural variables that contribute to the dissemination and acceptance of false information, they explore the function of social media platforms and the algorithms that define the information ecosystem.

H. Allcott and M. Gentzkow, through their journal [4] examined the role of social media platforms in the spread of fake news during the 2016 United States presidential election. The authors provide a comprehensive review of the existing literature on the subject, analysing the factors that contributed to the prevalence of fake news, its impact on the election, and the potential remedies to address this issue.

Fake reviews and fake news both are responsible for spreading misconceptions and disbeliefs among the people, according to the article [5] .The article has also emphasized that detecting fake news is even harder than the detection of fake reviews. It has categorized fake news into 3 groups: the first being false news, second as fake satire news, and lastly poorly written news articles. Here, the authors have introduced models for detecting both fake news and fake reviews.

In the paper by Kelly Stahl [6], they have considered past and current techniques for fake news identification in text formats while elucidating how and why fake news exists in any case. This paper incorporates a discussion on how the writing style of a paper can also impact its classification. They implemented their project using Naïve Bayes Classifier and Support Vector Machines methods. They looked into the semantic analysis of the text for classification.

The research article [7] has demonstrated the use of the Naive Bayes classifier for fake news detection. The authors utilized data from Facebook news posts for implementing the Naive Bayes classifier and achieved a decent accuracy of around 74%. They also emphasized that using a more complex model could lead to even greater accuracy and efficiency.

In article [8] various Machine Learning models like Naive Bayes, K nearest neighbour, Decision Tree, Random Forest and Deep Learning networks (CNN, LSTM, GRU) are used. He has also explored and used features like n-gram, TF-IDF. All these models were used and their corresponding accuracies were pointed out due to which evaluation becomes easier. On comparing all these models, CNN LSTM using together was found to have the highest accuracy of 97.3%.

# Chapter3 System Analysis

## 3.1 System Analysis

### 3.1.1 Requirement Analysis

**i. Functional Requirements:**

1. Account Creation and Login: Users should be able to create an account and log in to access the platform and interact with its features.
2. View News by Category: Users should be able to view news articles organized into specific categories for easy navigation.
3. Determine News Authenticity: User should be able to determine whether the news is fake or true.
4. Receive Recommendation: The system should recommend personalized news articles based on the user's preferences.
5. View Clustered News: User should be able to view clustered news as child safe or mature news.

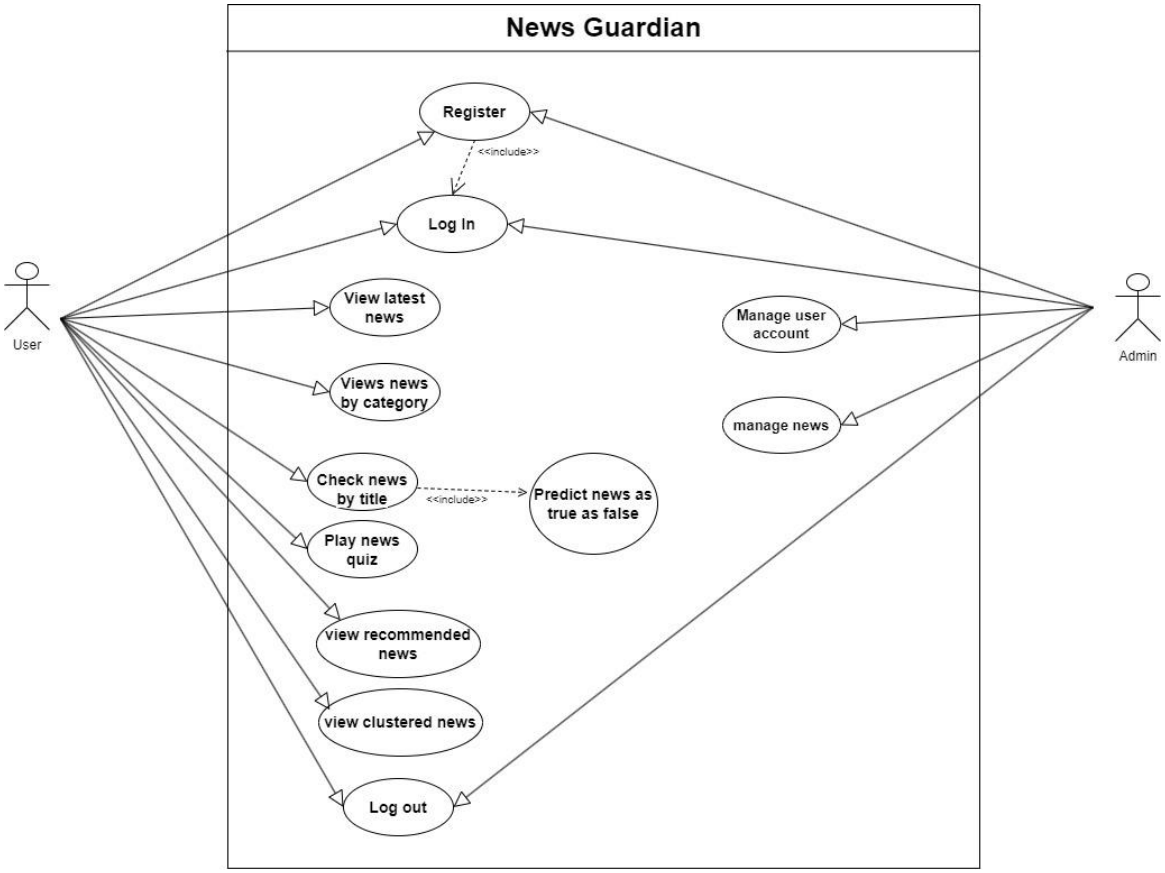



**Figure 3.2 Use Case Diagram of News Guardian**

**ii. Non-Functional Requirements:**

1. Response Time: The system should provide quick response ensuring minimal delays in determining news authenticity and delivering recommendations.
2. User Interface: The system should have an intuitive and user-friendly interface for seamless navigation and interaction with platform.
3. Accessibility: The system should be usable on standard screen sizes, including mobile devices, with basic support for screen readers.

### 3.1.2 Feasibility Study

A feasibility study is an incredibly crucial activity that is done for the development of a proper news portal. Following steps have been done in order to check the feasibility of the software:

**i. Technical Feasibility**

The system is a web application that works on existing software and hardware, requiring only an internet connection. It is compatible with any operating system

and is developed using freely available technologies.

**ii. Economic Feasibility**

All the resources are available to do the project is free of cost thus further expenses is not required.

**iii. Operational Feasibility**

The project aligns with user requirements and can be smoothly integrated into existing systems thus can be easily operable and maintained after deployment.

**iv. Schedule Feasibility**

The project can be completed within the allotted time frame according to the proposed Gantt chart.

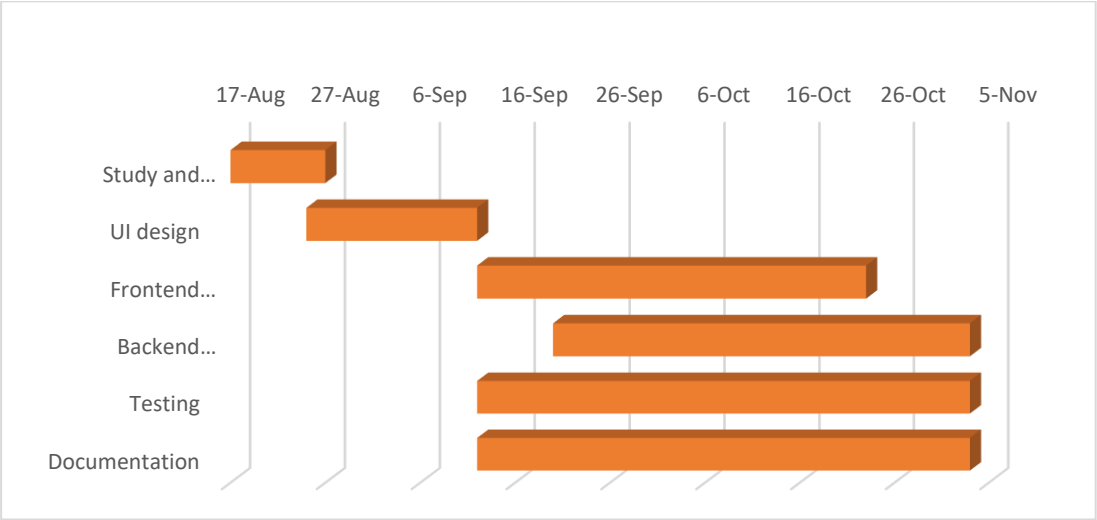



**Figure 3.3 Gantt Chart of News Guardian**

## 3.2 Analysis

### i. Flow Chart

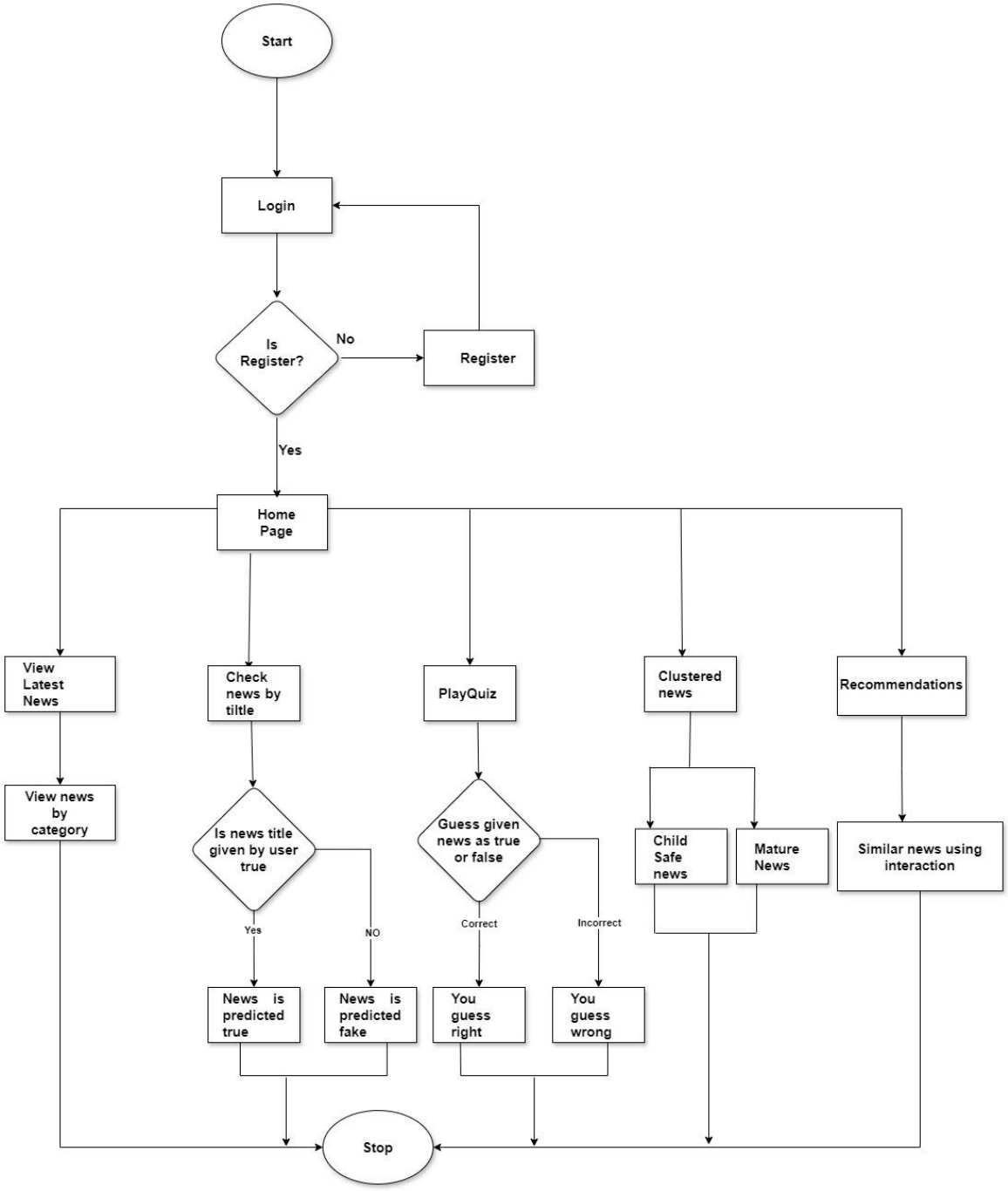



**Figure 3.4 Flowchart of News Guardian**

The above flowchart represents the user journey in News Guardian, starting with login or registration. After accessing the homepage, users can view the latest news, check news authenticity by title, play a quiz to guess news accuracy, and explore clustered news

categorized as child-safe or mature. The system also provides personalized news recommendations based on user interactions. The process concludes after users access their desired information.

### ii. DFD

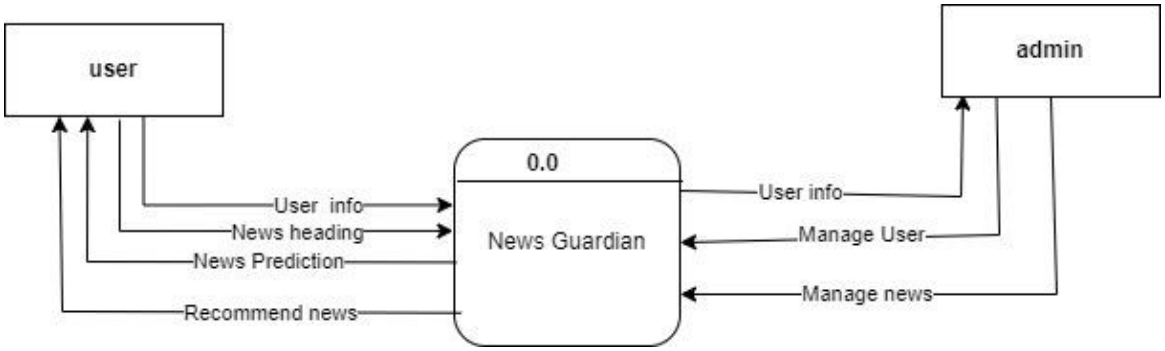



**Figure 3.5 Level 0 Data Flow Diagram**

The Level 0 DFD of News Guardian provides an overview of the system, illustrating interactions between users and admins. Users provide personal details, submit news headings, and receive news predictions and recommendations. Admins manage users and news data within the system. The system processes inputs and generates relevant outputs by interacting with external entities.

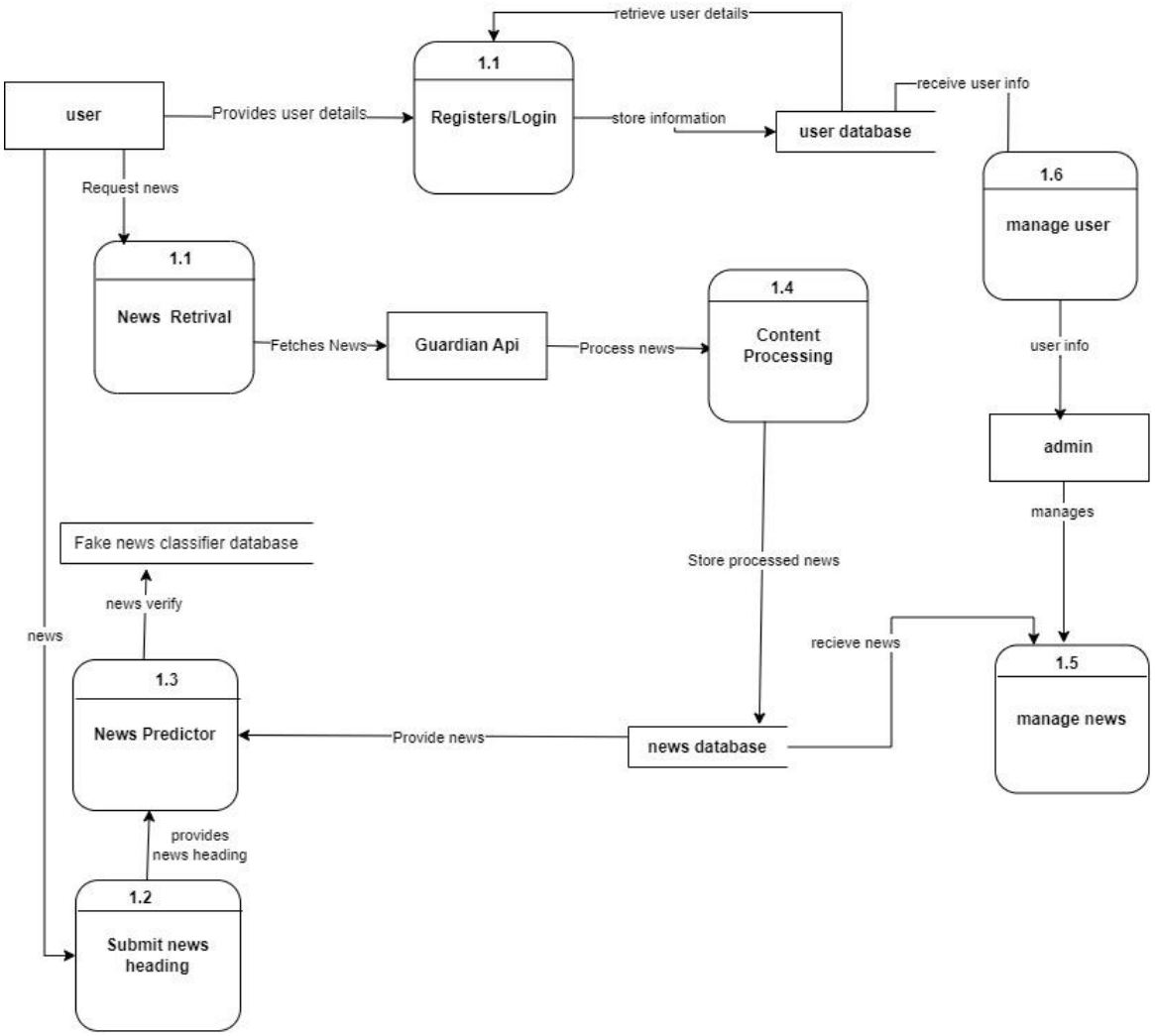



**Figure 3.6 Level 1 Data Flow Diagram**

The Level 1 DFD breaks down the system into specific processes such as user registration, news retrieval, and content processing. Users can register/login, request news from the Guardian API, and submit news for verification. The system processes and stores news, while admins manage users and news data. News predictions and recommendations are provided based on stored information

### iii. ER Diagram

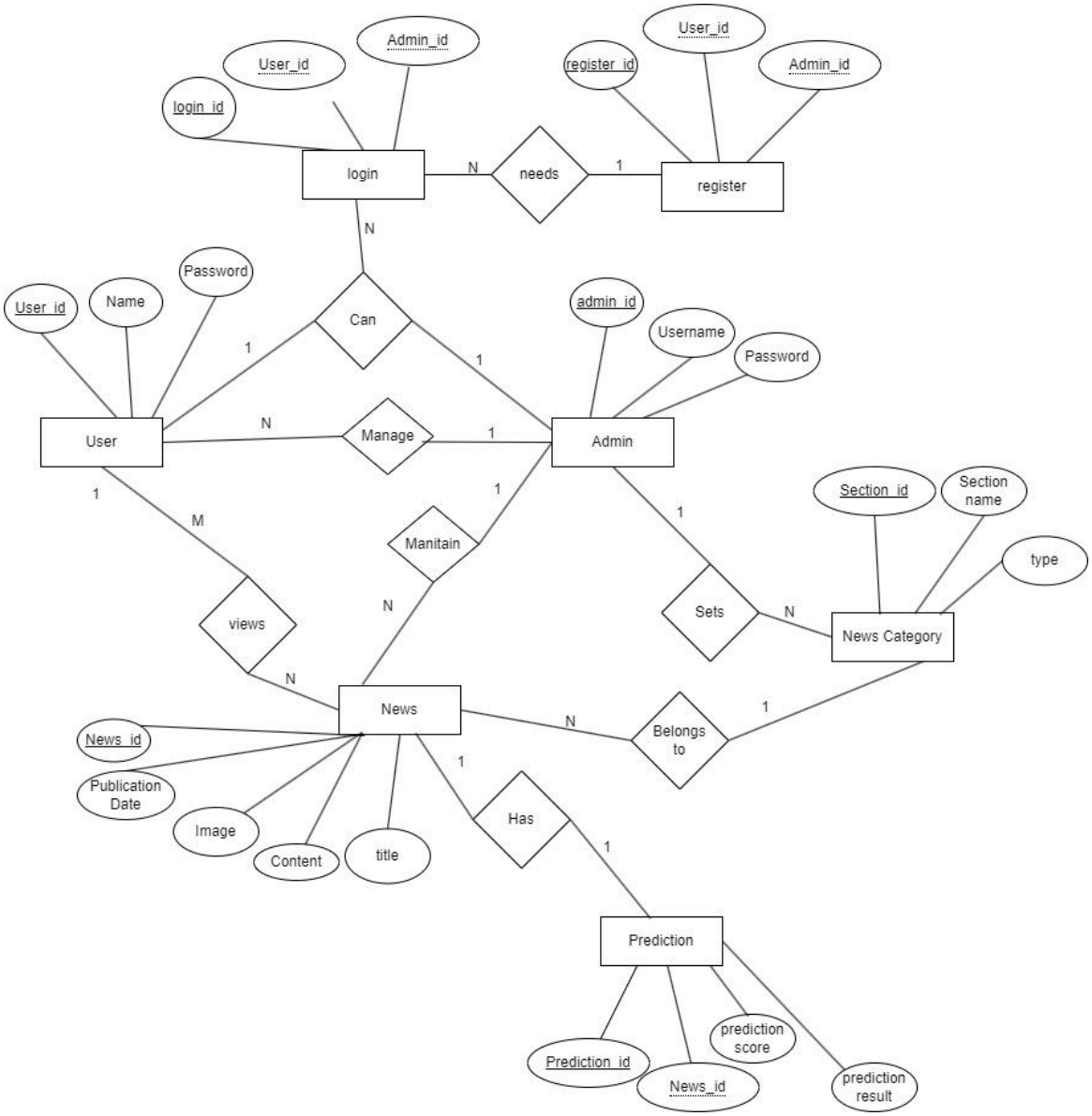



**Figure 3.7 E-R Diagram of News Guardian**

The above ER diagram for News Guardian represents the database structure and relationships between key entities. The user entity allows users to register, log in, and view news articles. The admin entity manages users and maintains news content. News articles belong to different news categories, which are set by admins. Each news article has attributes such as title, content, image, and publication date. The system includes a prediction entity that determines whether news is true or fake, storing attributes like prediction score and result. The relationships define how users interact with news and how admins manage content within the platform.

# Chapter4 System Design

## 4.1 Database Design

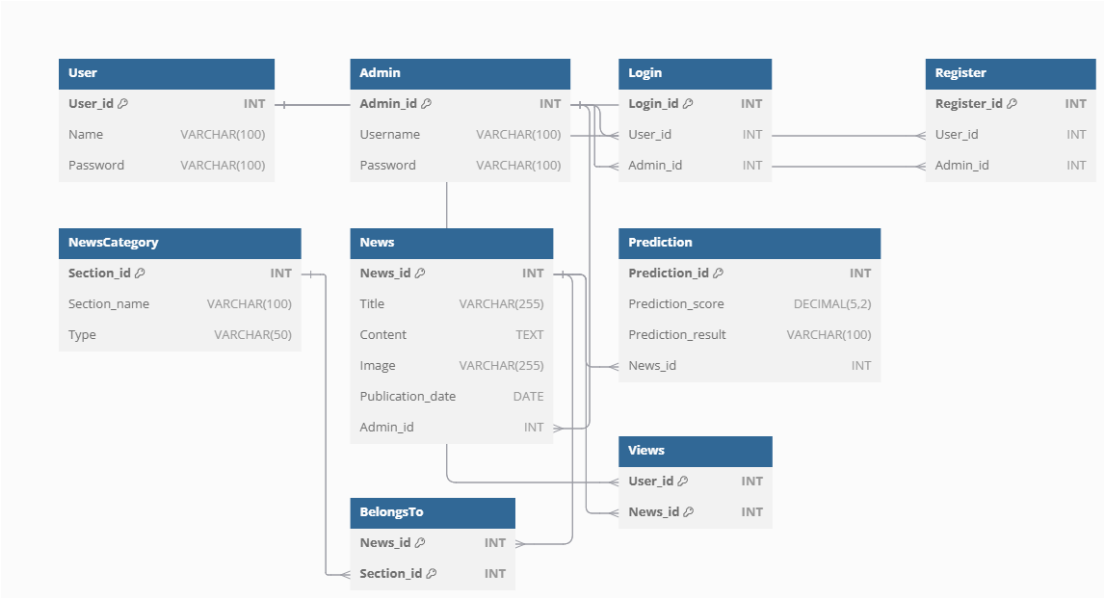



**Figure 4.1 Database Design for News Guardian**

The News Guardian database design defines the structure and relationships between key entities. Users and admins manage authentication, while news articles are stored with details like title, content, and publication date. News is categorized through the news category table and linked via the belongs to table. The prediction table analyzes news authenticity, and the views table tracks user interactions. This design ensures efficient news management and fake news detection.

## 4.2 Form and Report Design

Forms are an important part of a system as they are used to get data from the users. The different forms that will be required for the system are login and registration for admin and user.

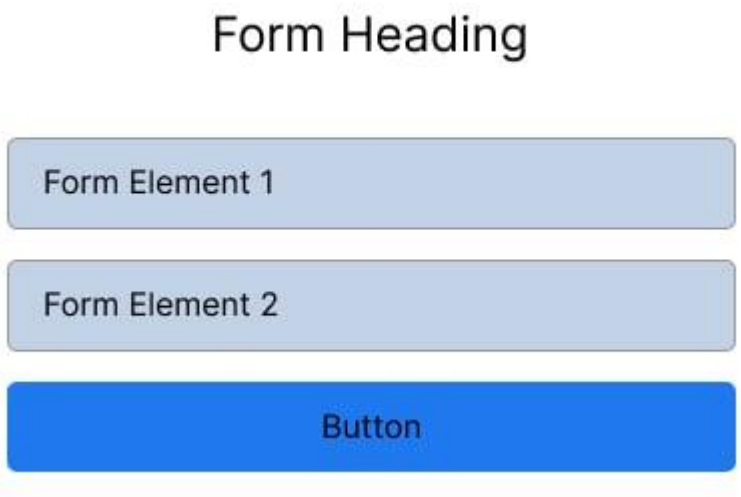



**Figure 4.2 Form and Report Design**

## 4.3 Interface Design

UI/Interface is one of the most important parts of the system as it determines how easy it is for a new user using the system to understand the different components listed and navigate through them in order to achieve the intended goal of using the system. The interface of our system will be presented according to the following representations:

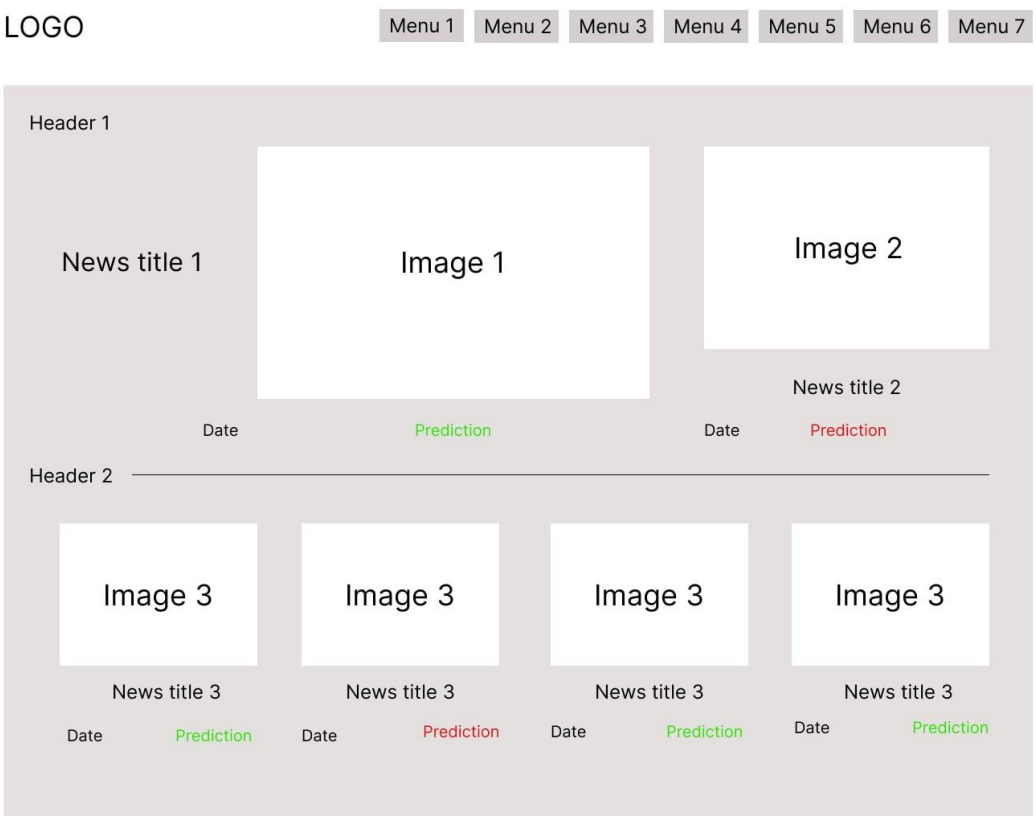



**Figure 4.3 Interface for Home Page**

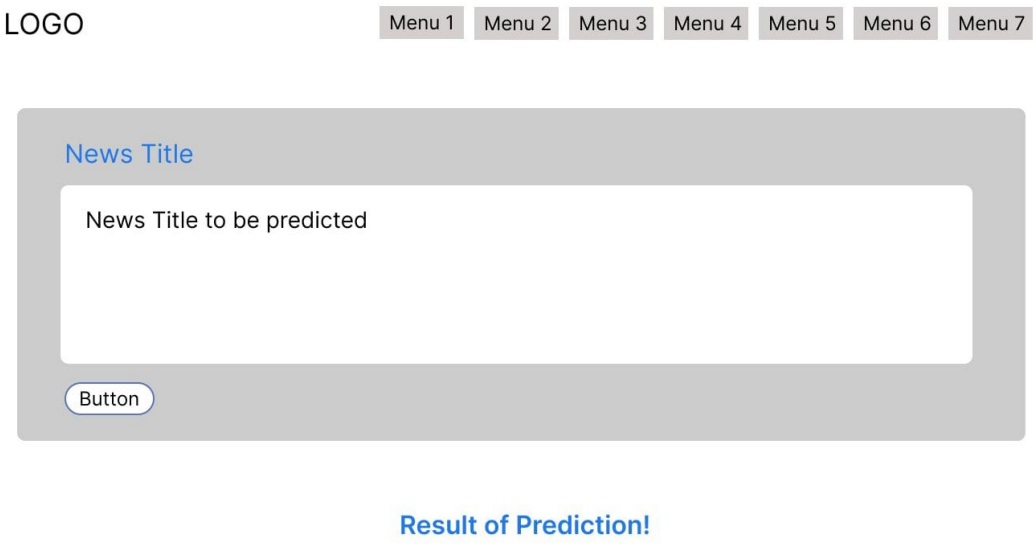



**Figure 4.4 Interface Design of News Title Page**

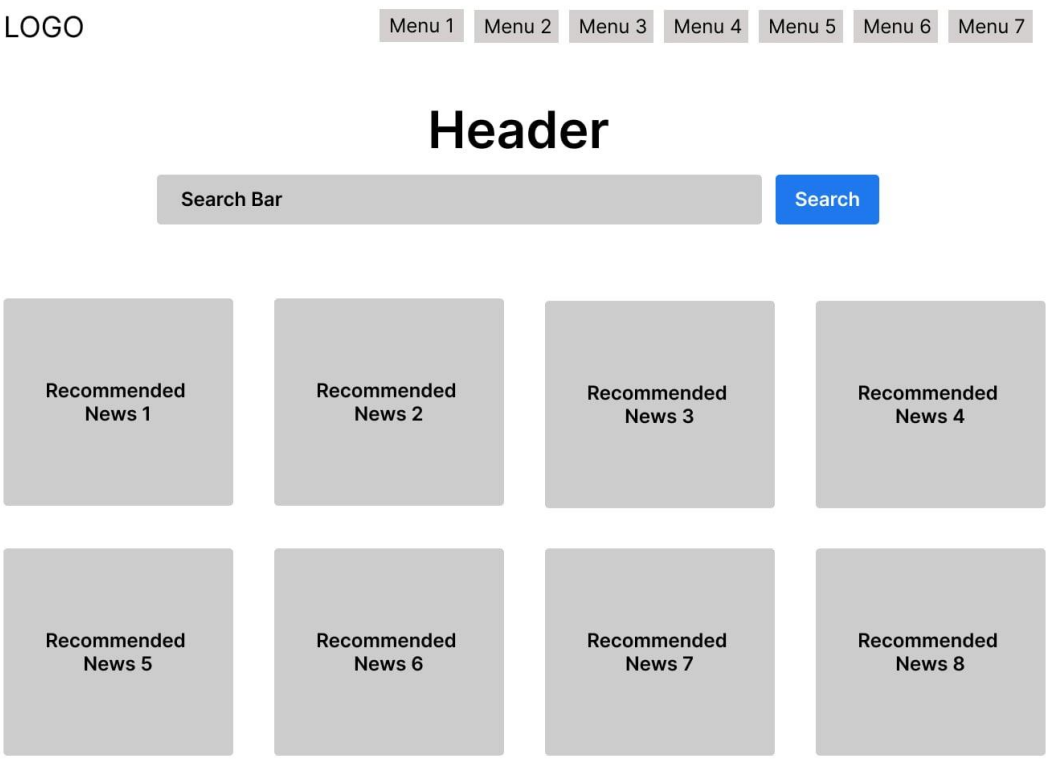



**Figure 4.5 Interface Design of For You Page**

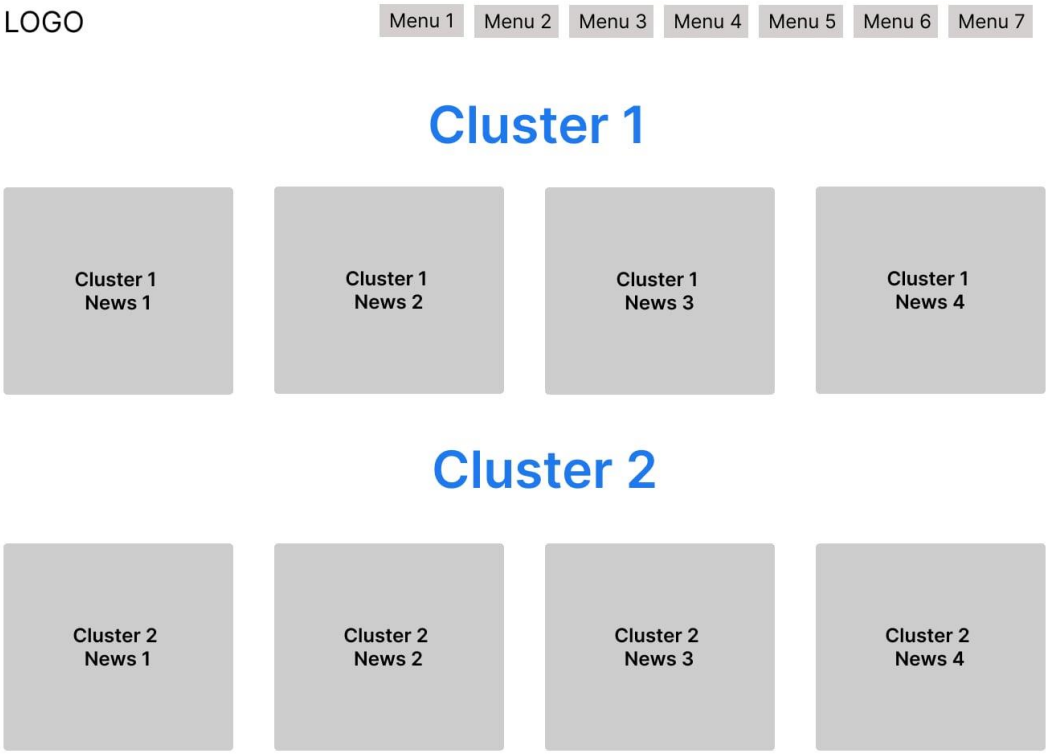



**Figure 4.6 Interface Design of Clustered Page**

## 4.4 Algorithm Details

### i. Content Based Filtering Recommendation Algorithm:

Content-based filtering is a recommendation algorithm that suggests news articles to users based on the features of the content and their personal preferences. In News Guardian, this involves extracting key features from each news article, such as keywords, topics, and categories. When new articles are available, the system calculates the similarity between the article's features using techniques like cosine similarity. Articles with higher similarity scores are recommended to the user. This approach enhances the user experience by providing relevant and engaging news tailored to individual preferences. [9]

**Term Frequency (TF)**

$$\mathrm{TF}(\mathrm{t},\mathrm{d}) = \frac{\text{Number of times term t appears in document d}}{\text{Total number ofterms in document d}}$$

**Inverse Document Frequency (IDF)**

$$\mathrm{IDF} = \log \frac{\text{total number of documents in the corpus D}}{\text{Number of document where term t appear}}$$

**TF-IDF Score**

$$\mathrm{TF}-\mathrm{IDF}(\mathrm{t},\mathrm{d},\mathrm{D}) = \mathrm{TF}(\mathrm{t},\mathrm{d}) \times \mathrm{IDF}(\mathrm{t},\mathrm{D})$$

**Building Document Vectors**

$$\mathrm{vd} = [\mathrm{TF}-\mathrm{IDF}(\mathrm{t1},\mathrm{d},\mathrm{D}), \mathrm{TF}-\mathrm{IDF}(\mathrm{t2},\mathrm{d},\mathrm{D}), \ldots, \mathrm{TF}-\mathrm{IDF}(\mathrm{tn},\mathrm{d},\mathrm{D})$$

**Cosine Similarity**

$$\text{Cosine Similarity}(\underset{\mathrm{vd}}{\rightarrow}, \underset{,\mathrm{vq}}{\rightarrow}) = \frac{\overrightarrow{\mathrm{vd}}.\overrightarrow{\mathrm{vq}}}{||\mathrm{vd}||*||\mathrm{vq}||}$$

### ii. K-means Clustering:

In News Guardian, K-means clustering is applied to categorize news articles into "Child-Safe" or "Mature/Violent" groups based on their content. The algorithm begins by transforming the textual features of articles, such as keywords, topics, and descriptions, into numerical representations using techniques like TF-IDF. These numerical vectors are then fed into the K-means algorithm, which groups articles into $k$ = 2 clusters, one representing child-safe news and the other representing mature/violent content. The grouping is achieved by minimizing the Euclidean distance

between articles within the same cluster and their respective cluster centroids. To determine the nature of a cluster, predefined labels are assigned based on content analysis. Articles in clusters containing sensitive language, violent descriptions, or mature themes are labeled "Mature/Violent," while others are marked as "Child-Safe." This automated categorization helps provide age-appropriate recommendations, ensuring a safer user experience for younger audiences while maintaining content relevance for others. As new articles are added, the system dynamically updates clusters to reflect any changes in the dataset. [10]

**Term Frequency (TF)**

$$\mathrm{TF(t,d)} = \frac{\text{Number of times term t appears in document d}}{\text{Total number ofterms in document d}}$$

**Inverse Document Frequency (IDF)**

$$\mathrm{IDF} = \log \frac{\text{total number of documents in the corpus D}}{\text{Number of document where term t appear}}$$

**TF-IDF Score**

$$\mathrm{TF-IDF(t,d,D)} = \mathrm{TF(t,d)} \times \mathrm{IDF(t,D)}$$

**Building Document Vectors**

$$\mathrm{vd} = [\mathrm{TF-IDF(t1,d,D)}, \mathrm{TF-IDF(t2,d,D)}, \ldots, \mathrm{TF-IDF(tn,d,D)}$$

**Euclidean Distance**

$$\mathrm{d} = \sqrt{[\,(\mathrm{x2} - \mathrm{x1}\,)^2 + (\mathrm{y2} - \mathrm{y1}\,)^2]}$$

**Centroid Update**

$$c_j = \frac{1}{n_j} \sum_{i=j}^{n_j} x_i$$

### iii. Naive Bayes Algorithm:

The Naive Bayes algorithm is implemented as the first model in the project. After the text is transformed into numerical data using ‘CountVectorizer’, the ‘MultinomialNB’ class from ‘sklearn’ is used to fit the model to the training data. This algorithm assumes that the occurrence of each word in a text is independent of others, given the class label (fake or real). During training, it calculates probabilities for each word in both classes based on their frequencies. When predicting, the algorithm multiplies these

probabilities to determine the class with the highest likelihood. Naive Bayes is particularly effective here because it is computationally efficient, making it well-suited for processing large text datasets. Its simplicity allows the model to quickly classify news articles based on word frequencies, providing a strong baseline for comparison. [11]

### iv. Random Forest Algorithm:

The Random Forest algorithm is introduced as the second model in the project. This ensemble method builds multiple decision trees during training, each trained on a random subset of the dataset and features. The 'Random Forest Classifier' with 300 estimators (trees) is used in the code. Each tree generates its own prediction, and the final classification is determined by majority voting across all trees. Random Forest excels at capturing complex patterns and interactions in the data, which makes it highly accurate for classification tasks. Unlike Naive Bayes, it does not rely on assumptions of independence, allowing it to consider relationships between words. However, it requires more computational power and training time. [12]

# Chapter5 Implementation and Testing

## 5.1 Implementation

With a clear blueprint established through analysis and design, the implementation phase transforms concepts into a physical reality, News Guardian is transformed during this phase, fueled by a selection of tools and technologies that facilitate the creation of a functional and aesthetically pleasing news portal.

### 5.1.1 Tools Used

- **Front-End:** The front-end of the application is built using React.js for a seamless and dynamic user interface, with CSS for creating responsive and visually appealing styling. Context API is utilized for efficient state management, ensuring smooth data flow and component communication.
- **Back-End:** The back-end leverages Python Django to handle server-side logic efficiently. SQLite is used as the database for reliable and lightweight data storage, while the Pandas library facilitates robust data manipulation and processing.
- **Version Control:** Git is a distributed version control system used to track changes in source code during software development. For News Guardian, Git manages the source code, enabling collaborative development and version control. GitHub, a web-based platform, facilitates this collaboration by providing a space for developers to manage and share code using Git.
- **Drawings:** Excel for gantt chart, draw.io for UML diagrams and dbdiagram.io for database design.

### 5.1.2 Implementation Details of Modules

There are different modules in the system described in the details below:

**i. Content Based Filtering Recommendation Algorithm:**

- **Input**

**Example News Title:**

"Real Madrid 3-1 Paris St-Germain (3-2 agg): Karim Benzema hat-trick inspires superb Real fightback"

**CSV File Content (Sample News Titles from Dataset):**

"Ed Sheeran denies Shape of You copyright claim at High Court trial",

"Fury's future plans, AJ's new coach and Khan's rematch ambition - Fight Talk",

"Zimbabwe clinics struggle for nurses after exodus to the UK",

"Toddler tossed to safety from burning building",

"Chelsea: Thomas Tuchel has faith in those in charge of club's future",

"Cancer: 'I thought I had Covid but it was terminal lung cancer'",

"Real Madrid 3-1 Paris St-Germain (3-2 agg): Karim Benzema hat-trick inspires superb Real fightback",

"Covid: Generation of musicians 'devastated' by restriction",

"'Dangerous' tanning products promoted by influencers",

"UK farmers turn to Nepal and Tajikistan for fruit pickers",

"India celebrates festival of lights Diwali",

"Explained: The weather conditions that brought the floods",

"Children to stay in education or training until 18",

"Murder investigation launched after man found shot"

- **Preprocessing**

**Original Title:**
"Real Madrid 3-1 Paris St-Germain (3-2 agg): Karim Benzema hat-trick inspires superb Real fightback"

**After Tokenization (Splitting into Words):**
['Real', 'Madrid', '3-1', 'Paris', 'St-Germain', '3-2', 'agg', 'Karim', 'Benzema', 'hat-trick', 'inspires', 'superb', 'Real', 'fightback']

**After Removing Stopwords, Punctuation, and Digits:**
['Real', 'Madrid', 'Paris', 'St-Germain', 'Karim', 'Benzema', 'hat-trick', 'inspires', 'superb', 'Real', 'fightback']

- **TF-IDF Vectorization**

Each news title is transformed into a TF-IDF vector to capture the importance of words.

- Example TF-IDF Vector for the News Title:

  [0.15, 0.20, 0.07, 0.12, 0.18, 0.25, 0.14, 0.10, 0.13, 0.09]

  (Each number represents the weight of a word in the document based on its frequency and uniqueness across the dataset.)

- **Cosine Similarity Calculation**

The TF-IDF vector of the input news article is compared with all other news articles using cosine similarity to find the most similar ones.

**Comparison of Similarities with Other News Titles:**

"Chelsea: Thomas Tuchel has faith in those in charge of club's future" → Similarity Score: 0.87

"Real Madrid 3-1 Paris St-Germain (3-2 agg): Karim Benzema hat-trick inspires superb Real fightback" → Similarity Score: 1.00 (Self)

"Fury's future plans, AJ's new coach and Khan's rematch ambition - Fight Talk" → Similarity Score: 0.76

"Toddler tossed to safety from burning building" → Similarity Score: 0.45

"Cancer: 'I thought I had Covid but it was terminal lung cancer'" → Similarity Score: 0.30

- **Recommendation Output**

The system sorts the news articles by similarity and returns the top 5 most similar news articles based on cosine similarity.

**Recommended Articles:**

1. "Chelsea: Thomas Tuchel has faith in those in charge of club's future" (Similarity: 0.87)
2. "Fury's future plans, AJ's new coach and Khan's rematch ambition - Fight Talk" (Similarity: 0.76)
3. "UK farmers turn to Nepal and Tajikistan for fruit pickers" (Similarity: 0.65)
4. "Explained: The weather conditions that brought the floods" (Similarity: 0.60)
5. "Covid: Generation of musicians 'devastated' by restriction" (Similarity: 0.58)

- **Interpretation**

The system correctly identified sports-related news as the most similar articles. The TF-IDF representation and cosine similarity effectively determine news articles that share similar terms and context. The model performs well in recommending relevant articles, though fine-tuning stop word removal and using word embeddings could improve it.

### ii. K-means Clustering:

- **Input**

**Example News Titles:**

- "Children to stay in education or training until 18"
- "Murder investigation launched after man found shot"
- "UK farmers turn to Nepal and Tajikistan for fruit pickers"

These titles will be preprocessed and vectorized to generate a TF-IDF matrix that reflects the importance of each term in the corpus of news articles.

- **Preprocessing and TF-IDF Vectorization**

The preprocess_and_vectorize_titles function processes the news titles, tokenizes them, and constructs a TF-IDF matrix. The function follows these steps:

**Tokenization of Titles:**
Each title is converted to lowercase and split into individual words. Example:

"Children to stay in education or training until 18" becomes ['children', 'to', 'stay', 'in', 'education', 'or', 'training', 'until', '18']

"Murder investigation launched after man found shot" becomes ['murder', 'investigation', 'launched', 'after', 'man', 'found', 'shot']

**Building the Vocabulary:**
After tokenization, a word frequency count is performed across all titles. The 1200 most common words are selected for the vocabulary. For simplicity, assume the selected vocabulary contains words like ['children', 'training', 'murder', 'education', 'shot', 'found', 'man', 'stay'].

**TF-IDF Matrix Construction:**
The matrix represents each title as a vector of term frequencies, weighted by inverse document frequency (IDF) to capture the relevance of each term. The result is a TF-IDF matrix where each row corresponds to a title, and each column corresponds to a word in the vocabulary. For instance:

> Title: "Children to stay in education or training until 18"
>
> TF-IDF Vector: [0.25, 0.15, 0.12, 0.10, 0.12, 0.13, 0.10, ...]
>
> (Vector of length 1200 representing term importance for the given title)

**Normalization:**
The matrix is normalized so that each row has a unit norm (i.e., magnitude of 1). This is done by dividing each vector by its L2 norm (Euclidean norm), ensuring that each document vector has equal weight in clustering.

- **Initial Centroid Initialization:**

Centroids are initialized manually with predefined keywords:

> Mature & Violent News Keywords:
> ['war', 'violence', 'death', 'protest', 'conflict', 'Ukraine', 'Russia', 'kill', 'murder']

Child-Safe News Keywords:
['fun', 'happy', 'kids', 'education', 'game', 'music', 'arts', 'anime', 'football']

These keywords are assigned unit vectors in the TF-IDF vector space. The centroids represent two groups:

Mature Centroid (Violent/Conflict News):
Words like "war", "kill", "murder" are weighted more heavily.

Child-Safe Centroid (Non-Violent News):
Words like "education", "fun", "kids" are weighted more heavily.

For example, the centroids may look like:

Mature Centroid: [0.1, 0.2, 0.3, ..., 0.0]

Child-Safe Centroid: [0.05, 0.08, 0.1, ..., 0.2]

- **K-means Clustering Iteration**

The k- means clustering function iterates through a maximum of 100 iterations to assign each news article to one of two clusters (mature/violent news or child-safe news).

**First Iteration:**

Distance Calculation**:** The Euclidean distance is calculated between each news article's vector and the centroids. Each news article is then assigned to the cluster with the closest centroid.

**Cluster Assignments:**

"Children to stay in education or training until 18" will likely be assigned to the Child-Safe cluster.

"Murder investigation launched after man found shot" will likely be assigned to the Mature/Violent cluster.

"UK farmers turn to Nepal and Tajikistan for fruit pickers" might be assigned to the Child-Safe cluster.

Example of initial cluster assignments:

[0, 1, 0]

(Cluster 0 = Child-Safe, Cluster 1 = Mature)

**Centroid Update:**
The centroids are updated by calculating the mean of the news article vectors assigned to each cluster. If no articles are assigned to a centroid, the previous centroid remains unchanged.

- **Convergence Check**

After a few iterations, if the centroids stop changing significantly, the algorithm converges and stops.

**Final Cluster Assignments (After Convergence):**

[0, 1, 0]

(Cluster 0 = Child-Safe, Cluster 1 = Mature)

In this case, "Children to stay in education or training until 18" and "UK farmers turn to Nepal and Tajikistan for fruit pickers" are classified as Child-Safe news, while "Murder investigation launched after man found shot" is classified as Mature/Violent.

- **Output**

Cluster 0 (Child-Safe News):

"Children to stay in education or training until 18"

"UK farmers turn to Nepal and Tajikistan for fruit pickers"

Cluster 1 (Mature/Violent News):

"Murder investigation launched after man found shot"

- **Interpretation**

The K-means clustering algorithm has successfully classified the news titles into two distinct groups: Child-Safe and Mature/Violent. The centroids were initialized with domain-specific keywords, and the algorithm effectively assigned each title to its respective group based on similarity to the centroids. The model shows reasonable accuracy based on the predefined labels, but it can be improved with more sophisticated initialization methods and a larger set of training data.

### iii. Naive Bayes Algorithm:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.88 | 0.91 | 0.89 | 11543 |
| 1 | 0.91 | 0.88 | 0.90 | 12065 |
| accuracy | | | 0.90 | 23608 |
| macro avg | 0.90 | 0.90 | 0.90 | 23608 |
| weighted avg | 0.90 | 0.90 | 0.90 | 23608 |

**Figure 6. 1 Confusion Matrix of Nave Bayes**

The confusion matrix for Naïve Bayes-based News Guardian fake news detection model indicates strong performance with 90% accuracy. The model effectively distinguishes between real and fake news, achieving a precision of 0.88 and recall of 0.91 for fake news (class 0), while for real news (class 1), it attains a precision of 0.91 and recall of 0.88. This suggests that the model correctly classifies most instances but occasionally misclassifies some fake news as real and vice versa. The balanced precision, recall, and F1-scores indicate a reliable detection system with minimal bias toward either class.

### iv. Random Forest Algorithm:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.94 | 0.93 | 0.94 | 11543 |
| 1 | 0.94 | 0.94 | 0.94 | 12065 |
| accuracy | | | 0.94 | 23608 |
| macro avg | 0.94 | 0.94 | 0.94 | 23608 |
| weighted avg | 0.94 | 0.94 | 0.94 | 23608 |

**Figure 6. 2 Confusion Matrix of Random Forest**

The confusion matrix for News Guardian fake news detection model using the Random Forest algorithm shows improved performance with an accuracy of 94%. The model achieves a precision of 0.94 and recall of 0.93 for fake news (class 0), while for real news

(class 1), both precision and recall are 0.94. This indicates a balanced and highly effective classification, with fewer misclassifications compared to the Naïve Bayes model. The high F1-scores (0.94 for both classes) confirm the model's robustness in detecting both real and fake news accurately.

## 5.2 Testing

Testing is a way of determining whether the actual software product meets the expected requirements and ensuring that the program is defect-free.

### 5.2.1 Test Cases for Unit Testing

Unit testing focuses on testing individual components or modules of the News Guardian. In the context of the Fake News Detection system, the following testcases can be considered:

**Table 5. 1 Test Case for register and login**

| SN | Action | Steps | Expected Outcomes | Actual Outcomes | Test Cases |
|---|---|---|---|---|---|
| 1. | Submit registration without details. | Leave all or any of the fields empty and click "Register." | Error message: "There was an error with registration." | Error message displayed. | Pass |
| 2. | Register with valid details | Enter valid username, password, then click "Register. | User is registered successfully. | User is registered successfully. | Pass |
| 3. | Login without details or unregistered user | Leave all or any fields empty or put unregistered user input and | Error message: "Invalid credentials. " | Error message displayed. | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | click "Login." | | | |
| 4. | Login with valid credentials | Enter registered username and correct password and click "Login." | User is logged in successfully. | User logged in successfully. | Pass |

**Table 5. 2 Test Case for News and News Category Display**

| SN | Action | Steps | Expected Outcomes | Actual outcomes | Test Results |
|---|---|---|---|---|---|
| 1. | View latest news | Open the homepage. | Latest news articles are displayed in chronological order. | Latest news displayed. | Pass |
| 2. | Open News Category Section | Click on the "News Category " section in the menu. | News categories (e.g., Sports, Technology, Politics) are displayed. | News categories displayed | Pass |
| 3. | View news articles in a category | Select a category (e.g., Technology). | News articles from the selected category are displayed. | Correct articles displayed. | Pass |

| | | | | | |
|---|---|---|---|---|---|
| 4. | Check dates for every news | Verify each news has a date | Each news article displays its publication date. | Dates displayed for all news. | Pass |
| 5. | Check prediction for news | Verify that each news article has a prediction (True/False). | Each news article shows its prediction as either True or False. | Predictions displayed correctly | Pass |

**Table 5. 3 Test Case for Check by Title Field**

| SN | Action | Steps | Expected Outcomes | Actual Outcomes | Test Results |
|---|---|---|---|---|---|
| 1. | Submit empty news title | Leave the news title field empty and click "Search." | Error message "Please enter a valid news title." | Error message displayed. | Pass |
| 2. | Enter valid news title for detection | Input a valid news title and click "Search." | System determines whether the news is true or false. | Correct result displayed. | Pass |
| 3. | Detect true news | Input a true news title. | System identifies the news as true. | Correct result displayed. | Pass |
| 4. | Detect fake news | Input a fake news title. | System identifies the | Correct result displayed. | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | | news as false. | | |

**Table 5. 4 Test Case for Play Quiz**

| SN | Action | Steps | Expected Outcomes | Actual Outcomes | Test Results |
|---|---|---|---|---|---|
| 1. | Start quiz | Click "Play Quiz" on the home page. | Quiz begins, and the first news headline is displayed. | Quiz started successfully. | Pass |
| 2. | Predict news as true | Read the displayed news and select “real” then submit. | System evaluates the answer and provides correct/incorrect feedback. | Feedback displayed correctly. | Pass |
| 3. | Predict news as false | Read the displayed news headline and select "False," then submit. | System evaluates the answer and provides correct/incorrect feedback. | Feedback displayed correctly. | Pass |
| 4. | Get a new quiz | Click the "Get New Quiz" button during or after completing the quiz. | A new set of quiz questions is displayed. | New quiz loaded successful | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | | | ly. | |

**Table 5. 5 Test Case for Recommendation**

| SN | Action | Steps | Expected Outcomes | Actual Outcomes | Test Results |
|---|---|---|---|---|---|
| 1. | Fetch default recommendations | Open the "For You" page. | Default recommendations are displayed. | Default recommendations displayed. | Pass |
| 2. | Fetch recommendations for a given title | Enter a valid news title in the search bar and click "Search." | Relevant recommendations are displayed. | Relevant recommendations displayed. | Pass |
| 3. | Handle empty input in the search field | Leave the search field blank and click "Search." | Error message: "Please enter a news title." | Error message displayed. | Pass |
| 4. | Validate TF-IDF similarity calculation | Provide a valid title to the backend and check calculation in logs. | TF-IDF and similarity scores are Scores calculated correctly | Scores calculated correctly | Pass |

**Table 5. 6 Test Case for Clustering**

| SN | Action | Steps | Expected Outcomes | Actual Outcomes | Test Results |
|---|---|---|---|---|---|
| 1. | Open clustered news section | Navigate to the clustered news section. | News is categorized as child-safe or mature. | Correct classification shown. | Pass |
| 2. | Verify child-safe news classification | Check child-safe news articles displayed. | Only child-safe news is displayed. | Correct articles shown. | Pass |
| 3. | Verify mature news classification | Check mature news articles displayed. | Only mature news is displayed. | Correct articles shown. | Pass |
| 4. | Test with ambiguous content | Add ambiguous news content and classify. | System assigns the correct category based on clustering. | Correct classification applied. | Pass |

### 5.2.2 Test Cases for System Testing

System testing is done on completed components. System testing aids in

emphasizing theoverall behavior of the module.

**Table 5. 7 System Test Case**

| Test Case | Input | Expected Result | Test Result |
|---|---|---|---|
| User Registration and Login | 1. Register with Username = user1, Password = password123<br>2. Login with Username = user1, Password = password123 | 1. Successful user registration<br>2. Successful login as a registered user | Pass |
| Registration and Login Failures | 1. Leave username or password blank for registration<br>2. Enter Username = user2, Password = password123 (unregistered user login)<br>3. Leave username and password blank for login | 1. Error: 'Please fill all fields'<br>2. Error: 'Invalid credentials'<br>3. Error: 'Please enter credentials' | Pass |
| View News: Latest and Categories | 1. Open homepage<br>2. Click on 'News Category' in the menu<br>3. Select 'Technology' category | 1. Latest news articles displayed in chronological order<br>2. News categories displayed<br>3. Technology news articles displayed | Pass |
| View News Details: Dates and Predictions | 1. Open a news article<br>2. Enter a news title and click 'Search' | 1. Publication date displayed<br>2. System determines | Pass |

| | | | |
|---|---|---|---|
| | | whether news is true or false | |
| News Detection: True/False and Errors | 1. Enter a known true news title<br>2. Enter a known fake news title<br>3. Leave the news title blank and click 'Search' | 1. System identifies the news as true<br>2. System identifies the news as false<br>3. Error: 'Please enter a valid news title' | Pass |
| News Quiz: Start, Predict, and Get New Quiz | 1. Click 'Play Quiz' on homepage<br>2. Read headline, select 'True' or 'False,' and submit<br>3. Click 'Get New Quiz' | 1. Quiz starts, first headline displayed<br>2. System provides correct/incorrect feedback<br>3. New quiz set displayed | Pass |
| News Recommendations: Default and Search | 1. Open 'For You' page<br>2. Enter a news title and click 'Search'<br>3. Leave the search field blank and click 'Search' | 1. Default recommendations displayed<br>2. Relevant recommendations displayed<br>3. Error: 'Please enter a news title' | Pass |
| News Classification: Child-Safe and Mature | 1. Navigate to 'Clustered News' section<br>2. View child-safe news articles<br>3. View mature news articles<br>4. Enter ambiguous news content and classify | 1. News categorized as child-safe or mature<br>2. Only child-safe news displayed<br>3. Only mature news displayed<br>4. Correct classification for ambiguous content | Pass |

# Chapter6 Conclusion and Future Recommendation

## 6.1 Conclusion

The News Guardian system has been developed to address the critical issue of misinformation by providing users with a platform for detecting and understanding fake news. By leveraging Naive Bayes and Random Forest classifiers, the system effectively classifies news as true or fake, while K-means clustering categorizes content into child-safe and mature segments. Content-based filtering ensures personalized news recommendations tailored to user preferences, enriching their news consumption experience. These advanced technologies collectively empower users to identify credible news sources and navigate the digital information space confidently.

The platform offers a comprehensive suite of features, including access to the latest news, category-based browsing, interactive quizzes to predict news authenticity, and personalized recommendations. The clustering mechanism enhances user safety by categorizing sensitive content appropriately. This blend of functionality and usability ensures an engaging, intuitive experience that meets the needs of diverse user groups. The system's ability to provide accurate classifications and meaningful recommendations highlights its effectiveness in combating misinformation while fostering trust in news sources.

In conclusion, News Guardian successfully fulfills its objective of delivering a reliable solution to misinformation and enhancing the user experience. The integration of machine learning algorithms, content-based filtering, and intuitive design makes it a robust and efficient platform for modern news consumption. With continued enhancements and the adoption of emerging technologies, News Guardian can further strengthen its capabilities and remain relevant in the ever-evolving information landscape.

## 6.2 Future recommendations

To further enhance the News Guardian project, the following recommendations can be considered for future development:

- **Integration of multiple news sources:** Expand the system by incorporating additional news platforms, ensuring a wider variety of articles and more diverse perspectives, which will help improve the accuracy and comprehensiveness of news recommendations and fake news detection.

- **User engagement features:** Add features that allow users to interact with news articles, such as upvoting, commenting, or sharing, to foster engagement and create a more dynamic news-sharing environment. This could also enhance the recommendation system by considering user interactions as additional data.
- **Enhanced fake news detection:** Continuously improve the fake news detection algorithms by integrating more advanced techniques, such as deep learning models or leveraging external fact-checking sources, to increase accuracy and reliability in identifying misinformation.
- **Automated Summarization:** Introduce an automated news summarization feature that provides brief overviews of articles, allowing users to quickly grasp key information, especially when dealing with long or complex news stories. This could enhance the user experience for those with limited time.

# References


[1] J. Highsmith, "Introduction to agile methodology," in *Agile Project Management*, Pearson Education, 2004, pp. 4-6.

[2] [Online]. Available: https://www.rand.org/research/projects/truth-decay/fighting-disinformation/search.html.

[3] M. A. Baum, M. J. Lazer and Y. Benkler, "The science of fake news," vol. 359, no. 6380, pp. 1094-1096, 2018.

[4] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. XXXI, no. 2, pp. 211-236, 2017.

[5] H. Ahmed, I. Traore and S. Saad, "Detecting opinion spams andfake news using text classification," 2018.

[6] K. Stahl, "Fake news detection in social media," 2018.

[7] M. Mykhailo Granik and V. Mesyura, "Fake news detection using naiveBayes classifier," vol. I, no. 1, pp. 900-903, 2017.

[8] R. K. Kaliyar, "Fake news detection using a deep neural network," pp. 1-7, 2018.

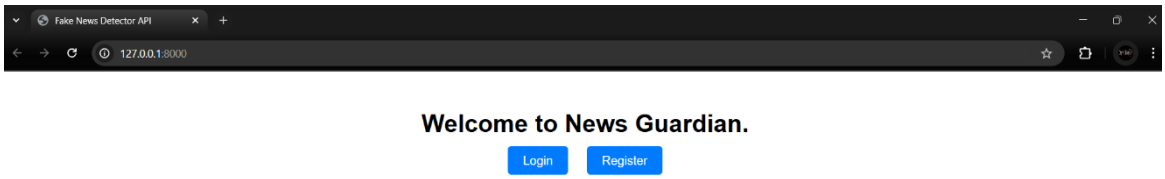
[9] C. C. Aggrawal, "Recommender System: The TextBook," New York, Springer, 2016, pp. 1-28.

[10] [Online]. Available: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning.

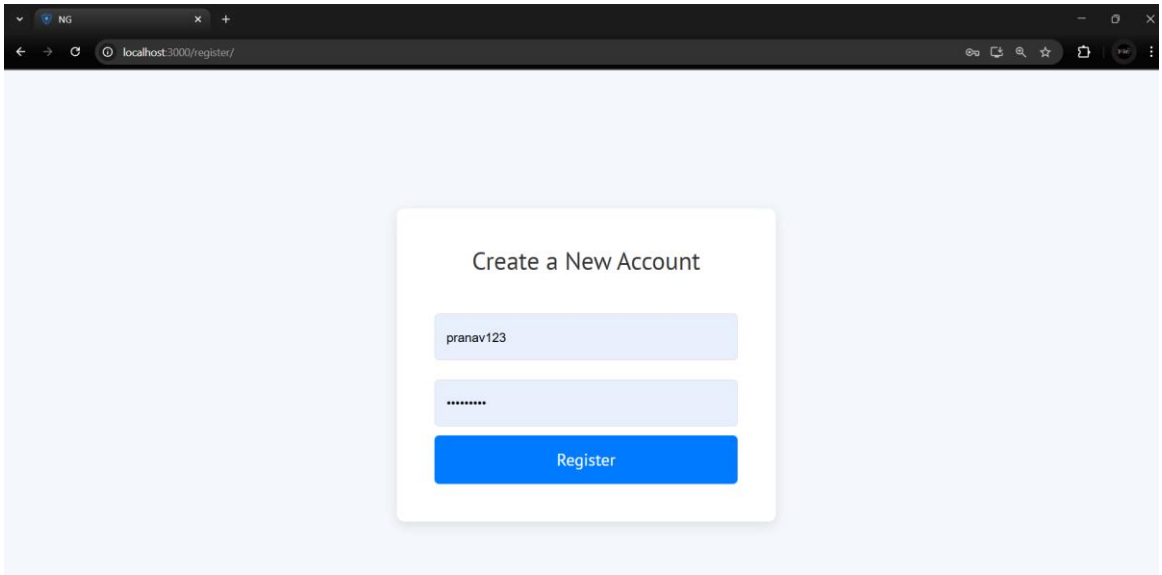
[11] [Online]. Available: https://www.turing.com/kb/an-introduction-to-naive-bayes-algorithm-for-beginners.

[12] [Online]. Available: https://www.turing.com/kb/an-introduction-to-naive-bayes-algorithm-for-beginners.

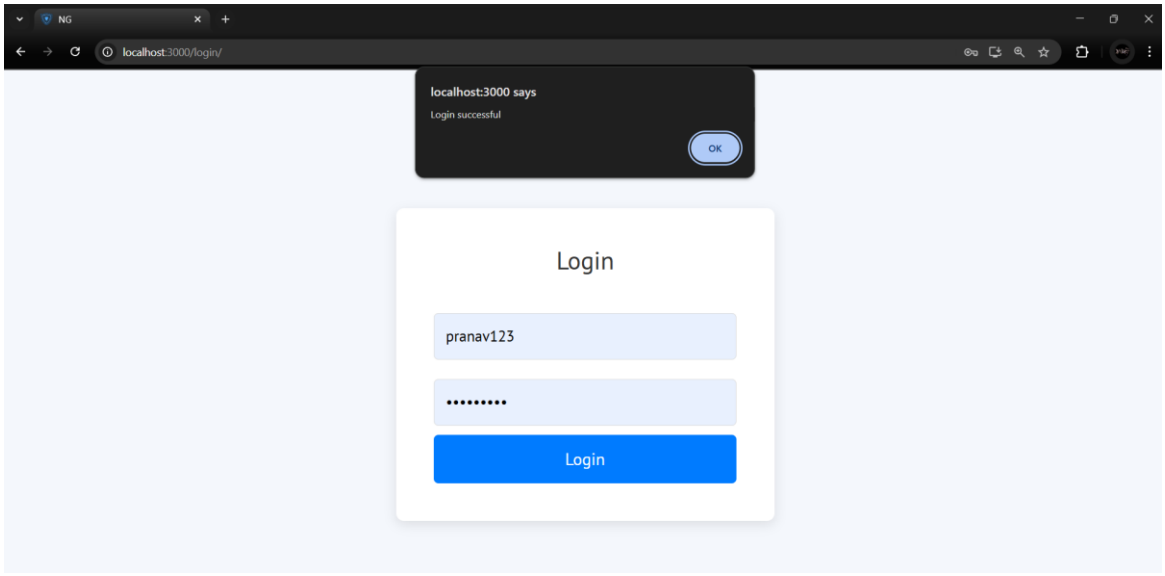
# Appendices



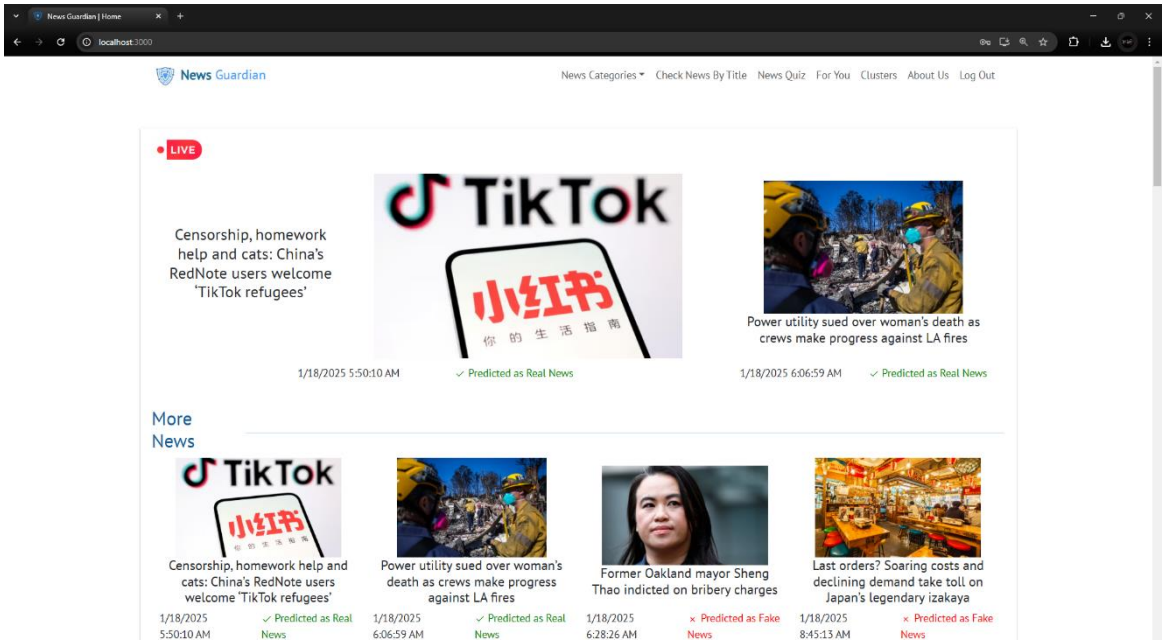

**Annex 1 Welcome Screen**



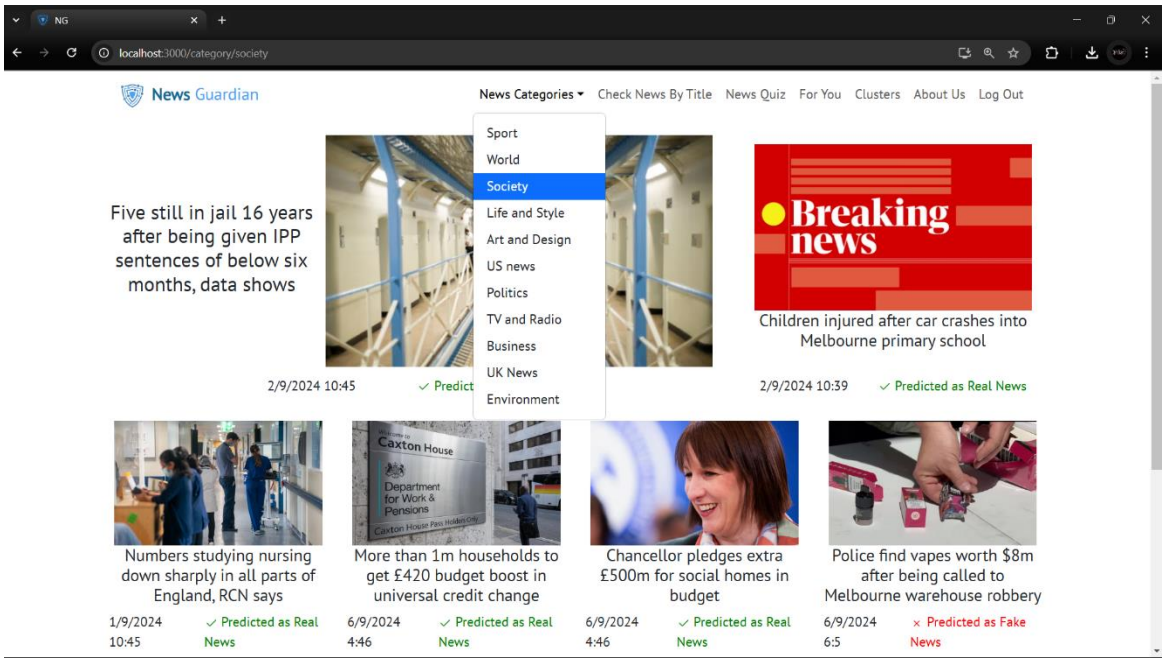

**Annex 2 Registration Page**



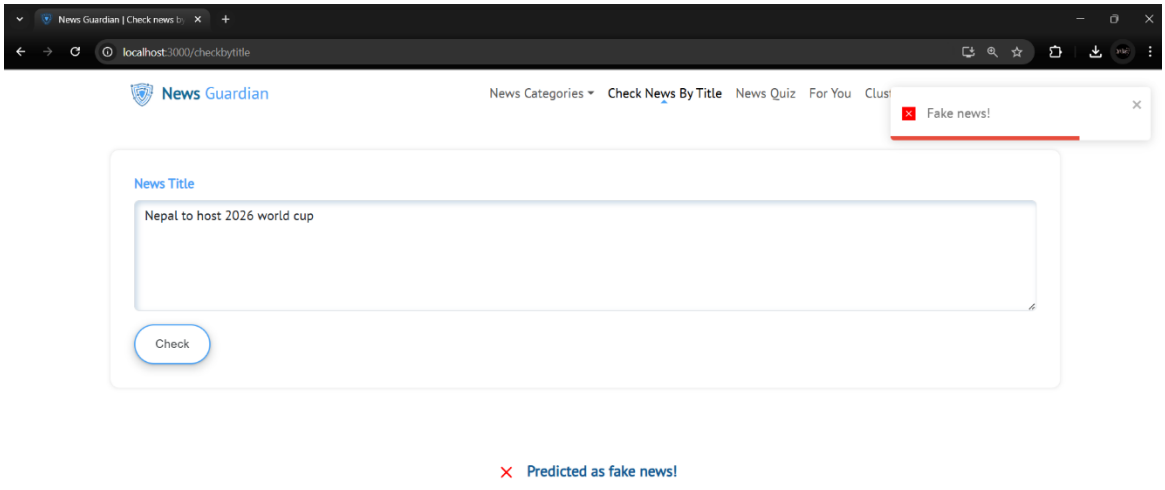

**Annex 3 Login Page**



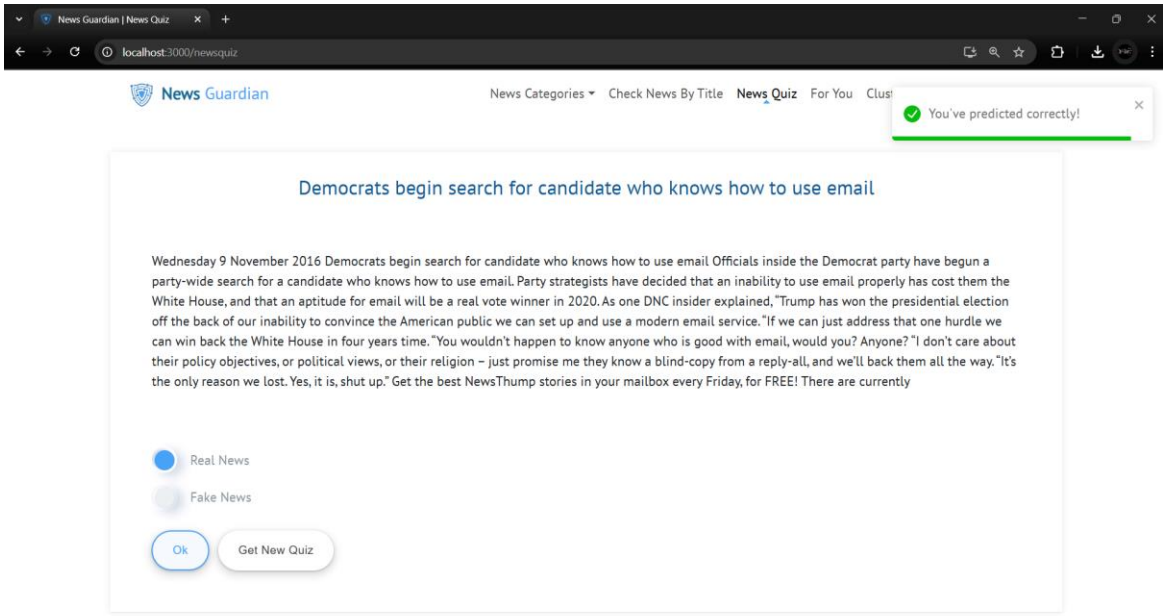

**Annex 4 Home Page**




**Annex 5 News Categories**

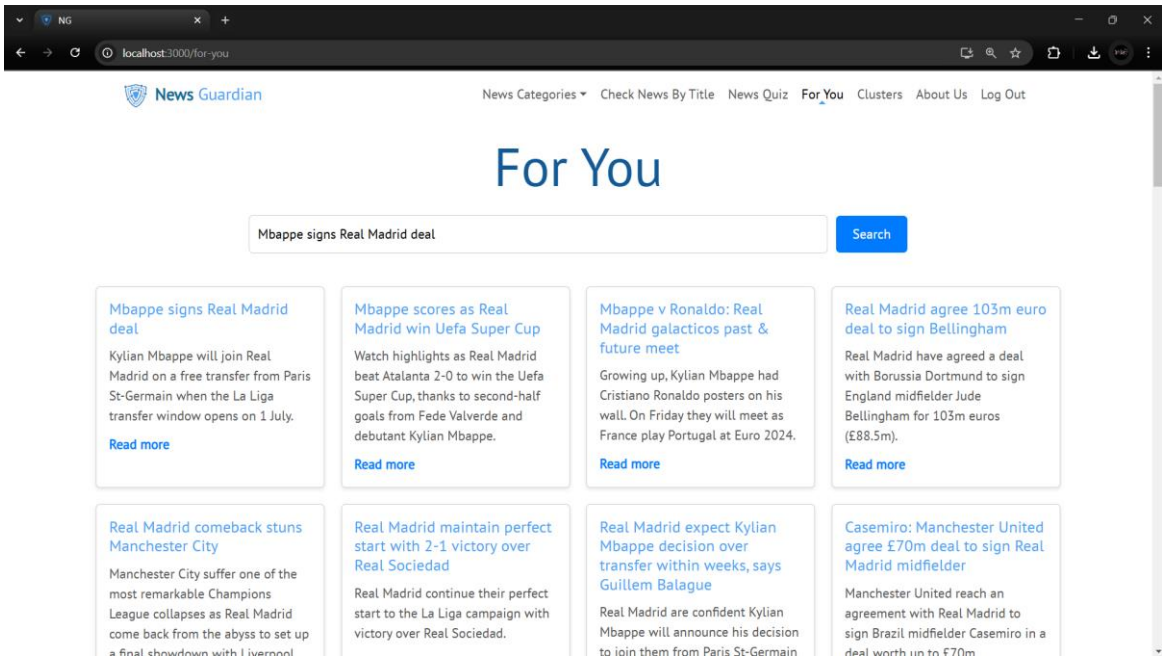



**Annex 6 Check News By Title**

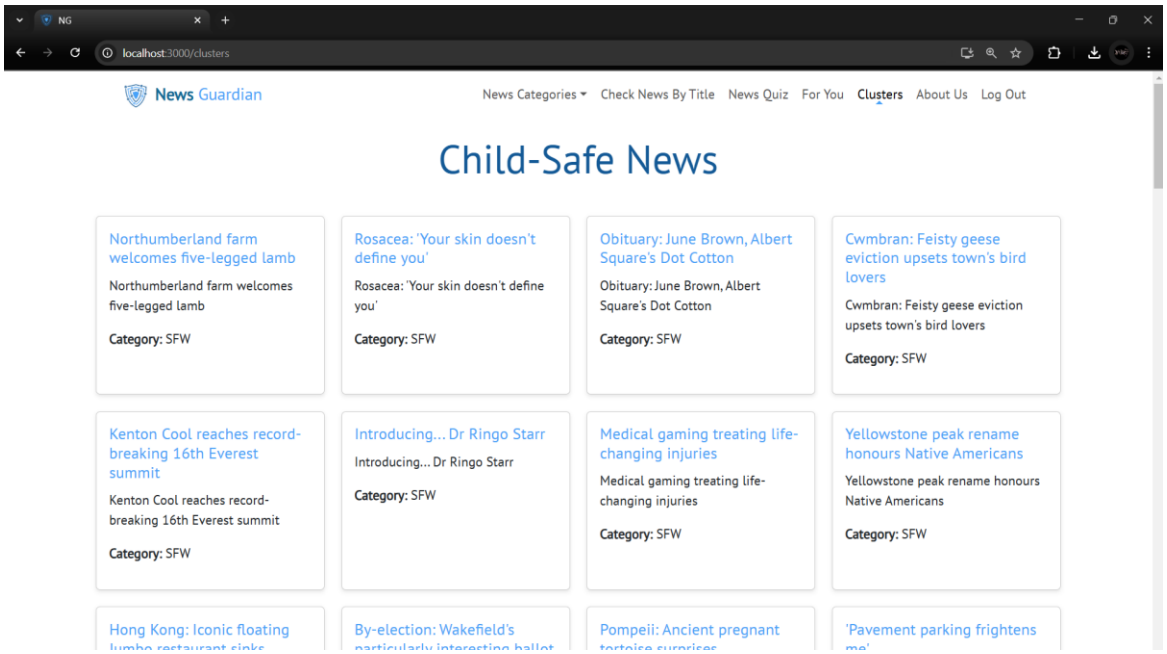


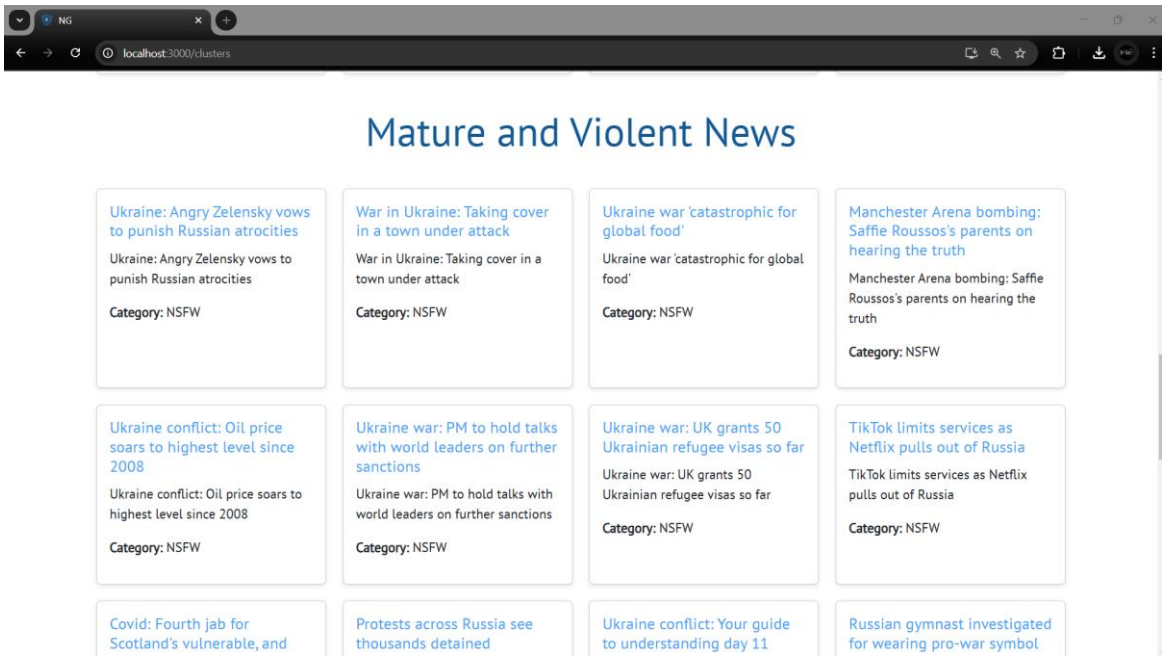

**Annex 7 News Quiz**

For You (Recommendations)




**Annex 8 For You (Recommendation)**




**Annex 9 Clusters**

### Code for Content Based Filtering Recommendation Algorithm:

```
# Function to recommend similar news articles based on the title
def recommend_similar_news(news_data, title):
    # Prepare the list of all titles for the corpus (split into tokens)
    titles = [item['title'].split() for item in news_data]  # Split title into words (tokens)

    # Compute the TF-IDF scores for the entire corpus (titles)
    tfidf_documents = compute_tfidf(titles)

    # Find the index of the current news article based on the title
     title_index = next((index for index, item in enumerate(news_data) if item['title'] == title), None)
    if title_index is None:
        return []  # If the title is not found, return empty list

    # Get the TF-IDF vector for the current news article
    current_news_tfidf = tfidf_documents[title_index]

    # Calculate the cosine similarity between the current news article and all other articles
    similarities = []
    for i, tfidf in enumerate(tfidf_documents):
        if i != title_index:  # Skip the current article itself
            similarity = cosine_similarity(current_news_tfidf, tfidf)
            similarities.append((news_data[i], similarity))

    # Sort by similarity score and return the top 5 most similar news articles
    similarities.sort(key=lambda x: x[1], reverse=True)
    recommended_news = [news for news, _ in similarities[:47]]

    return recommended_news
```

### Code for K-means Clustering:

```
def get_initial_centroids(vocab):
     mature_keywords = ['war', 'violence', 'death', 'protest', 'conflict', 'Ukraine', 'Russia','kill', 'conflict']
```

```
        child_safe_keywords = ['fun', 'happy', 'kids', 'education',
'game','music','arts', 'anime', 'football']

    mature_centroid = np.zeros(len(vocab), dtype=np.float32)
    child_safe_centroid = np.zeros(len(vocab), dtype=np.float32)

    for word, idx in vocab.items():
        if word in mature_keywords:
            mature_centroid[idx] = 1.0
        if word in child_safe_keywords:
            child_safe_centroid[idx] = 1.0

    return np.array([mature_centroid, child_safe_centroid])

def k_means_clustering(tf_idf_matrix, initial_centroids, max_iter=100):
    centroids = initial_centroids.copy()

    for _ in range(max_iter):
        distances = np.linalg.norm(tf_idf_matrix[:, None] - centroids, axis=2)
        cluster_assignments = np.argmin(distances, axis=1)

        new_centroids = np.array([np.mean(tf_idf_matrix[cluster_assignments ==
cluster], axis=0) for cluster in range(2)])
        for i, centroid in enumerate(new_centroids):
            if np.isnan(centroid).any():
                new_centroids[i] = centroids[i]

        if np.allclose(centroids, new_centroids):
            break
        centroids = new_centroids

    return cluster_assignments, centroids
```

**Code for Naive Bayes Algorithm:**

```
# Import the Naive Bayes classifier
from sklearn.naive_bayes import MultinomialNB

# Instantiate the Naive Bayes model
```

```
nb_classifier = MultinomialNB()

# Train the Naive Bayes model using the training data
nb_classifier.fit(count_train, y_train)

# Predict on the test set using the trained model
pred = nb_classifier.predict(count_test)
```

**Code for Random Forest Algorithm:**

```
# Import the Random Forest classifier
from sklearn.ensemble import RandomForestClassifier

# Instantiate the Random Forest model with 300 trees
model = RandomForestClassifier(n_estimators=300)

# Train the Random Forest model using the training data
model.fit(count_train, y_train)

# Predict on the test set using the trained model
pred2 = model.predict(count_test)
```