NEWS CHATBOT

A Project Report

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Soham Sanjay Mokal

2023-2024

ABSTRACT

The "News Chatbot" project aims to address the growing concern of fake news in today's digital age. With the rapid dissemination of news articles through online platforms, it has become increasingly challenging for users to discern between reliable information and misinformation. This project seeks to empower users by providing them with a user-friendly application that can detect and flag potential fake news articles.

The project utilizes machine learning algorithms and natural language processing techniques to analyse news articles and identify indicators of fake news. By considering factors such as misleading information, biased language, and unreliable sources, the application help users make informed decisions about the information they consume.

To enhance the accuracy of fake news detection, the application integrates with reputable fact-checking services and APIs. This integration provides users with additional information about the credibility of news sources and articles, further enabling them to verify the accuracy of the news they encounter.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards the Information and Technology Department of Sathaye College.

After working so hard for months, finally, I am very excited to present my final year project. The project work was full of new experiences and learning and a bit difficult too. We would personally like to Thank and acknowledge all who made our project possible and successful. It is my earnest endeavour to express my sincere regards to the faculty for their kind cooperation, help, and never-ending support.

Firstly, we would take the opportunity and would like to thank our professor **Ms. Larissa Pegado** for her continuous support and encouragement, without which the successful completion of this project would have been impossible. Though it was a very difficult job they made it simpler with timely guidance which helped us greatly in completing our project.

Without her encouragement, constant guidance, interest, and innovative ideas, we would not finish this project. Without her vast knowledge of programming language and various aspects of software development, we would have been unable to accomplish the task of completing this project.

Last but not least I would like to thank all other professors of the Information Technology Department of Sathaye College and every individual who directly and indirectly helped me in this project.

~Thank You!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 BACKGROUND

In recent years, the proliferation of digital media and social networking platforms has led to an exponential increase in the spread of fake news. Fake news refers to deliberately false or misleading information presented as factual news, often with the intention to deceive or manipulate readers. This phenomenon has significant implications for society, as it can distort public opinion, undermine trust in media, and even impact political processes.

The rapid dissemination of news articles through online platforms has made it increasingly challenging for users to discern between reliable information and fake news. Traditional methods of news verification, such as fact-checking by journalists or relying on established news sources, are often insufficient to address the scale and speed at which fake news spreads.

To combat this issue, the development of a "News Chatbot" application for Android devices has become crucial to analyse news articles and identify potential indicators of fake news. By providing users with a reliable tool to verify the credibility of news sources and make informed decisions about the information they consume, the "News Chatbot" application seeks to empower individuals and promote media literacy.

1.2 OBJECTIVES

1. Combat Fake News: The primary objective of the project is to develop an Android application that effectively detects and flags fake news articles. The application aims to identify potential indicators of fake news, such as biased language, unreliable sources, or misleading information.

2. Promote Media Literacy: The project aims to promote media literacy among users by providing them with a reliable tool to verify the credibility of news sources.

3. Enhance Information Trustworthiness: The project aims to contribute to a more reliable and trustworthy information ecosystem. By promoting credible news sources and discouraging the spread of fake news, application seeks to improve the overall quality of information available to users.

4. Contribute to a Safer Online Environment: The project aims to contribute to a safer online environment by reducing the spread of fake news. By detecting and flagging fake news articles, the application seeks to minimize the potential harm caused by misinformation and its impact on individuals and society.

1.3 PURPOSE, SCOPE AND APPLICABILITY

1.3.1 Purpose:

The purpose of the project is to develop an Android application that empowers users to identify and combat the spread of fake news. By leveraging machine learning the application aims to analyse news articles and provide users with insights into the credibility of news sources. The ultimate purpose is to promote media literacy, enhance information trustworthiness, and foster critical thinking among users.

1.3.2 Scope:

The scope of the project includes the development of an Android application with features such as news article analysis, credibility verification, user feedback, and fact-checking integration. The application will provide users with a user-friendly interface to access news articles, analyse their credibility, and receive fact-checking updates. The project also encompasses the implementation of machine learning algorithms and natural language processing techniques to enhance the accuracy of fake news detection.

1.3.3 Applicability:

The application is applicable to a wide range of users, including individuals, journalists, researchers, and educators. It can be used by individuals to verify the credibility of news sources and make informed decisions about the information they consume. Journalists can utilize the application to fact-check news articles and ensure the accuracy of their reporting. Researchers can leverage the application to analyse trends in fake news and develop strategies to combat its spread. Educators can incorporate the application into media literacy programs to teach students critical thinking skills and promote responsible news consumption.

1.4 ACHIEVEMENT

1. Fake News Detection Accuracy: Achieve a high level of accuracy in detecting and flagging fake news articles, ensuring that users can rely on the application to make informed decisions about the credibility of news sources.

2. User Engagement and Adoption: Measure the level of user engagement and adoption of the "Fake News Detector" application, indicating its usefulness and value to users in combating the spread of fake news.

3. Positive User Feedback: Gather positive user feedback and testimonials, demonstrating the effectiveness of the application in helping users identify and avoid fake news.

4. Fact-Checking Integration: Successfully integrate with reputable fact-checking services or APIs to provide users with additional information and verification about the credibility of news articles.

5. Media Literacy Promotion: Contribute to the promotion of media literacy by providing users with tools and insights to critically evaluate news sources and identify potential indicators of fake news.

6. Continuous Improvement: Continuously update and improve the "Fake News Detector" application based on user feedback, technological advancements, and evolving patterns of fake news.

Chapter 2

Survey of Technologies

DESCRIPTION

In the era of information overload and the prevalence of fake news, there is a growing need for reliable sources to verify the authenticity of news. With advancements in technology, chatbots equipped with real-time databases have emerged as a potential solution. A news chatbot that analyses news input by users in real-time can play a crucial role in combating fake news. By leveraging its access to a comprehensive database of verified information, this chatbot can quickly assess the credibility of news articles or claims. This real-time analysis capability empowers users to make informed decisions about the information they consume. By leveraging technology and reliable data sources, this chatbot aims to contribute towards promoting accurate and trustworthy journalism in an increasingly digital world.

1.OIGETIT FAKE NEWS FILTER

DESCRIPTION

Oigetit, deciding the trustworthiness of news articles is no longer an issue! Our mission is to eliminate the time you spend reading fake news. Instead, Oigetit brings the trust back into news articles. In the past, you may have received your news from maybe 8 to 10 news sources network TV, and maybe US and local papers too. But thanks to Oigetit, you can access your news from approximately 1,00,000 news sources.

FEATURES

1. Uses AI and unique algorithm to compute the facts, credibility and trustworthiness of news you read.
2. The app provides free access to articles of approximately 1M news sources.
3. It has both mobile and desktop versions available, across your favourite platforms.

TECHNOLOGY

1.Sentiment-Represents whether the sentiment towards the news is positive, negative or neutral.

2.Reliability-Reliability Score of the source.

3.Real-time-News is scraped from the real-time database.

Source- https://play.google.com/store/apps/details?id=io.scal.oigetit

2.FAKE BANANAS

DESCRIPTION

Fake news detection is based on the concept of stance detection. Fake news is tough to identify. Many 'facts' are highly complex and difficult to check, or exist on a 'continuum of truth' or are compound sentences with fact and fiction overlapping. The best way to attack this problem is not through fact checking, but by comparing how reputable sources feel about a claim.

FEATURES

1.Users input a claim like "The Afghanistan war was bad for the world".

2.Our program will search the thousands of global and local news sources for their 'stance' on that topic.

3.We run sources through our Reputability Algorithm. If lots of reputable sources all agree with your claim, then it's probably true!

4.Then we cite our sources so our users can click through and read more about that topic.

Technology

1. News Sources-After combing through numerous newspaper and natural language processing APIs, discovered that the best way to find related articles is by searching for keywords.
2. Determining Reputation-Using a large set of default sources with hard coded reputability, our database of sources continues to become more accurate with each web scraping by adding new sources and articles.
3. Stance Detection-Stance detection is run by Google's Tensorflow.
4. Frontend/backend-backend is written on a Flask python server which connects to our front-end written in JavaScript.

   Source- https://arxiv.org/pdf/1707.03264v2.pdf

3.SNOPES

DESCRIPTION

When misinformation obscures the truth and readers don’t know what to trust, Snopes’ fact-checking and original investigative reporting lights the way to evidence-based and contextualized analysis. We always link to and document our sources so readers are empowered to do independent research and make up their own minds. Snopes got its start in 1994, investigating urban legends, hoaxes, and folklore. As demand for reliable fact checks grew, so did Snopes. Now it’s the oldest and largest fact-checking site online, widely regarded by journalists, folklorists, and readers as an invaluable research companion.

FEATURES

1.Fact Checks- Rumors and questionable claims we have researched recently.

2.Latest- The latest fact checks and original reporting from Snopes' editorial team.

TECHNOLOGY

1.Methodology- the material we tackle can range from everything to analyzing whether an image has been digitally manipulated to explicating the text of a Congressional bill, we can't describe any single method that applies to all of our fact-checking.

2.Sources -We attempt to use non-partisan information and data sources (e.g., peer-reviewed journals, government agency statistics) as much as possible, and to alert readers that information and data from sources such as political advocacy organizations and partisan think tanks should be regarded with skepticism.

Source- https://www.snopes.com/about/

COMPARATIVE ANALYSIS

Our application stands out from other systems by offering a unique feature - the ability to take input directly from users for news identification. Unlike other systems that rely solely on algorithms and data analysis, our application values the input and insights of users in the news identification process.

By allowing users to provide their own input, we aim to enhance the accuracy and relevance of news identification. Users can contribute their knowledge, opinions, and perspectives, which can be invaluable in distinguishing between reliable and unreliable sources or identifying bias in news articles.

Chapter 3

Requirement and Analysis

3.1 PROBLEM DEFINATION

Current state of the system: The current state of the system is an Android application that provides users with current news articles. However, it lacks the capability to detect and flag fake news. Users are unable to verify the credibility of the news sources or identify potentially misleading or false information.

Expected state of the system : The expected state of the system is to enhance the existing Android application by incorporating a "fake news detector" feature. This feature will enable users to not only access current news articles but also verify the credibility of the sources and identify potential fake news. The application will provide users with a reliable tool to make informed decisions about the information they consume.

GAP analysis : The gap between the current and expected state of the system lies in the absence of a fake news detection feature in the existing Android application. The current state only provides users with current news articles without any mechanism to verify their credibility. The expected state aims to bridge this gap by integrating a "fake news detector" feature that can analyze news articles and help users identify and flag potential fake news.

3.2 REQUIREMENT SPECIFICATION

3.2.1 Functions of our system

1.News Article Retrieval: Fetching and displaying current news articles from reliable sources. Categorizing news articles based on topics such as politics, sports, entertainment, etc.

2.User Authentication and Profile Management: Allowing users to create accounts and log in securely. Managing user profiles, including personal information and preferences.

3.Fake News Detection: Analyzing news articles using machine learning algorithms and natural language processing techniques. Identifying potential indicators of fake news, such as misleading information, biased language, or unreliable sources. Assigning credibility scores or labels to news articles based on the likelihood of them being fake.

4.Credibility Verification: Integrating with fact-checking services or APIs to verify the credibility of news sources and articles. Displaying additional information about the credibility of news articles, such as fact-checking results or source reputation.

5.User Feedback and Reporting: Allowing users to provide feedback on news articles, such as flagging potential fake news or reporting inaccuracies. Implementing a reporting system to handle user feedback and take appropriate actions.

6.User Interface and User Experience: Designing an intuitive and user-friendly interface for easy navigation and interaction. Providing visual indicators or labels to highlight the credibility of news articles. Customizing the user interface based on user preferences and settings.

7.Data Privacy and Security: Implementing secure data storage and transmission protocols to protect user information. Ensuring compliance with data privacy regulations and guidelines.

3.3 SOFTWARE AND HARDWARE SPECIFICATION

3.3.1 SOFTWARE SPECIFICATION

Frontend:

- XML

Backend:

- Java
- Firebase
- Python

3.3.1 HARDWARE SPECIFICATION

Operating System: Windows 10 Pro 64-bit (10.0, Build 19044)

Processor: Intel(R) Core(TM) i5-4690 CPU @ 3.50GHz   3.50 GHz

RAM: 16.0 GB

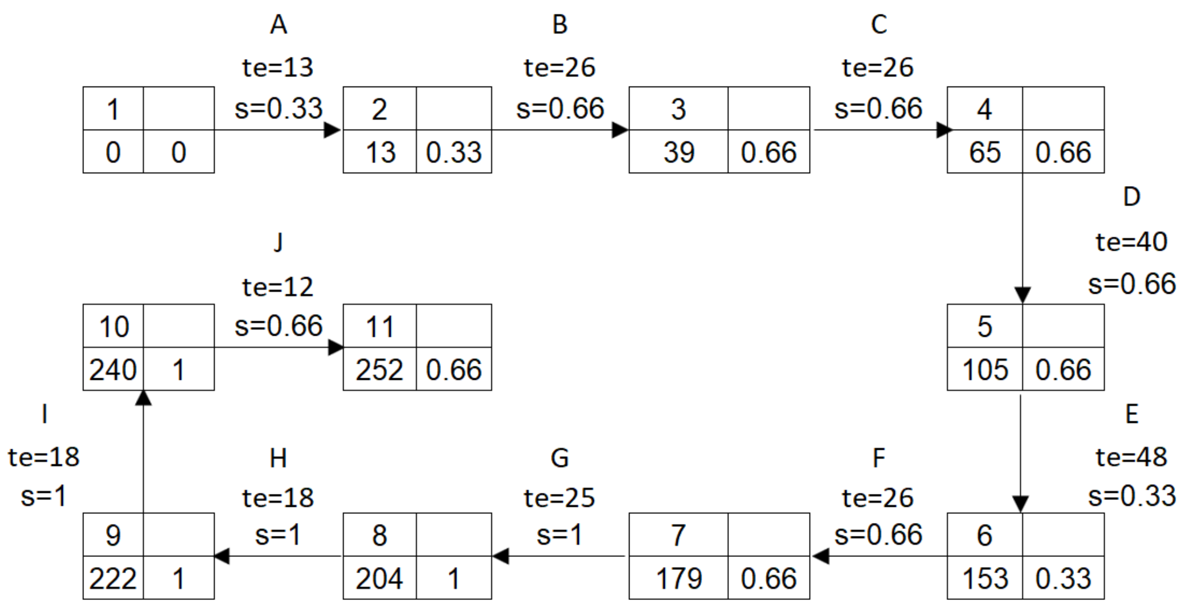
Graphic Card: NVIDIA GeForce GTX 750 Ti (4 GB)

System Type: 64-bit operating system, x64-based processor

3.4 PLANNING AND SCHEDULING

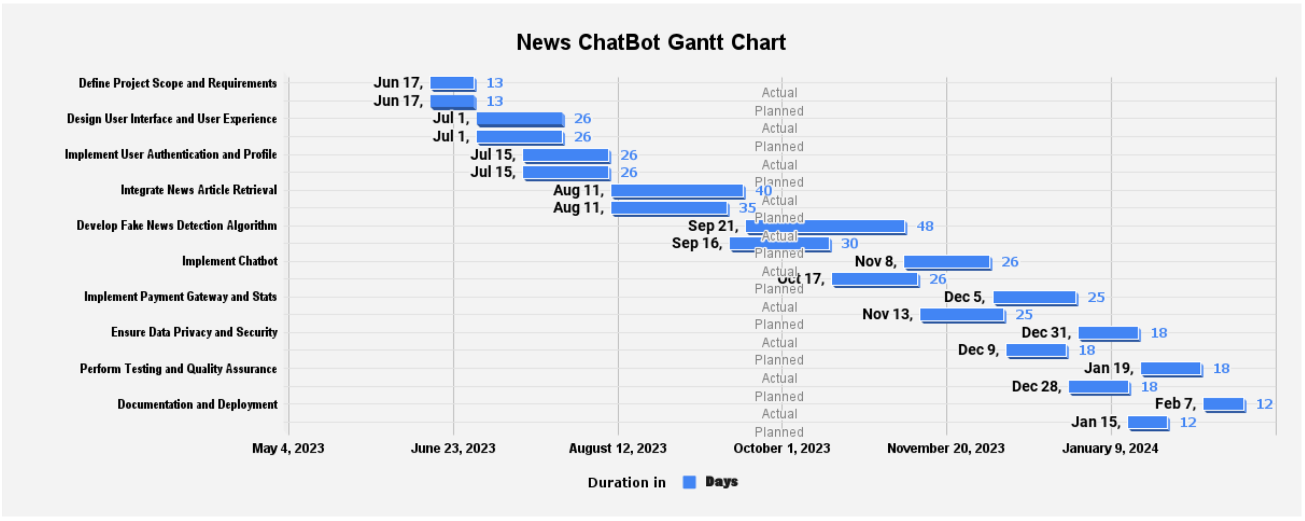
3.4.1 PERT CHART

ID	Activity	Optimistic Time(a)	Most Likely(m)	Pessimistic(b)	Expected Time(te)	Standard Deviation(sd)
A	Define Project Scope and Requirements	12	13	14	13	0.33
B	Design User Interface and User Experience:	24	26	28	26	0.66
C	Implement User Authentication and Profile Management:	24	26	28	26	0.66
D	Integrate News Article Retrieval	38	40	42	40	0.66
E	Develop Fake News Detection Algorithm	47	48	49	48	0.33
F	Implement Chatbot	24	26	28	26	0.66
G	Implement Payment Gateway and Stats	22	25	28	25	1
H	Ensure Data Privacy and Security	15	18	21	18	1
I	Perform Testing and Quality Assurance	15	18	21	18	1
J	Documentation and Deployment	10	12	14	12	0.66
Total		216 Days	252 Days	273 Days		



3.4.2 GANTT CHART

Activity	Start Date		Days
Define Project Scope and Requirements	Actual	17-June	13
	Planned	17-June	13
Design User Interface and User Experience	Actual	1-July	26
	Planned	1-July	26
Implement User Authentication and Profile Management	Actual	15-July	26
	Planned	15-July	26
Integrate News Article Retrieval	Actual	11-August	40
	Planned	11-August	35
Develop Fake News Detection Algorithm	Actual	21-September	48
	Planned	16-September	30
Implement Chatbot	Actual	8-November	26
	Planned	17-October	26
Implement Payment Gateway and Stats	Actual	5-December	25
	Planned	13-November	25
Ensure Data Privacy and Security	Actual	31-December	18
	Planned	9-December	18
Perform Testing and Quality Assurance	Actual	19-January	18
	Planned	28-December	18
Documentation and Deployment	Actual	7-February	12
	Planned	15-January	12



3.5 PRELIMINARY PRODUCT DESCRIPTION

Features:

1. Fake News Detection: Analyze news articles and identify potential indicators of fake news, such as misleading information, biased language, or unreliable sources.

2. Credibility Verification: Integrate with fact-checking services or APIs to verify the credibility of news sources and articles. Provide users with additional information, such as fact-checking results or source reputation, to aid in their assessment.

3. User Feedback and Reporting: Allow users to provide feedback on news articles, flag potential fake news, and report inaccuracies. Implement a reporting system to handle user feedback and take appropriate actions.

4. News Article Retrieval: Fetch and display current news articles from reliable sources, categorized by topics such as politics, sports, entertainment, etc., to keep users informed about the latest events.

5. User Authentication and Profile Management: Enable users to create accounts, log in securely, and manage their profiles. Personalize the user experience based on preferences and settings.

6. User Interface and User Experience: Design an intuitive and user-friendly interface for easy navigation and interaction. Provide visual indicators or labels to highlight the credibility of news articles and enhance the user experience.

Objectives:

1. Empower Users: Provide users with a reliable tool to verify the credibility of news sources and make informed decisions about the information they consume.

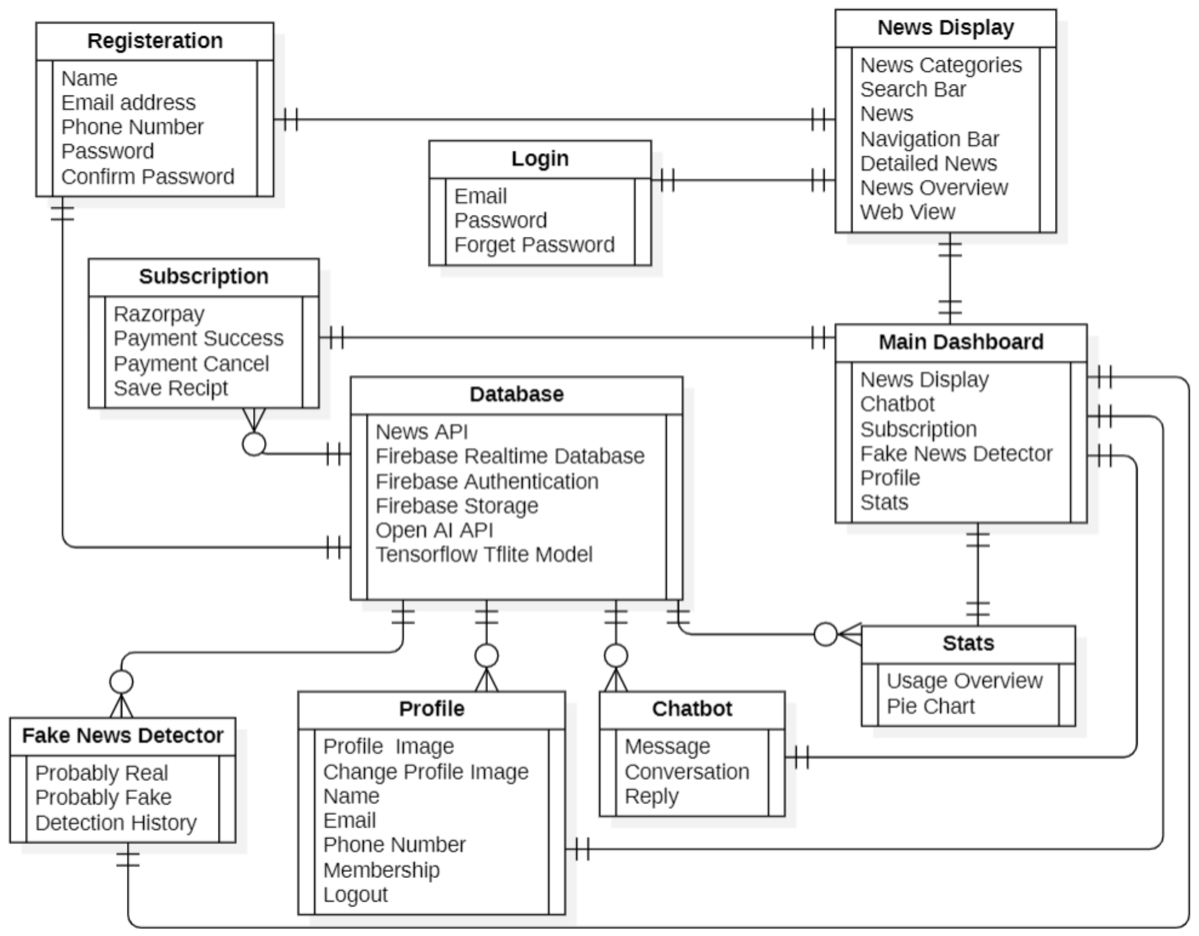
2. Combat Fake News: Detect and flag potential fake news articles, contributing to the reduction of misinformation and its harmful effects.

3. Promote Media Literacy: Educate users about the importance of media literacy and critical thinking when consuming news articles.

4. Enhance User Experience: Design an intuitive and user-friendly interface that ensures a seamless and engaging user experience.

3.6 CONCEPTUAL MODELS

3.6.1 ER DIAGRAM



Data Dictionary:

Firebase_Table

Column Name	Data Type	Constraints
User	varchar(50)	Primary Key

User_Table

Column Name	Data Type	Constraints
UID	varchar(50)	Primary Key

UID_Table

Column Name	Data Type	Constraints
Email	varchar(50)	Primary Key
Name	varchar(50)	Not Null
New Detect Input History	varchar(50)	Not Null
Order Count	int	Not Null
OrderID	int	Not Null
Password	varchar(50)	Not Null
Phone	int	Not Null
Bot_Chat	varchar(50)	Not Null

News_Display_Table

Column Name	Data Type	Constraints
News Title	varchar(50)	Primary key
AuthorID	varchar(50)	Not Null
Content	varchar(50)	Not Null
URL	varchar(50)	Not Null
Date	varchar(50)	Not Null

Bot_Chat_Table

Column Name	Data Type	Constraints
UID	varchar(50)	Primary Key
Message	varchar(50)	Not Null
Send_By	varchar(50)	Not Null

News_Detect_History_Table

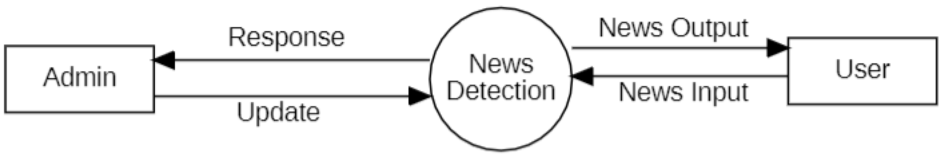
Column Name	Data Type	Constraints
News Detect History	varchar(50)	Primary Key

Payment_Table

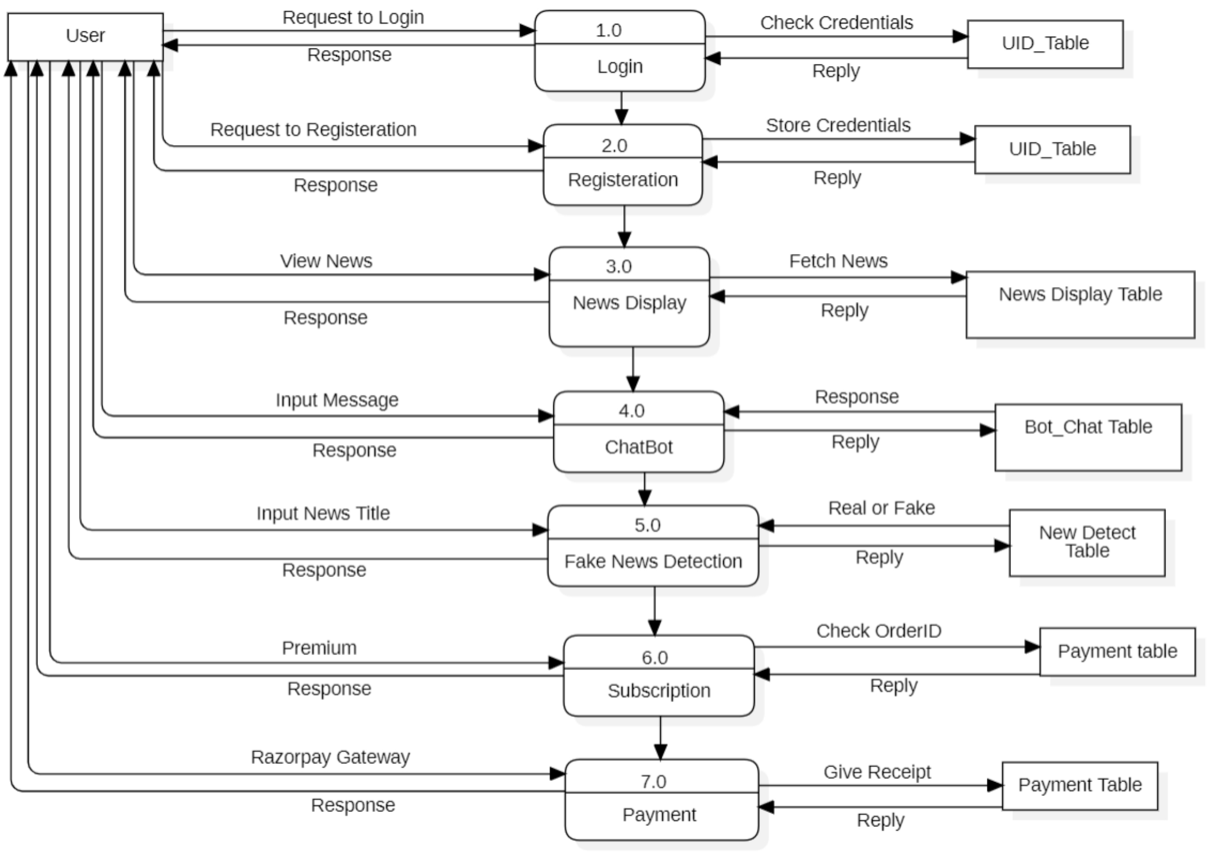
Column Name	Data Type	Constraints
UID	varchar(50)	Primary key
Payment_ID	varchar(50)	Not Null
Amount	varchar(50)	Not Null
Status	varchar(50)	Not Null

3.6.2 DFD DIAGRAM

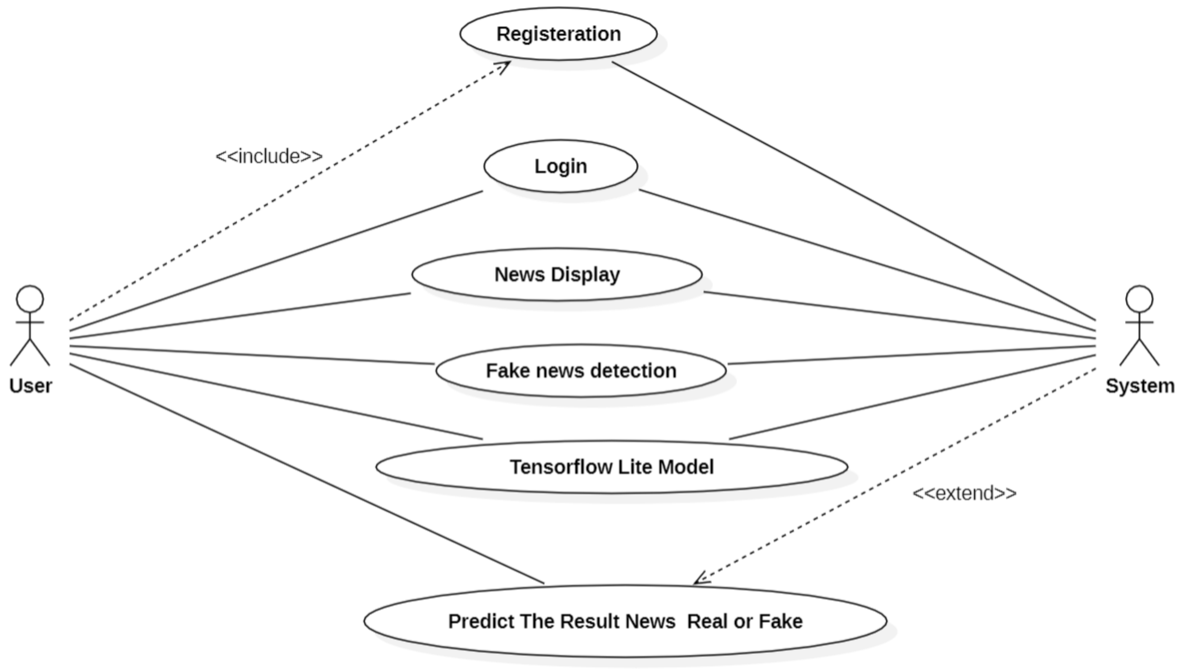
DFD 0



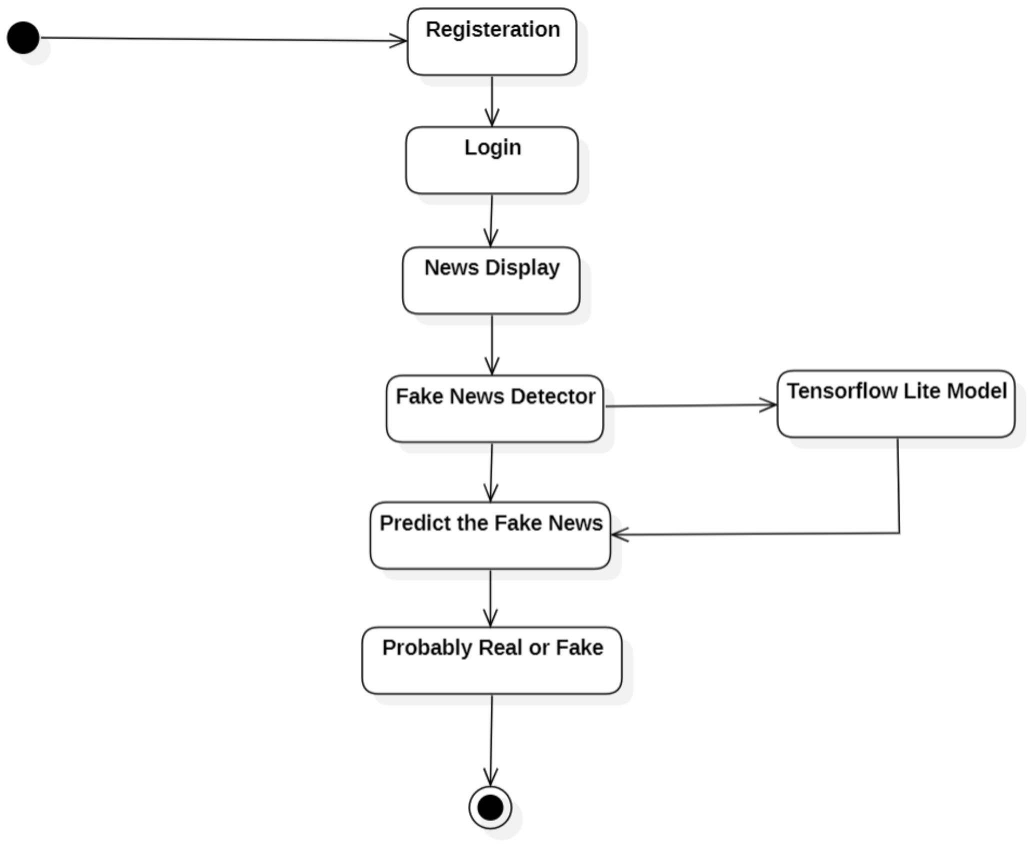
DFD 1



3.6.3 USE CASE DIAGRAM



3.6.4 ACTIVITY DIAGRAM



Chapter 4

System Design

The project we are working on consist of several segments as we know that the entire project can't be developed in one shot, so developing it in several segment and then assembling those makes the work easily and time saving. Through the module development technique one can make out the best use of human resource. The project been developed, it consists of various modules, which will help to build application. It is developed using different modules so that it can be handled properly.

Following are the modules developed:

1. Registration Module.

2. Login Module.

3. News Module.

4. News Detector Module.

5. News Display Module.

4.1 REGISTRATION MODULE:

Registration Module will ask you for your Name, Email Address, Phone No, Password and Confirm Password. User should do registering on your application.

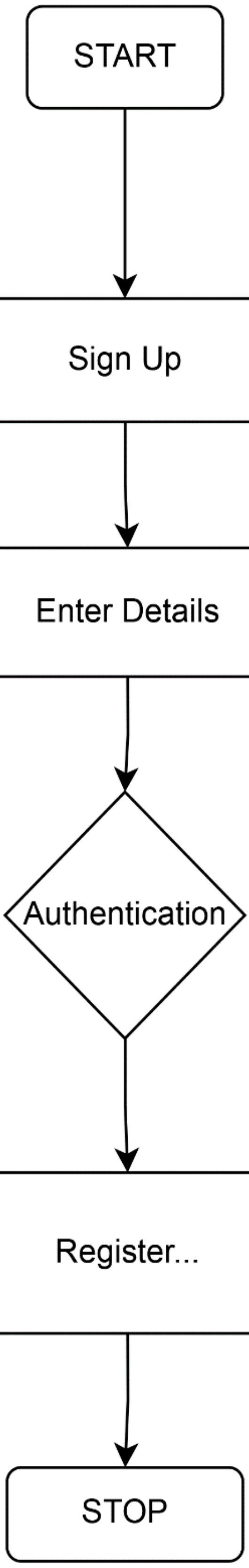
4.1.1 Algorithm:

Step 1: After opening the application user have to sign up to register your account.

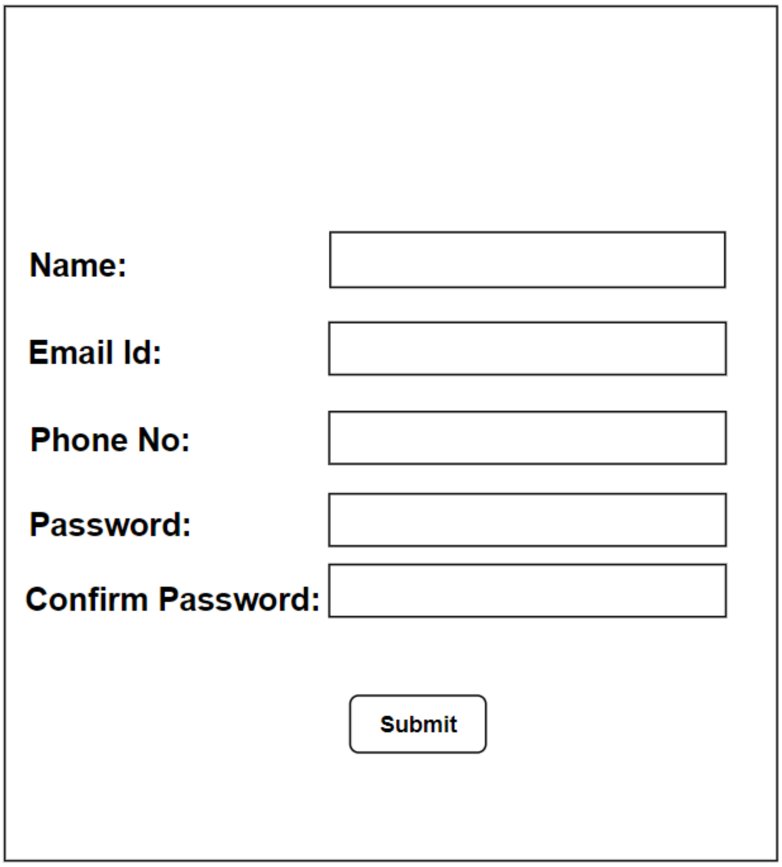
Step 2: User have fill following credentials: Name. Email Address, Phone No, Password and Confirm Password.

Step 3: After clicking on sign up button user will be logged-in our database. Next time user should log-in and directly enter the application.

4.1.2 Flowchart:



4.1.3 Interface:



4.2 LOGIN MODULE:

Login Module will contain Name and Password Authentication to enter your account on application.

4.2.1 Algorithm:

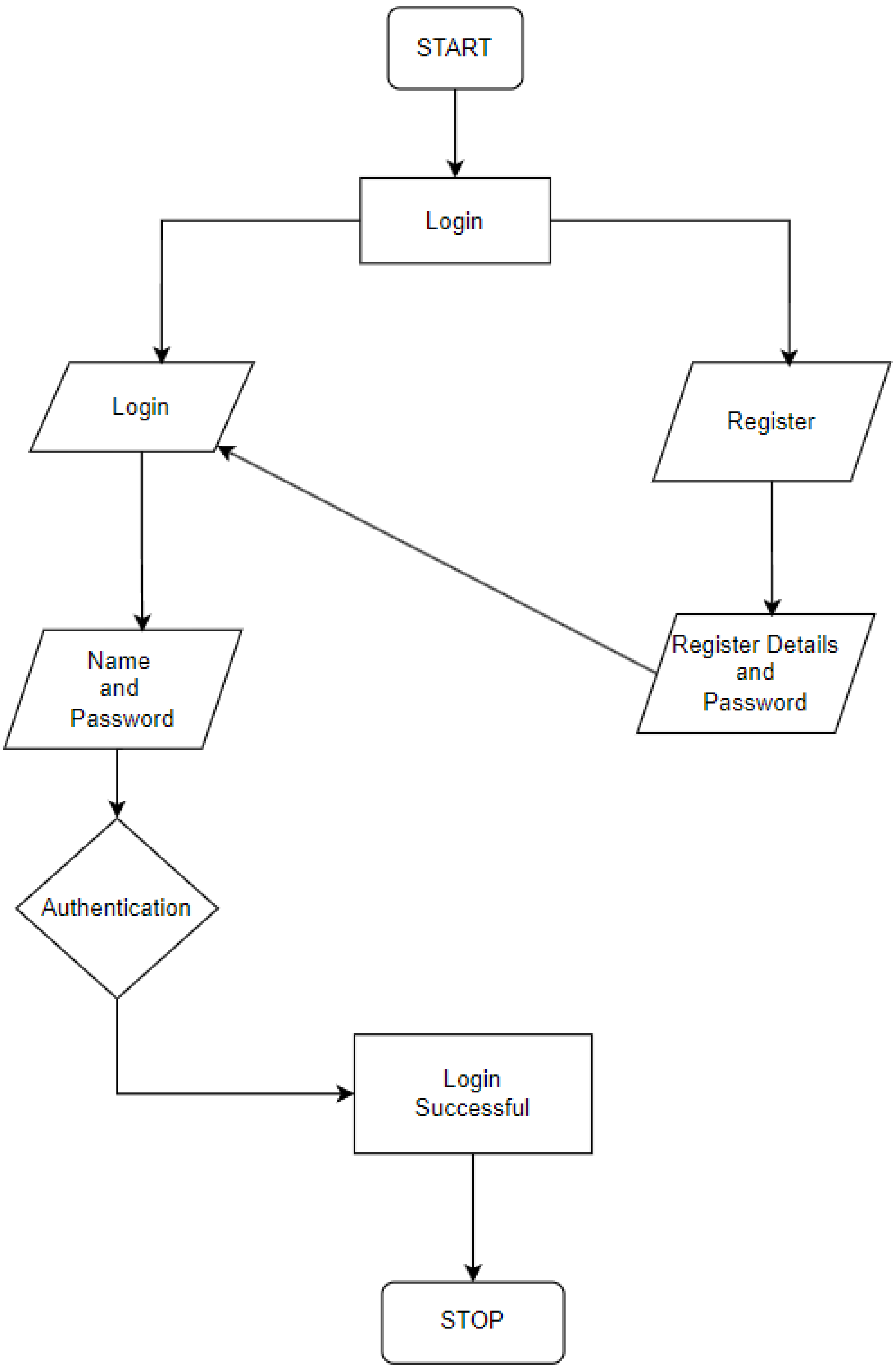
Step 1: User should go to the application, have to go to login button

Step 2: User to login the application.

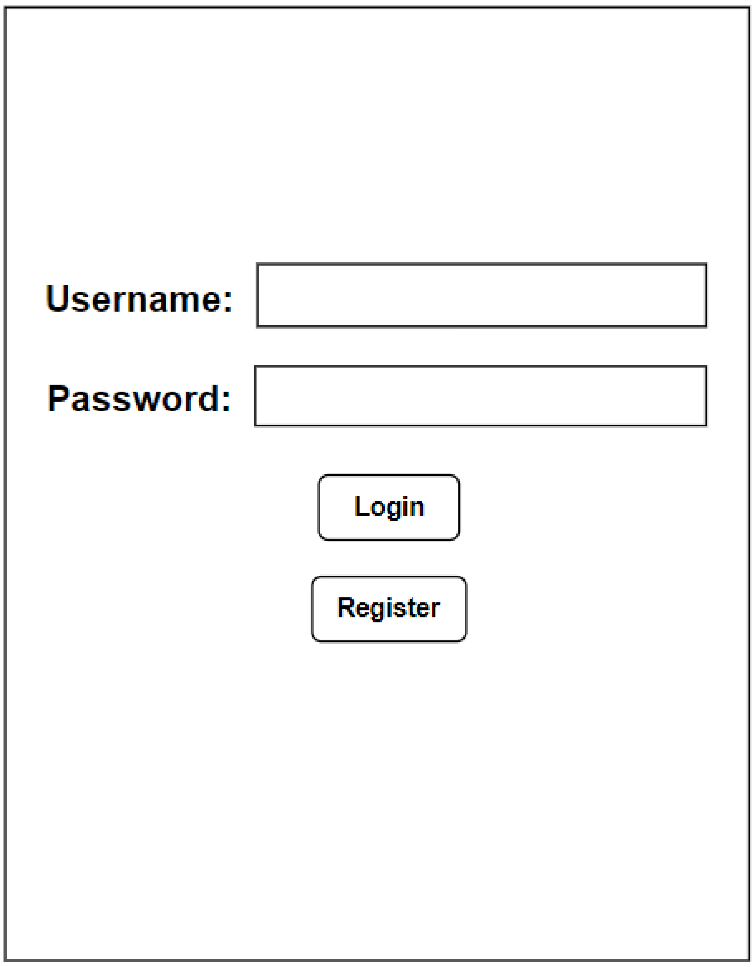
Step 3: Then the user will be asked Name and Password, which he has to enter to get logged in.

Step 4: If user is not signed in already should go to sign up option page to get signed in. Step 5: After registering or signing in the suer will go to main dashboard.

4.2.2 Flowchart:



4.2.3 Interface:



4.3 NEWS MODULE:

The module will have a user-friendly interface with intuitive navigation for seamless news browsing which will have news's id, url, author, title, description, publish date and content.

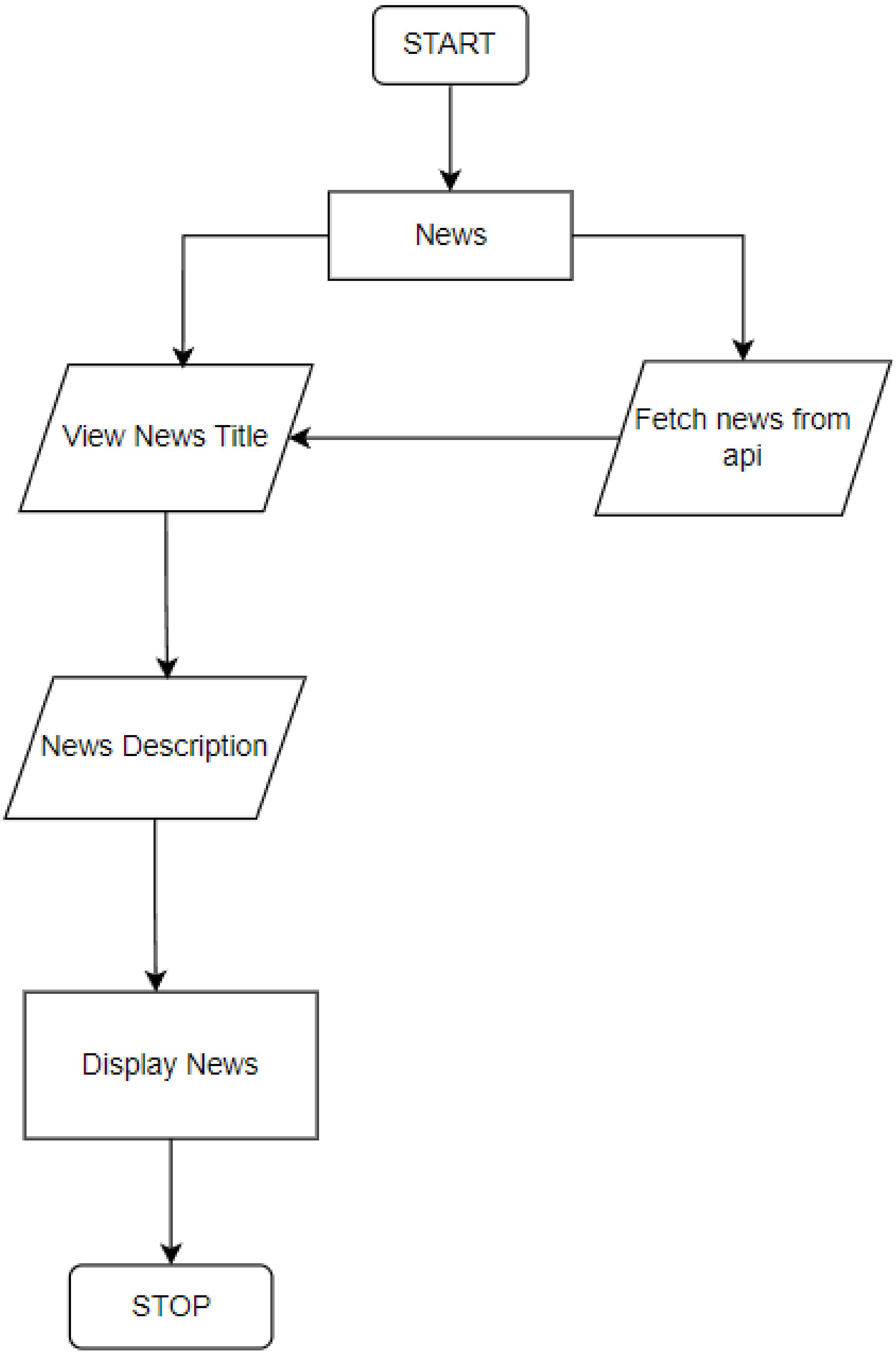
4.3.1 Algorithm:

Step 1: User can view the main dashboard of the application.

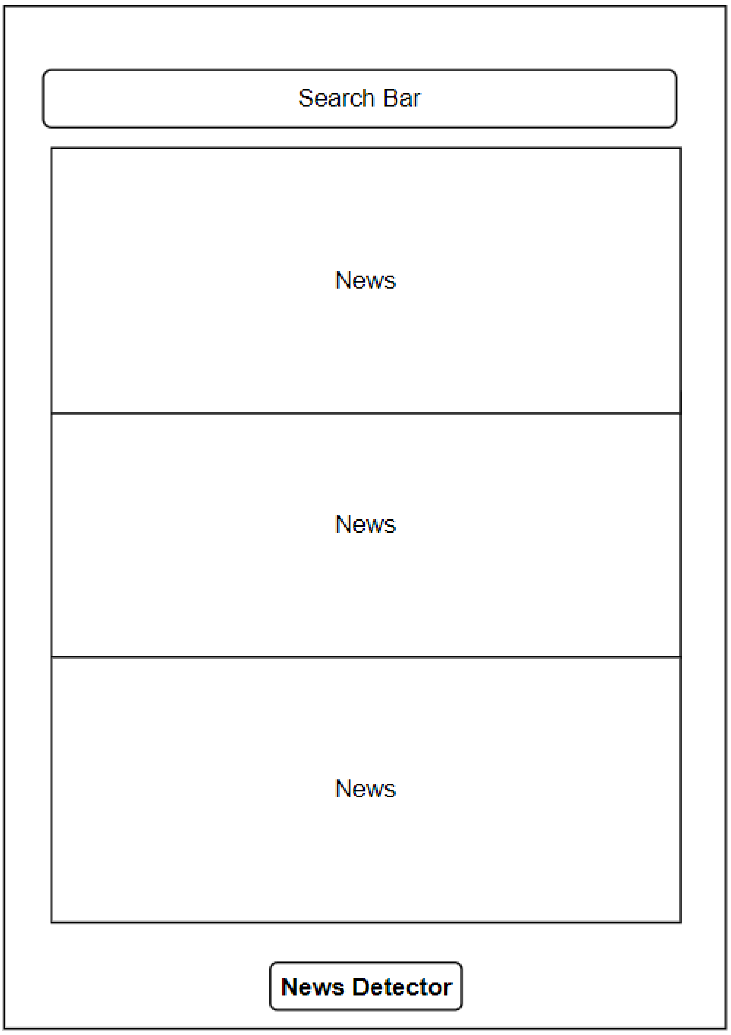
Step 2: User can browse the real-time news on the application which displays news's id, url, author, title, description, publish date and content.

Step 3: By clicking the news title, user can display the original source of the news.

4.3.2 Flowchart:



4.3.3 Interface:



4.4 NEWS DETECTOR MODULE:

The news module will employ natural language processing and machine learning techniques to analyse and classify news articles as reliable or potentially misleading/fake.

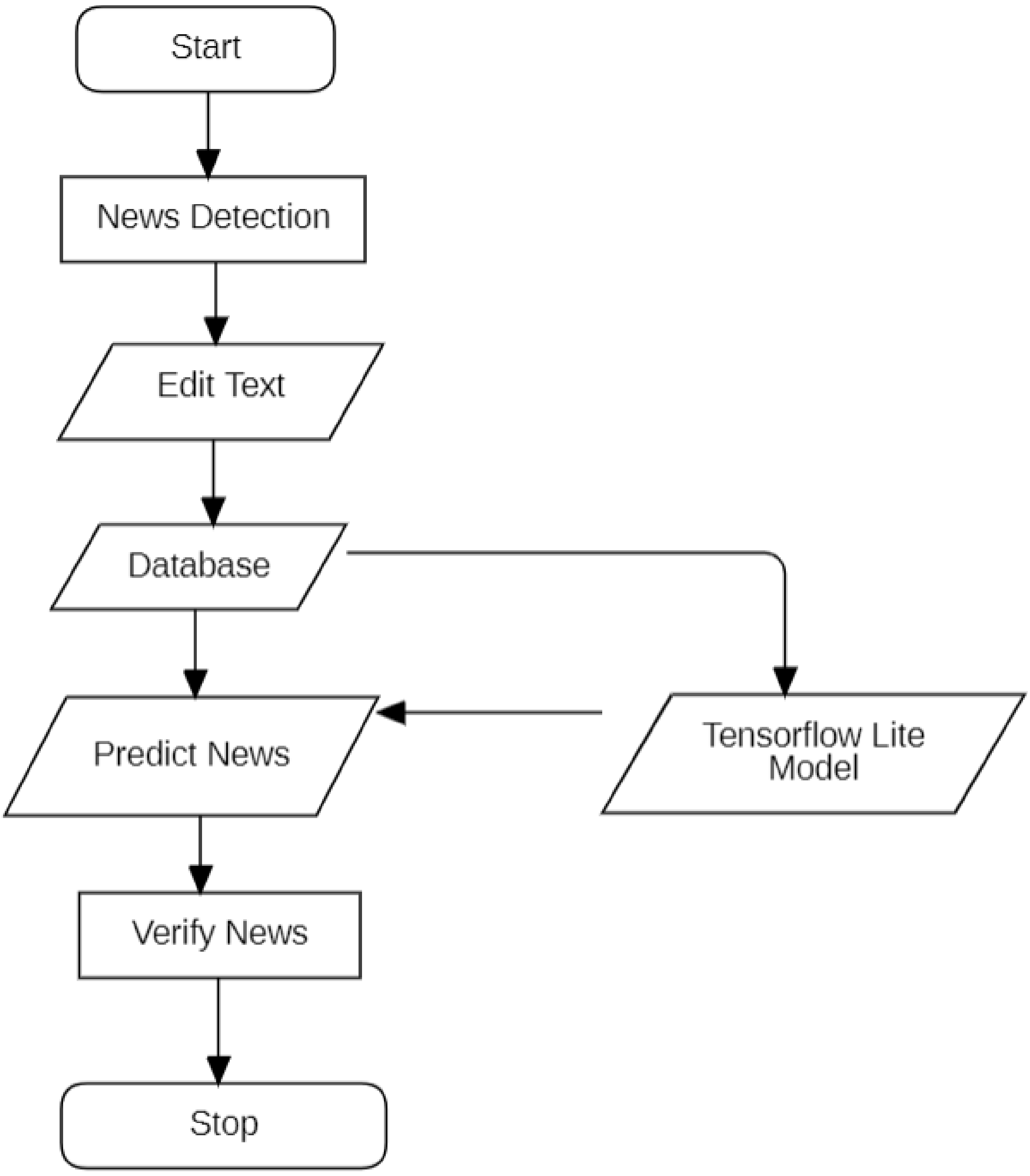
4.4.1 Algorithm:

Step 1: User will type the news he/she wants to verify.

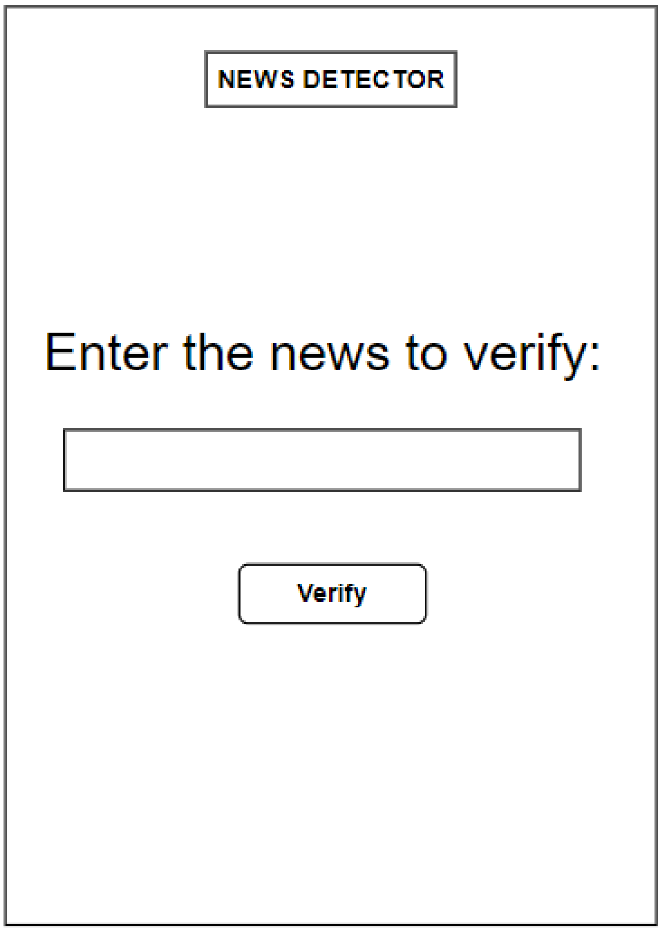
Step 2: The news will then be transferred to the database will do natural language processing.

Step 3: Then the news will be analysed by machine learning(logistic regression) from the trained news model.

4.4.2 Flowchart:



4.4.3 Interface:



4.5 DISPLAY MODULE:

The display module will present the news probably true/false in a clean and user-friendly interface and also providing the news source if true.

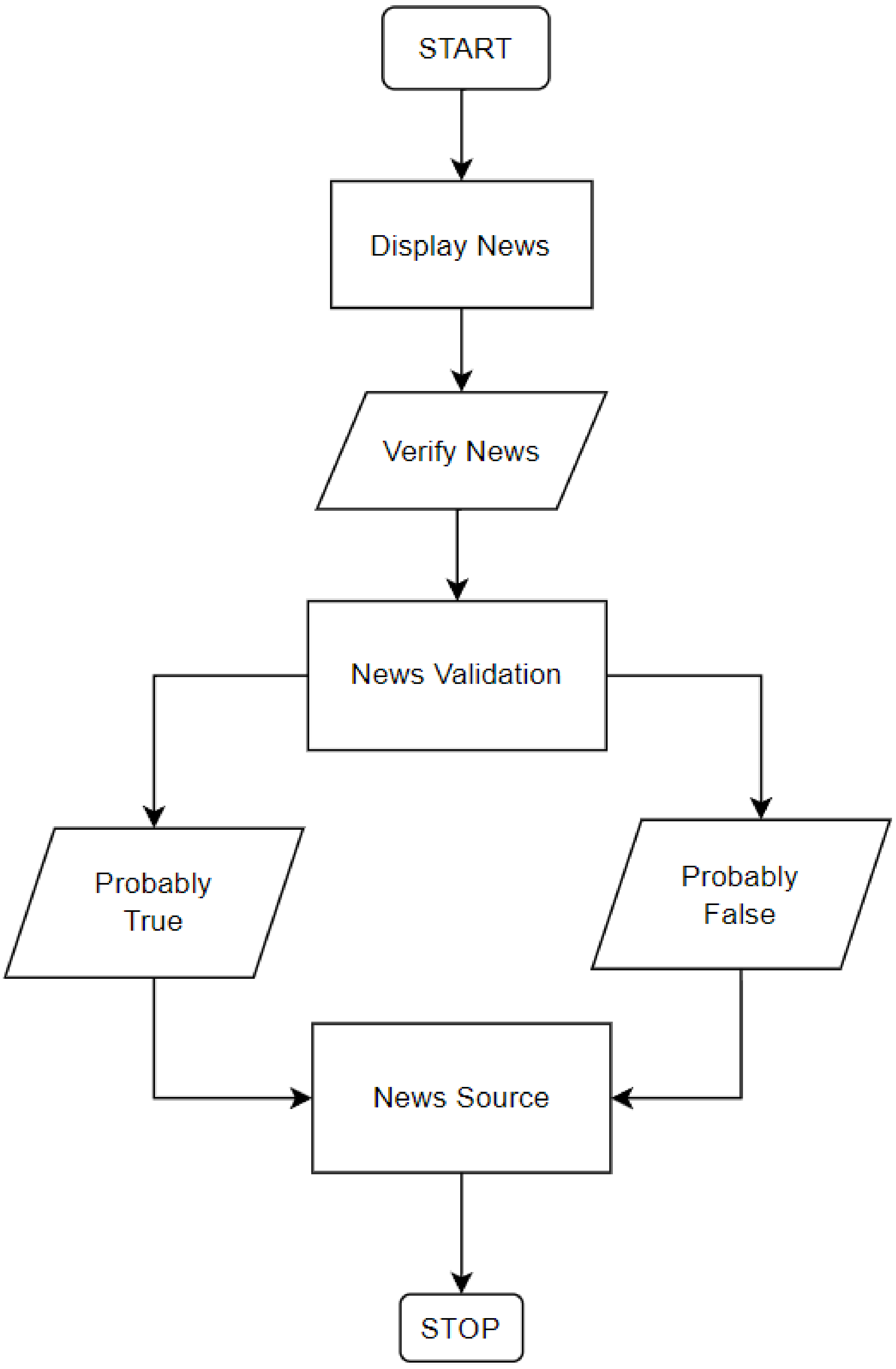
4.5.1 Algorithm:

Step 1: User will get the verification of the news he typed in detector module.

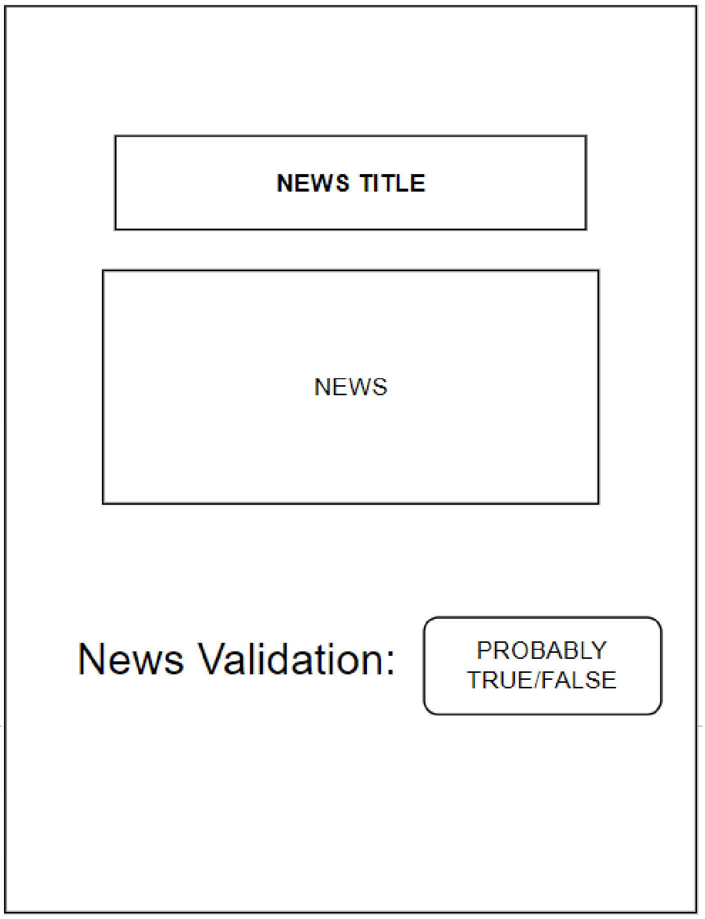
Step 2: The module will show whether the news is probably true or probably false.

Step 3: User will also get a source button for the news to visit the actual source.

4.5.2 Flowchart:



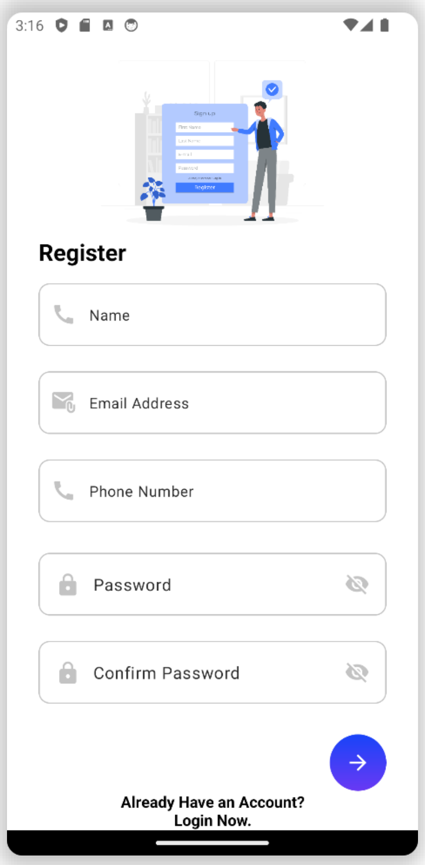
4.5.3 Interface:

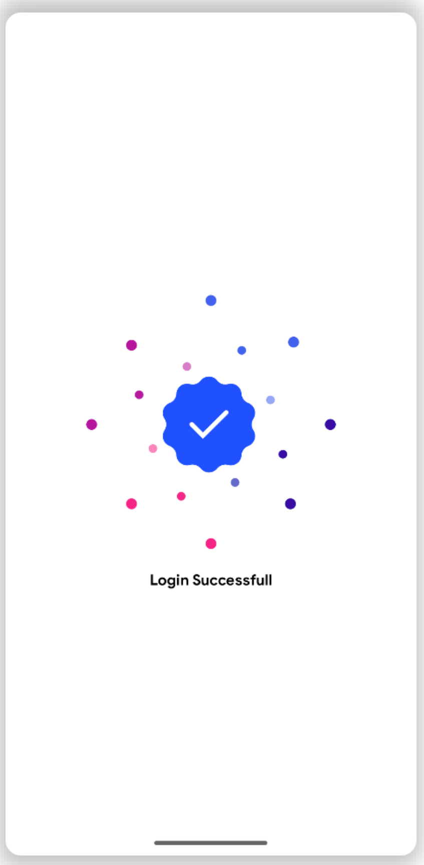


Chapter 5

Implementation

5.1 Registration:

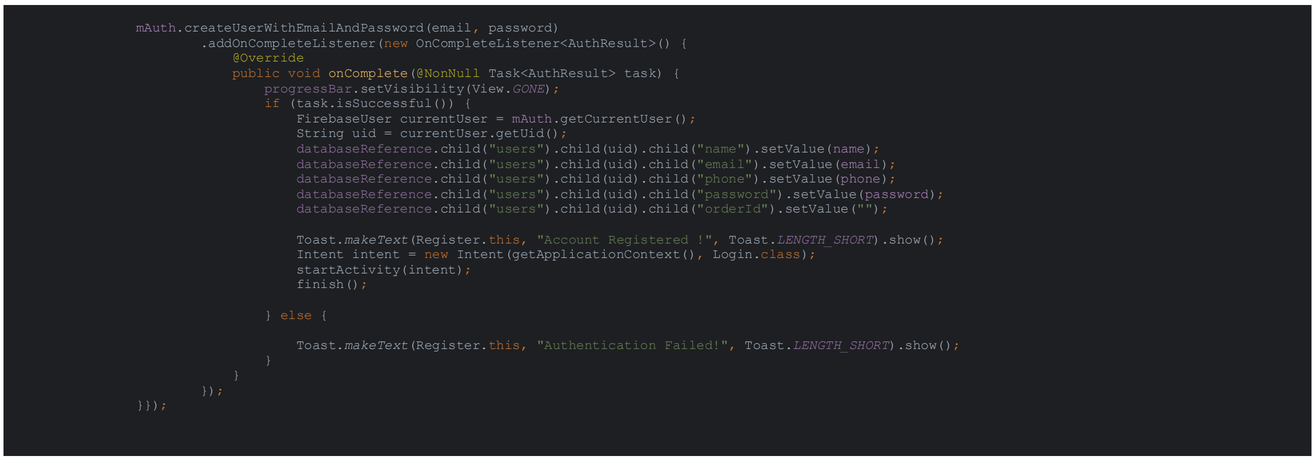




Description:

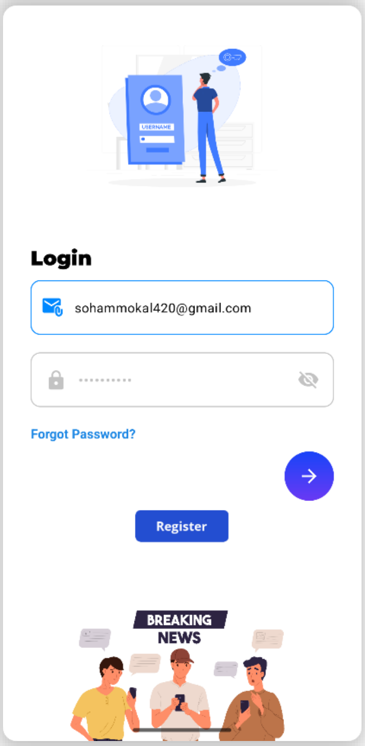
The Register page allows users to create a new account by providing necessary details such as Name, Email, Phone number and Password. Upon successful registration, users gain access to the app's features and functionalities.

Code:-



```
mAuth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                progressBar.setVisibility(View.GONE);
                if (task.isSuccessful()) {
                    FirebaseUser currentUser = mAuth.getCurrentUser();
                    String uid = currentUser.getUid();
                    databaseReference.child("users").child(uid).child("name").setValue(name);
                    databaseReference.child("users").child(uid).child("email").setValue(email);
                    databaseReference.child("users").child(uid).child("phone").setValue(phone);
                    databaseReference.child("users").child(uid).child("password").setValue(password);
                    databaseReference.child("users").child(uid).child("orderId").setValue("");

                    Toast.makeText(Register.this, "Account Registered !", Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(getApplicationContext(), Login.class);
                    startActivity(intent);
                    finish();

                } else {

                    Toast.makeText(Register.this, "Authentication Failed!", Toast.LENGTH_SHORT).show();
                }
            }
        });
}});
```

5.2 Login:

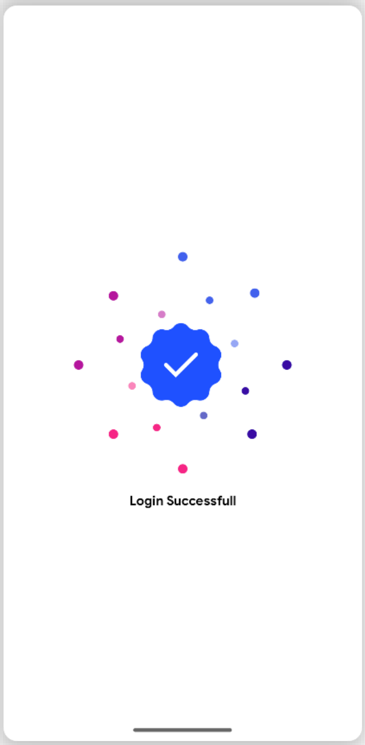

Login

sohammokal420@gmail.com

••••••••••

Forgot Password?

Register


Login Successfull

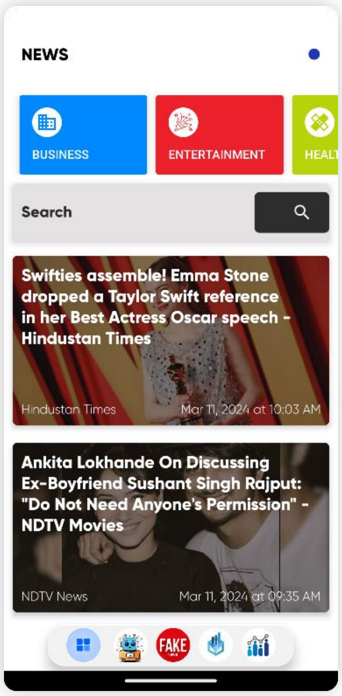
Description:

The Login page enables existing users to securely access their accounts by entering their credentials, typically a email and password combination. Upon successful authentication, users gain entry to the app's features and personalized content.

Code:

```
                //firebase authentication
                 mAuth.signInWithEmailAndPassword(email, password)
                              .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                                  @Override
                                  public void onComplete(@NonNull Task<AuthResult> task) {
                                      progressBar.setVisibility(View.GONE);
                                      if (task.isSuccessful()) {
                                          Toast.makeText(getApplicationContext(), "Login Successfull !",
Toast.LENGTH_SHORT).show();
                                          Intent intent = new Intent(getApplicationContext(),
SuccessfulLoginAnim.class);
                                          startActivity(intent);
                                          finish();

                                      } else {
                                          Toast.makeText(Login.this, "Login failed !", Toast.LENGTH_SHORT).show();
                                      }
                                  }
                              });
                 }

        });

    }
```

5.3 News Display:

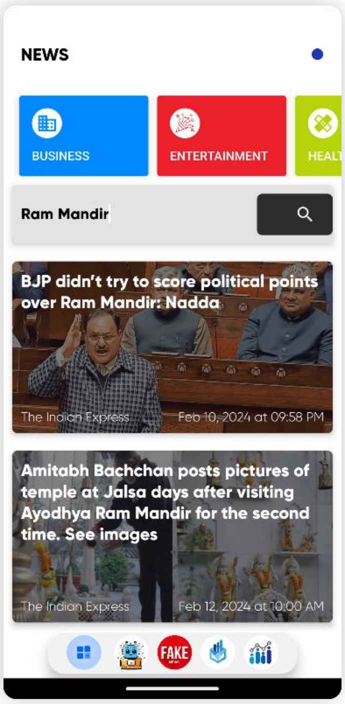

NEWS

Swifties assemble! Emma Stone dropped a Taylor Swift reference in her Best Actress Oscar speech - Hindustan Times

Hindustan Times Mar 11, 2024 at 10:03 AM

Ankita Lokhande On Discussing Ex-Boyfriend Sushant Singh Rajput: "Do Not Need Anyone's Permission" - NDTV Movies

NDTV News Mar 11, 2024 at 09:35 AM

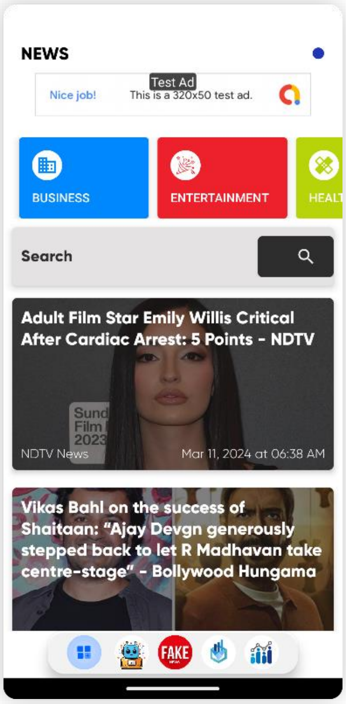

NEWS

BJP didn't try to score political points over Ram Mandir: Nadda

The Indian Express Feb 10, 2024 at 09:58 PM

Amitabh Bachchan posts pictures of temple at Jalsa days after visiting Ayodhya Ram Mandir for the second time. See images

The Indian Express Feb 12, 2024 at 10:00 AM


NEWS

Test Ad
Nice job! This is a 320x50 test ad.

Adult Film Star Emily Willis Critical After Cardiac Arrest: 5 Points - NDTV

NDTV News Mar 11, 2024 at 06:38 AM

Vikas Bahl on the success of Shaitaan: "Ajay Devgn generously stepped back to let R Madhavan take centre-stage" - Bollywood Hungama

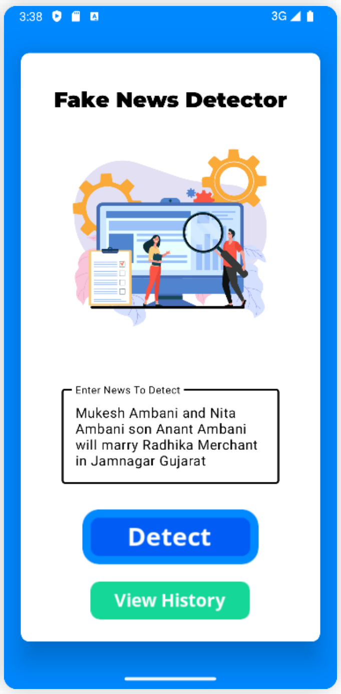
Description:

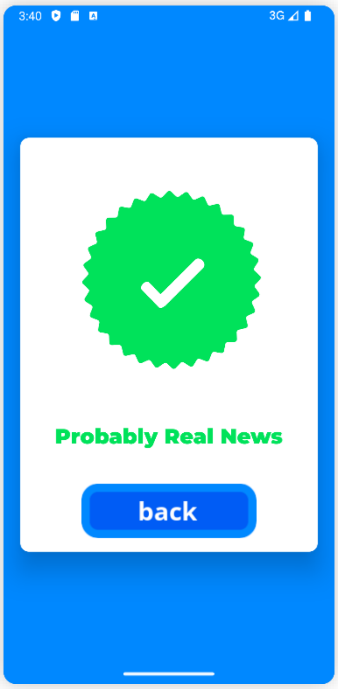
The News Display feature presents curated articles and updates in an intuitive interface, ensuring easy access to relevant information. With a user-friendly design, it offers a seamless browsing experience, allowing users to stay informed effortlessly.

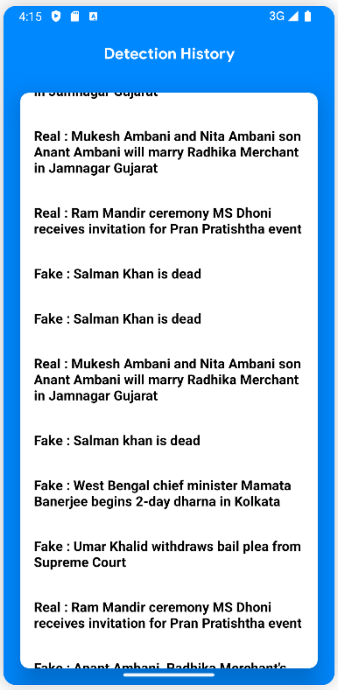
Code:

```java
public void retrieveJson(String query, String category, String apiKey) {
    swipeRefreshLayout.setRefreshing(true);
    String country = getCountry();

    Call<Headlines> call;
    if (!query.equals("")) {
        call = ApiClient.getInstance().getApi().getSpecificData(query, apiKey);
    } else {
        call = ApiClient.getInstance().getApi().getCategoryNews(country, category, apiKey);
    }

    call.enqueue(new Callback<Headlines>() {
        @Override
        public void onResponse(Call<Headlines> call, Response<Headlines> response) {
            swipeRefreshLayout.setRefreshing(false);
            if (response.isSuccessful() && response.body() != null) {
                List<Articles> newArticles = response.body().getArticles();
                updateRecyclerView(newArticles);
            } else {
                Toast.makeText(NewsDisplay.this, "Failed to retrieve data", Toast.LENGTH_SHORT).show();
            }
        }

        @Override
        public void onFailure(Call<Headlines> call, Throwable t) {
            swipeRefreshLayout.setRefreshing(false);
            Toast.makeText(NewsDisplay.this, t.getLocalizedMessage(), Toast.LENGTH_SHORT).show();
        }
    });
}
```

5.4 Fake News Detection:


3:38
3G
Fake News Detector
Enter News To Detect
Mukesh Ambani and Nita Ambani son Anant Ambani will marry Radhika Merchant in Jamnagar Gujarat
Detect
View History





in Jamnagar Gujarat

Real : Mukesh Ambani and Nita Ambani son Anant Ambani will marry Radhika Merchant in Jamnagar Gujarat

Real : Ram Mandir ceremony MS Dhoni receives invitation for Pran Pratishtha event

Fake : Salman Khan is dead

Fake : Salman Khan is dead

Real : Mukesh Ambani and Nita Ambani son Anant Ambani will marry Radhika Merchant in Jamnagar Gujarat

Fake : Salman khan is dead

Fake : West Bengal chief minister Mamata Banerjee begins 2-day dharna in Kolkata

Fake : Umar Khalid withdraws bail plea from Supreme Court

Real : Ram Mandir ceremony MS Dhoni receives invitation for Pran Pratishtha event

Fake : Anant Ambani, Radhika Merchant's

Description:

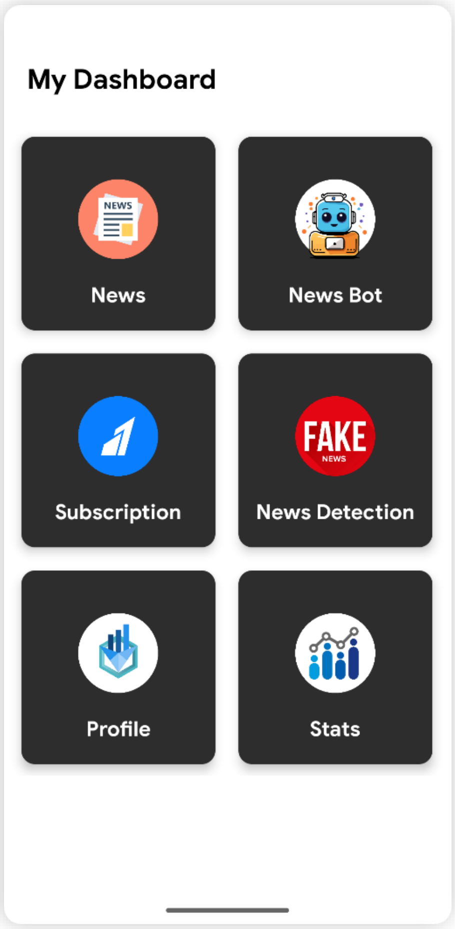
Powered by a TensorFlow model, our feature evaluates news authenticity using advanced deep learning techniques. It analyzes textual content to determine the probability of fake news, helping users identify misinformation efficiently.

Code:

```
        this.detect.setOnClickListener(new OnClickListener() {
            public void onClick(View view) {
                // Get the text from the EditText
                String editTextValue = FakeorRealActivity2.this.newstext.getText().toString();
                String fake = "Fake : "+ editTextValue ;
                String real = "Real : "+ editTextValue ;
                // Set the value of the EditText in the generated key
                if (!FakeorRealActivity2.this.newstext.getText().toString().isEmpty()) {
                    FakeorRealActivity2 fakeorRealActivity2 = FakeorRealActivity2.this;
                    fakeorRealActivity2.stringToPython = fakeorRealActivity2.newstext.getText().toString();
                    fakeorRealActivity2 = FakeorRealActivity2.this;
                    if (((double) fakeorRealActivity2.doInference(fakeorRealActivity2.stringToPython)) > 0.5d) {
                        //fake news
                        DatabaseReference inputHistoryRef =
mDatabase.child("users").child(uid).child("news_detect_input_history").push();
                        inputHistoryRef.setValue(fake);
                        Intent intent = new Intent(getApplicationContext(), DatabaseCheckingAnimFake.class);
                        startActivity(intent);
                        finish();
                    } else {
                        //real news
                        DatabaseReference inputHistoryRef =
mDatabase.child("users").child(uid).child("news_detect_input_history").push();
                        inputHistoryRef.setValue(real);
                        Intent intent = new Intent(getApplicationContext(), DatabaseCheckingAnimReal.class);
                        startActivity(intent);
                        finish();
                    }
                }
            }
        });
    }
    /* Access modifiers changed, original: protected */
    public void onStart() {
        super.onStart();
        readcsvraw();
        this.detect.setVisibility(View.VISIBLE);
```

```java
    }
    private void readcsvraw() {
        String str = "ReadCSVraw";
        Log.d(str, "Reaching here");
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(getResources().openRawResource(R.raw.tokenizer_nlp), StandardCharsets.UTF_8));
        while (true) {
            try {
                String readLine = bufferedReader.readLine();
                if (readLine == null) {
                    break;
                }
                String[] split = readLine.split(",");
                this.map2.put(split[0], Integer.valueOf(Integer.parseInt(split[1])));
                System.out.println(this.map2.get(split[0]));
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        Log.d(str, String.valueOf(this.map2.get("african")));
    }

    private void splittextandint(String str) {
        System.out.println(str.replaceAll("[^0-9]", ""));
    }

    private float doInference(String str) {
        int i;
        Class cls = float.class;
        String[] split = this.newstext.getText().toString().toLowerCase().split("[^\\D]+");
        System.out.println(Arrays.toString(split));
        String str2 = "DoInference";
        Log.d(str2, "Reaching here");
        System.out.println(split.length);
        System.out.println(split[0]);
        split = split[0].split("\\s+");
        for (i = 0; i < split.length; i++) {
            split[i] = split[i].replaceAll("[^\\w]", "");
        }
        System.out.println(split.length);
        for (i = 0; i < this.maxlen; i++) {
            if (i >= split.length) {
                this.data[i] = 0.0f;
            } else if (this.map2.get(split[i]) != null) {
                this.data[i] = (float) Integer.parseInt(String.valueOf(this.map2.get(split[i])));
                Log.d(str2, "Reaching here in IF");
                System.out.println(split[i]);
            } else {
                this.data[i] = 0.0f;
            }
        }
        System.out.println(Arrays.toString(this.data));
        float[][] fArr = new float[1][this.maxlen];
        for (i = 0; i < this.maxlen; i++) {
            fArr[0][i] = this.data[i];
            PrintStream printStream = System.out;
            StringBuilder stringBuilder = new StringBuilder();
            stringBuilder.append("itemTF = ");
            stringBuilder.append(fArr[0][i]);
            printStream.println(stringBuilder.toString());
        }
        float[][] fArr2 = (float[][]) Array.newInstance(cls, new int[]{1, 1});
        this.tflite.run(fArr, fArr2);
        return fArr2[0][0];
    }

    private MappedByteBuffer loadModelFile() throws IOException {
        AssetFileDescriptor openFd = getAssets().openFd("model_whatsapp_fakenews_500.tflite");
        return new FileInputStream(openFd.getFileDescriptor()).getChannel().map(MapMode.READ_ONLY,
openFd.getStartOffset(), openFd.getDeclaredLength());
    }
    private boolean isInternetAvailable() {
        ConnectivityManager connectivityManager = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
        if (connectivityManager != null) {
            NetworkInfo activeNetworkInfo = connectivityManager.getActiveNetworkInfo();
            return activeNetworkInfo != null && activeNetworkInfo.isConnected();
        }
        return false;
    }
    @Override
    public void onBackPressed() {
        Intent intent = new Intent(getApplicationContext(), Chat.class);
        startActivity(intent);
        finish();

    }
```

5.5 Main Dashboard:



Description:

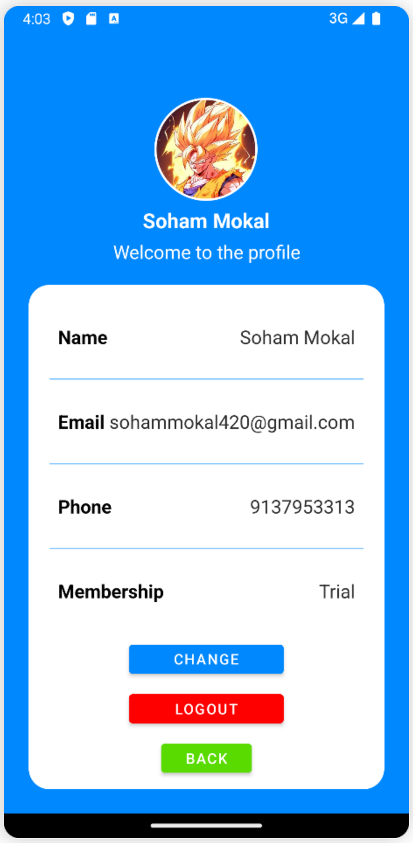
The Main Dashboard offers a comprehensive user experience with access to News display, Chatbot, Subscription, News detection, Profile customization, and Statistical insights. Users can seamlessly navigate between these features, staying informed, engaged, and empowered within the app.

Code:

```
public class Chat extends AppCompatActivity {

    FirebaseUser user;
    FirebaseAuth auth;
    FirebaseAuth mAuth;


    CardView stats,fake ,bot,news,rzp,profile;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

getWindow().setFlags(WindowManager.LayoutParams.FLAG_LAYOUT_NO_LIMITS,WindowManager.LayoutParams.FLAG
_LAYOUT_NO_LIMITS);
        setContentView(R.layout.activity_chat);
        auth= FirebaseAuth.getInstance();
        user= auth.getCurrentUser();
        bot=findViewById(R.id.bot);
```

```java
        news=findViewById(R.id.news);
        rzp=findViewById(R.id.rzp);
        fake=findViewById(R.id.fake);
        profile=findViewById(R.id.profile);
        stats=findViewById(R.id.stats);

        stats.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), StatisticalHistory.class);
                startActivity(intent);
                finish();

            }
        });
        profile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), ProfilePage.class);
                startActivity(intent);
                finish();

            }
        });
        fake.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), FakeorRealActivity2.class);
                startActivity(intent);
                finish();

            }
        });
        bot.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                startActivity(intent);
                finish();

            }
        });
        rzp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), PaymentActivity.class);
                startActivity(intent);
                finish();

            }
        });
        news.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), NewsDisplay.class);
                startActivity(intent);
                finish();

            }
        });



    }


}
```

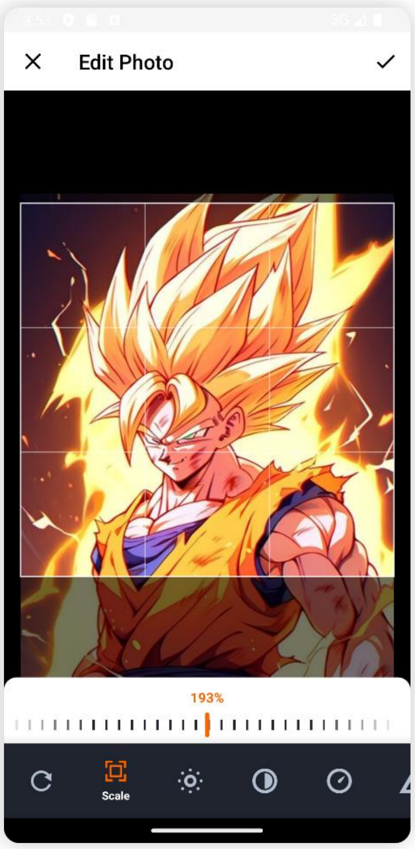
5.6 Profile:


4:03 3G

Soham Mokal

Welcome to the profile

Name	Soham Mokal
Email	sohammokal420@gmail.com
Phone	9137953313
Membership	Trial



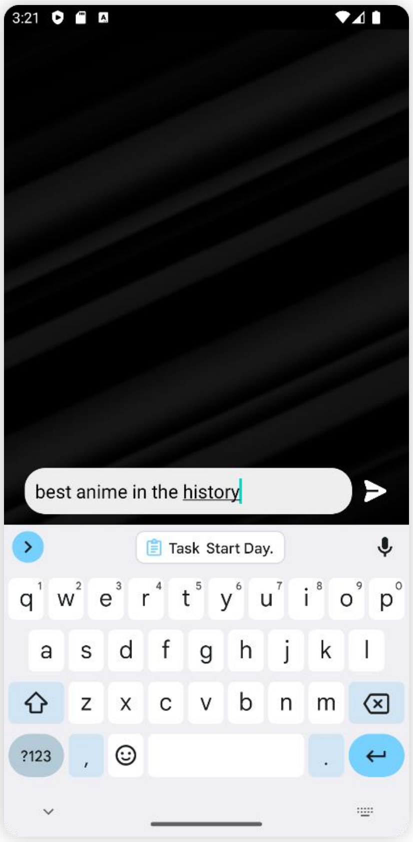
Description:

The Profile Page showcases user details and offers the ability to customize the profile picture. Users can view their personal information and effortlessly update their profile picture, enhancing personalization within the app.

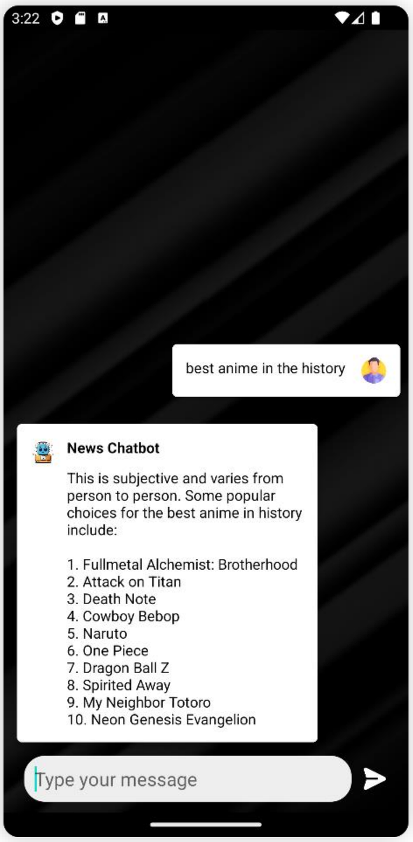
Code:

```
private void getAndSetProfileEmail() {
    mAuth = FirebaseAuth.getInstance();
    FirebaseUser currentUser = mAuth.getCurrentUser();
    String email = currentUser.getEmail();
    this.profileEmail.setText(email);
}

private void getAndSetProfilePhone() {
    mAuth = FirebaseAuth.getInstance();
    FirebaseUser currentUser = mAuth.getCurrentUser();
    String uid = currentUser.getUid();
    mDatabase = FirebaseDatabase.getInstance().getReference("users");

    mDatabase.child(uid).child("phone").get().addOnCompleteListener(task -> {
        if (task.isSuccessful()) {
            String profilePhone = task.getResult().getValue(String.class);
            if (profileName != null) {
                this.profilePhone.setText(profilePhone);
            }
        } else {
            // Handle the error
        }
    });
}

private void getAndSetProfileMembership() {
    mAuth = FirebaseAuth.getInstance();
    FirebaseUser currentUser = mAuth.getCurrentUser();
    String uid = currentUser.getUid();
    mDatabase = FirebaseDatabase.getInstance().getReference("users");

    mDatabase.child(uid).child("orderId").get().addOnCompleteListener(task -> {
        if (task.isSuccessful()) {
            String orderId = task.getResult().getValue(String.class);

            if (orderId != null && !orderId.isEmpty()) {
                this.profileMembership.setText("Premium");
            } else {
                this.profileMembership.setText("Trial");
            }
        } else {
            // Handle the error
        }
    });
}

private void logoutUser() {
    mAuth = FirebaseAuth.getInstance();
    mAuth.signOut();

    Intent intent = new Intent(ProfilePage.this, Login.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent);
    finish();
}

private void uploadProfilePicture(Uri croppedUri) {
    mAuth = FirebaseAuth.getInstance();
    FirebaseUser currentUser = mAuth.getCurrentUser();
    String uid = currentUser.getUid();

    StorageReference profilePicRef = storageReference.child("profile_pictures/" + uid +
".jpg");

    profilePicRef.putFile(croppedUri)
            .addOnSuccessListener(taskSnapshot -> {
                loadProfilePicture();
            })
            .addOnFailureListener(e -> {
                Toast.makeText(ProfilePage.this, "Failed to upload profile picture",
Toast.LENGTH_SHORT).show();
            });
}

private void loadProfilePicture() {
    mAuth = FirebaseAuth.getInstance();
    FirebaseUser currentUser = mAuth.getCurrentUser();
    String uid = currentUser.getUid();

    StorageReference profilePicRef = storageReference.child("profile_pictures/" + uid +
".jpg");

    profilePicRef.getDownloadUrl().addOnSuccessListener(uri -> {
        Picasso.get().load(uri).into(profileImg);
    }).addOnFailureListener(e -> {
        // Handle the error
    });
}
```

5.7 Chat Bot:



best anime in the history



best anime in the history

News Chatbot

This is subjective and varies from person to person. Some popular choices for the best anime in history include:

1. Fullmetal Alchemist: Brotherhood
2. Attack on Titan
3. Death Note
4. Cowboy Bebop
5. Naruto
6. One Piece
7. Dragon Ball Z
8. Spirited Away
9. My Neighbor Totoro
10. Neon Genesis Evangelion

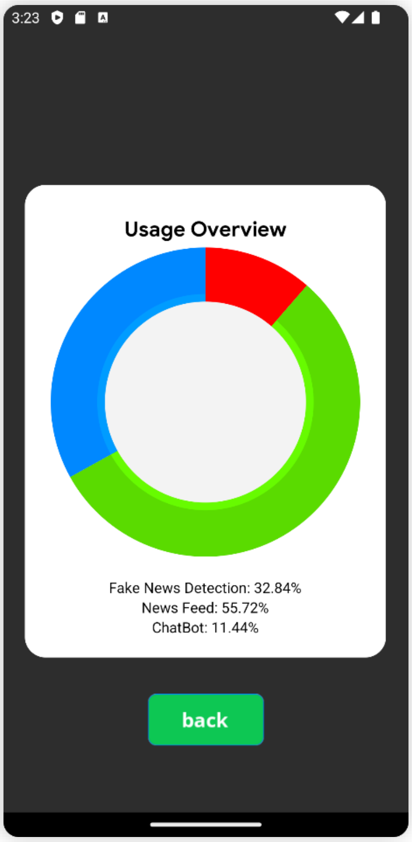
Description:

The Chatbot feature utilizes an API key to access external services, enhancing its functionality by integrating with external platforms or services. This enables the Chatbot to provide users with real-time information, personalized recommendations, and interactive assistance, enhancing the overall user experience within the app.

Code:

```
void callAPI(String question){
    // okhttp
    messageList.add(new Message("Typing...", Message.SEND_BY_BOT));

    JSONObject jsonBody = new JSONObject();
    try {
        jsonBody.put("model","gpt-3.5-turbo-instruct");
        jsonBody.put("prompt", question);
        jsonBody.put("max_tokens",4000);
        jsonBody.put("temperature",0);
    } catch (JSONException e) {
        throw new RuntimeException(e);
    }

    RequestBody requestBody = RequestBody.create(JSON, jsonBody.toString());
    Request request = new Request.Builder()
            .url(API.API_URL)
            .header("Authorization","Bearer "+API.API)
            .post(requestBody)
            .build();
    client.newCall(request).enqueue(new Callback() {
        @Override
        public void onFailure(@NonNull Call call, @NonNull IOException e) {
            addResponse("Failed to load response due to"+e.getMessage());
        }

        @Override
        public void onResponse(@NonNull Call call, @NonNull Response response)
throws IOException {
            if (response.isSuccessful()){
                JSONObject jsonObject = null;
                try {
                    jsonObject = new JSONObject(response.body().string());
                    JSONArray jsonArray = jsonObject.getJSONArray("choices");
                    String result = jsonArray.getJSONObject(0).getString("text");
                    addResponse(result.trim());
                } catch (JSONException e) {
                    throw new RuntimeException(e);
                }
            } else {
                addResponse("Failed to load response due
to"+response.body().toString());
            }

        }
    });

} // callAPI End Here =============
void saveMessageToFirebase(String message, String sendBy) {
    // Check if userChatRef is not null (user is authenticated)
    if (userChatRef != null) {
        // Create a unique key for each message
        String messageId = userChatRef.push().getKey();

        // Create a Message object
        Message firebaseMessage = new Message(message, sendBy);

        // Save the message to Firebase using the unique key under the user's
path
        userChatRef.child(messageId).setValue(firebaseMessage);
    }
}
```

5.8 Stats:

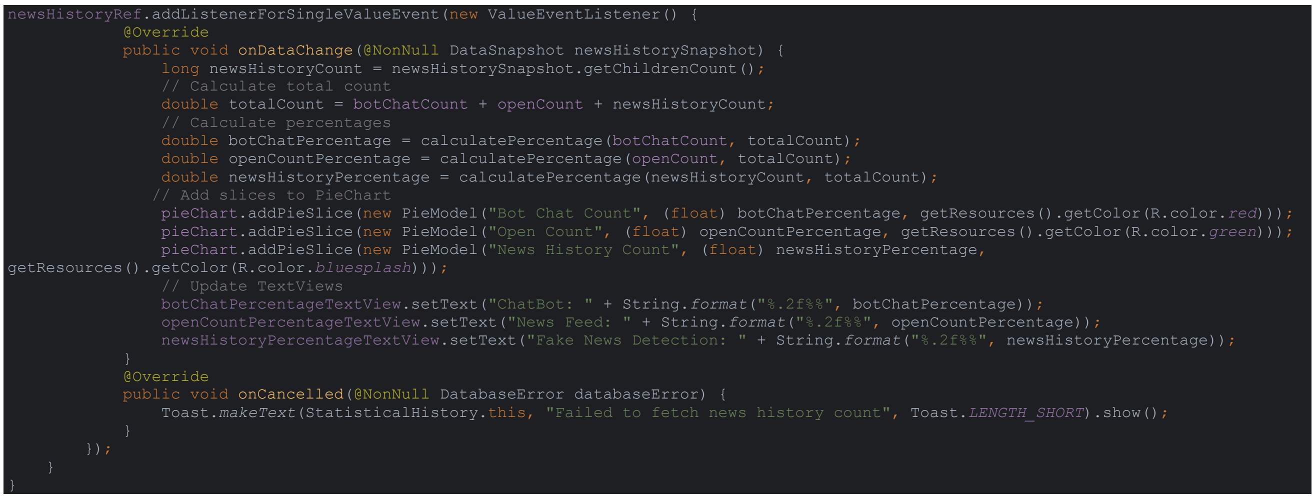


Fake News Detection: 32.84%
News Feed: 55.72%
ChatBot: 11.44%

Description:

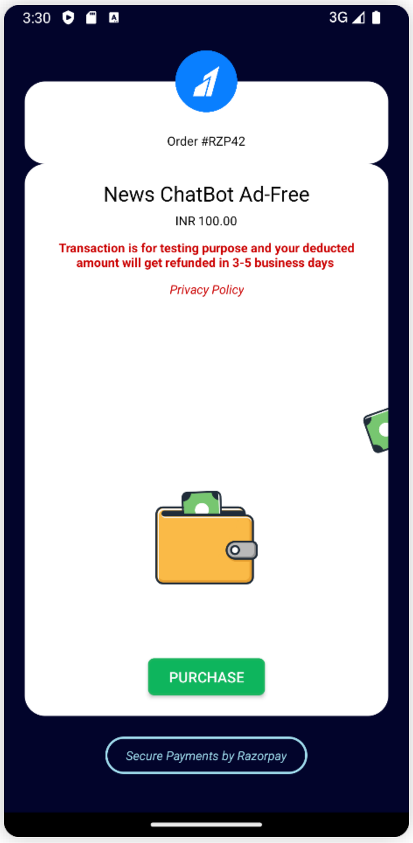
The Stats feature visualizes app functionality usage through interactive pie charts, offering users a clear overview of their engagement. Each segment represents a different app functionality, such as news reading, chatbot interactions and news detection. This intuitive presentation allows users to easily grasp their usage patterns and make informed decisions about their app experience.

Code:



```java
newsHistoryRef.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot newsHistorySnapshot) {
                long newsHistoryCount = newsHistorySnapshot.getChildrenCount();
                // Calculate total count
                double totalCount = botChatCount + openCount + newsHistoryCount;
                // Calculate percentages
                double botChatPercentage = calculatePercentage(botChatCount, totalCount);
                double openCountPercentage = calculatePercentage(openCount, totalCount);
                double newsHistoryPercentage = calculatePercentage(newsHistoryCount, totalCount);
               // Add slices to PieChart
                pieChart.addPieSlice(new PieModel("Bot Chat Count", (float) botChatPercentage, getResources().getColor(R.color.red)));
                pieChart.addPieSlice(new PieModel("Open Count", (float) openCountPercentage, getResources().getColor(R.color.green)));
                pieChart.addPieSlice(new PieModel("News History Count", (float) newsHistoryPercentage,
getResources().getColor(R.color.bluesplash)));
                // Update TextViews
                botChatPercentageTextView.setText("ChatBot: " + String.format("%.2f%%", botChatPercentage));
                openCountPercentageTextView.setText("News Feed: " + String.format("%.2f%%", openCountPercentage));
                newsHistoryPercentageTextView.setText("Fake News Detection: " + String.format("%.2f%%", newsHistoryPercentage));
            }
            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
                Toast.makeText(StatisticalHistory.this, "Failed to fetch news history count", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

5.9 Subscription:

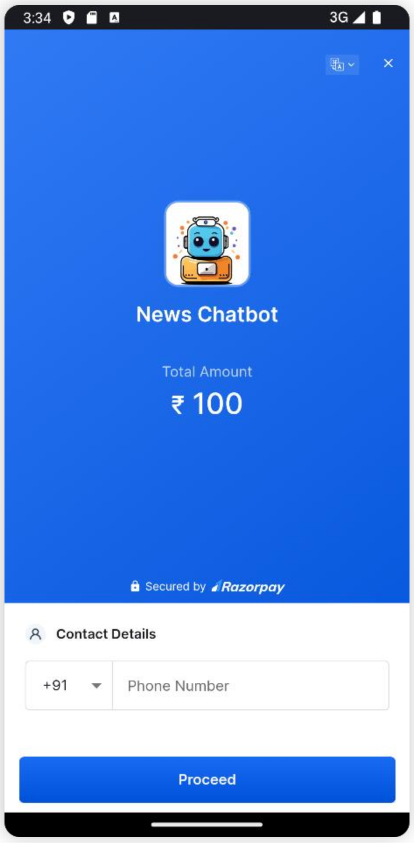

3:30 3G

Order #RZP42

News ChatBot Ad-Free

INR 100.00

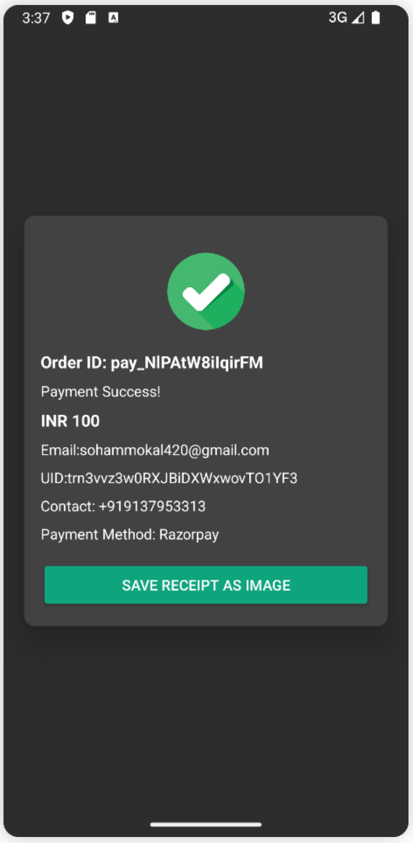
Transaction is for testing purpose and your deducted amount will get refunded in 3-5 business days


3:34 3G

News Chatbot

Total Amount

₹ 100


3:37 3G

Order ID: pay_NlPAtW8iIqirFM

Payment Success!

INR 100

Email:sohammokal420@gmail.com

UID:trn3vvz3w0RXJBiDXWxwovTO1YF3

Contact: +919137953313

Payment Method: Razorpay

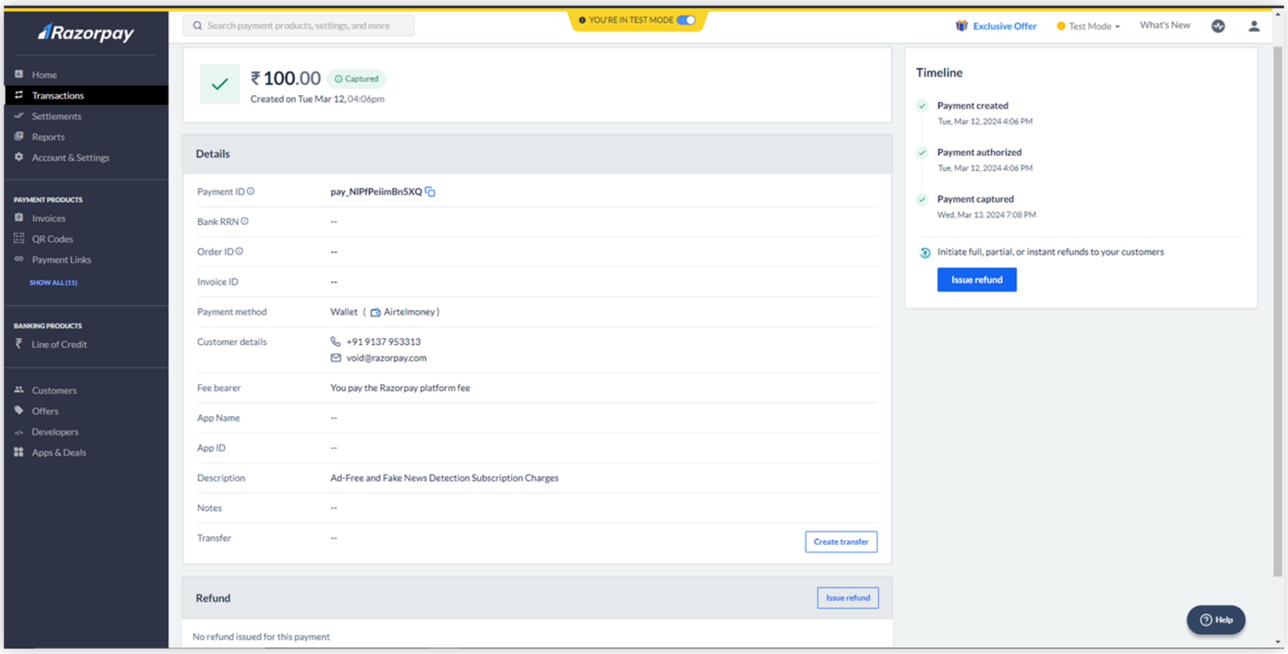
Description:

The Subscription Razor Pay Gateway facilitates seamless management of subscriptions within the app, providing users with an advertisement-free experience and unlimited trials of news and chatbot functionalities. Leveraging Razor Pay's secure payment gateway, users can easily subscribe to access premium features, ensuring uninterrupted usage and enhanced engagement.

Code:

```
public void startPayment() {
    /*
      You need to pass current activity in order to let Razorpay create CheckoutActivity
     */
    final Activity activity = this;

    final Checkout co = new Checkout();
    String etApiKey ="rzp_test_gUj1OOI3Bxhddr";
    co.setKeyID(etApiKey);

    try {
        JSONObject options = new JSONObject();
        options.put("name", "News Chatbot");
        options.put("description", "Ad-Free and Fake News Detection Subscription
Charges");
        options.put("send_sms_hash",true);
        options.put("allow_rotation", true);
        //You can omit the image option to fetch the image from dashboard
        options.put("image", "https://i.postimg.cc/prNpgy3V/newscb.png");
        options.put("currency", "INR");
        options.put("amount", "10000");
        co.open(activity, options);
    } catch (Exception e) {
        Toast.makeText(activity, "Error in payment: " + e.getMessage(),
Toast.LENGTH_SHORT)
                .show();
        e.printStackTrace();
    }
}
@Override
public void onPaymentSuccess(String s, PaymentData paymentData) {
    try{


        alertDialogBuilder.setMessage("Payment Successful :\nPayment ID: "+s+"\nPayment
Data: "+paymentData.getData());
        alertDialogBuilder.show();
        String paymentId = paymentData.getPaymentId();
        String name = paymentData.getUserContact();
        String data = String.valueOf(paymentData.getData());
        String email = paymentData.getUserEmail();



        // Create an intent with the payment details
        Intent intent = new Intent(this, PaymentReceipt.class);
        intent.putExtra("paymentId", paymentId);
        intent.putExtra("orderId", s);
        intent.putExtra("name", name);
        intent.putExtra("email", email);

        intent.putExtra("data", data);


        // Start the PaymentReceipt activity
        startActivity(intent);
    }catch (Exception e){
        e.printStackTrace();
    }
}

@Override
public void onPaymentError(int i, String s, PaymentData paymentData) {
    try{

        String paymentId = paymentData.getPaymentId();
        String name = paymentData.getUserContact();
        String data = String.valueOf(paymentData.getData());
        String email = paymentData.getUserEmail();



        // Create an intent with the payment details
        Intent intent = new Intent(this, PaymentFailed.class);
        intent.putExtra("paymentId", paymentId);
        intent.putExtra("orderId", s);
        intent.putExtra("name", name);
        intent.putExtra("email", email);

        intent.putExtra("data", data);


        // Start the PaymentReceipt activity
        startActivity(intent);
    }catch (Exception e){
        e.printStackTrace();
    }
}
```

5.10 Admin Side Report:



₹ 100.00 Captured
Created on Tue Mar 12, 04:06pm

Details

Payment ID	pay_NlPfPeiimBn5XQ
Bank RRN	--
Order ID	--
Invoice ID	--
Payment method	Wallet (Airtelmoney)
Customer details	+91 9137 953313 void@razorpay.com
Fee bearer	You pay the Razorpay platform fee
App Name	--
App ID	--
Description	Ad-Free and Fake News Detection Subscription Charges
Notes	--
Transfer	--

Create transfer

Refund Issue refund

No refund issued for this payment

Timeline

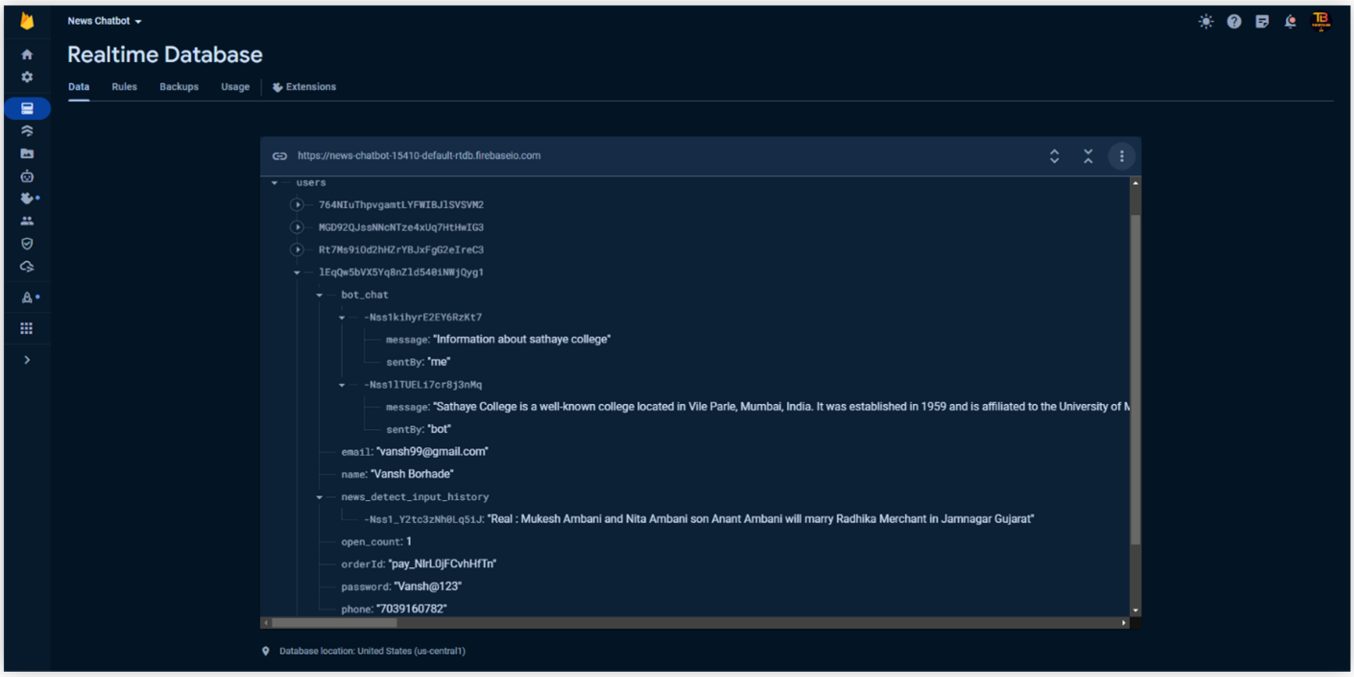
Payment created
Tue, Mar 12, 2024 4:06 PM

Payment authorized
Tue, Mar 12, 2024 4:06 PM

Payment captured
Wed, Mar 13, 2024 7:08 PM

Initiate full, partial, or instant refunds to your customers

Issue refund

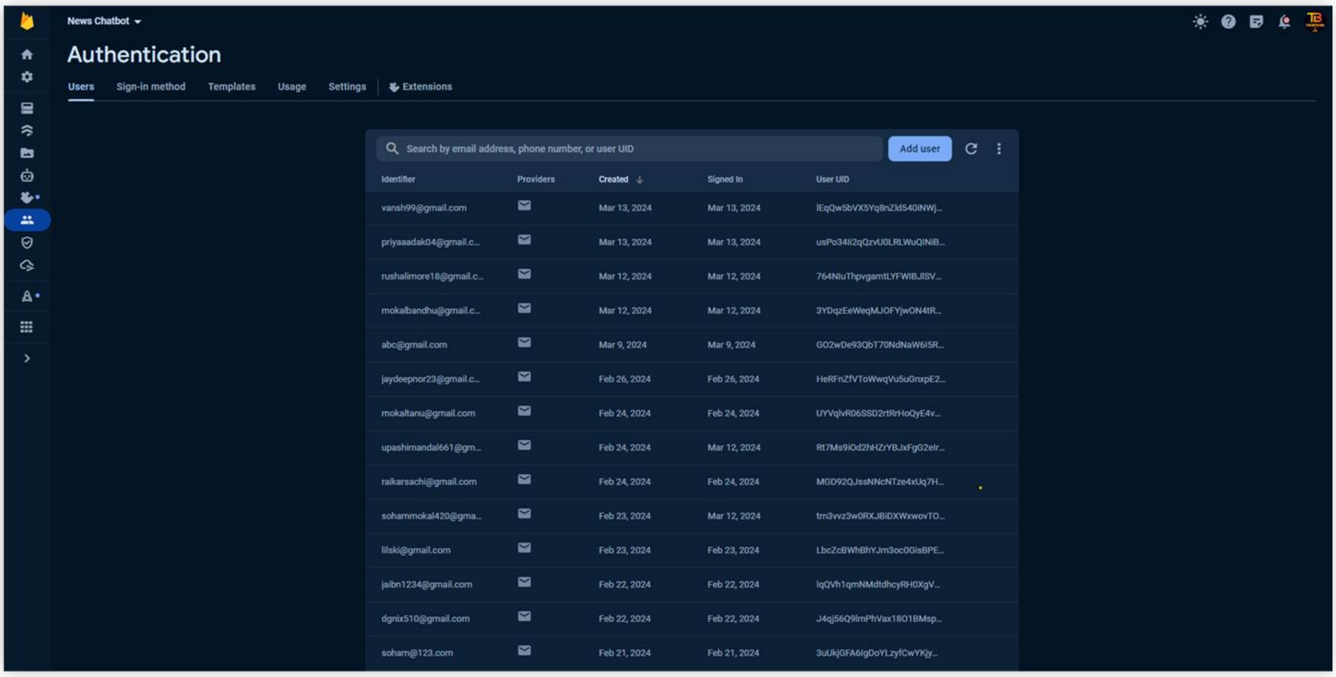


Realtime Database

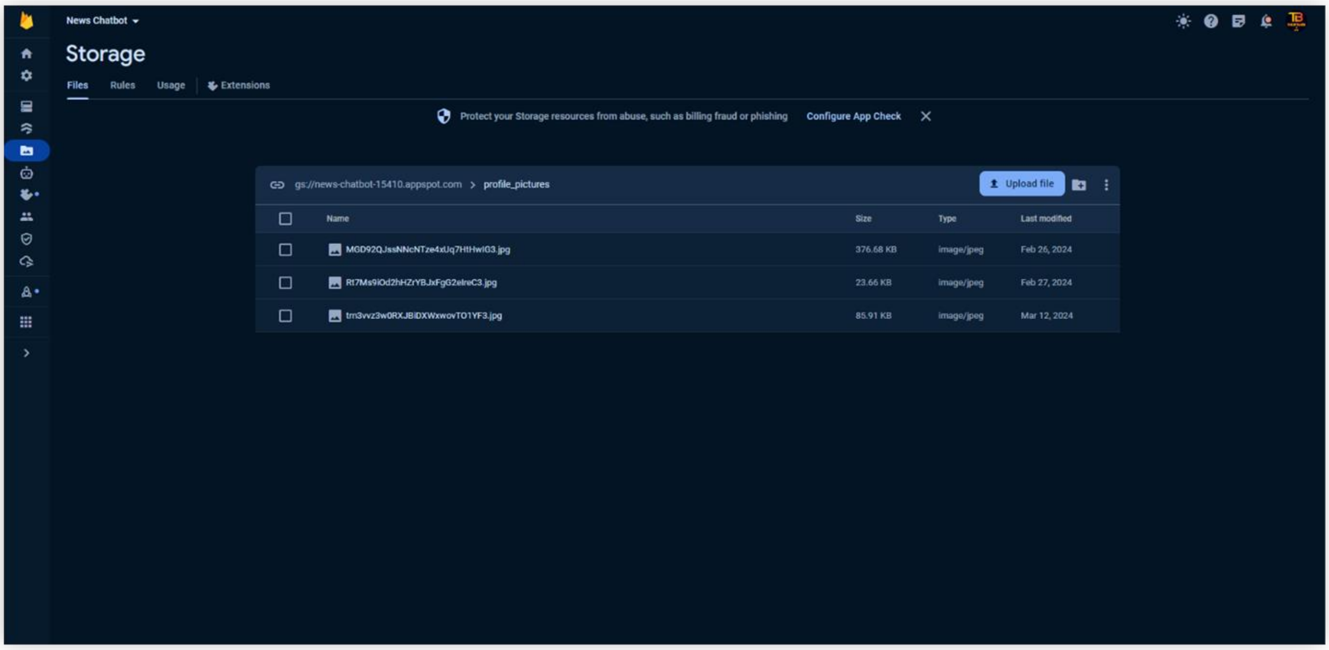
https://news-chatbot-15410-default-rtdb.firebaseio.com

```
users
    764NIuThpvgamtLYFWIBJlSVSVM2
    MGD92QJssNNcNTze4xUq7HtHwIG3
    Rt7Ms910d2hHZrYBJxFgG2eIreC3
    1EqQw5bVX5Yq8nZ1d540iNWjQyg1
        bot_chat
            -Nss1kihyrE2EY6RzKt7
                message: "Information about sathaye college"
                sentBy: "me"
            -Nss1lTUELi7cr8j3nMq
                message: "Sathaye College is a well-known college located in Vile Parle, Mumbai, India. It was established in 1959 and is affiliated to the University of M
                sentBy: "bot"
        email: "vansh99@gmail.com"
        name: "Vansh Borhade"
        news_detect_input_history
            -Nss1_Y2tc3zNh0Lq5iJ: "Real : Mukesh Ambani and Nita Ambani son Anant Ambani will marry Radhika Merchant in Jamnagar Gujarat"
        open_count: 1
        orderId: "pay_NlrL0jFCvhHfTn"
        password: "Vansh@123"
        phone: "7039160782"
```

Database location: United States (us-central1)



Identifier	Providers	Created	Signed In	User UID
vansh99@gmail.com	✉	Mar 13, 2024	Mar 13, 2024	lEqQw5bVX5Yq8nZld540iNWj...
priyaaadak04@gmail.c...	✉	Mar 13, 2024	Mar 13, 2024	usPo34li2qQzvU0LRLWuQINiB...
rushalimore18@gmail.c...	✉	Mar 12, 2024	Mar 12, 2024	764NluThpvgamtLYFWIBJlSV...
mokalbandhu@gmail.c...	✉	Mar 12, 2024	Mar 12, 2024	3YDqzEeWeqMJOFYjwON4tR...
abc@gmail.com	✉	Mar 9, 2024	Mar 9, 2024	GO2wDe93QbT70NdNaW6I5R...
jaydeepnor23@gmail.c...	✉	Feb 26, 2024	Feb 26, 2024	HeRFnZfVToWwqVu5uGnxpE2...
mokaltanu@gmail.com	✉	Feb 24, 2024	Feb 24, 2024	UYVqlvR06SSD2rtRrHoQyE4v...
upashimandal661@gm...	✉	Feb 24, 2024	Mar 12, 2024	Rt7Ms9iOd2hHZrYBJxFgG2elr...
raikarsachi@gmail.com	✉	Feb 24, 2024	Feb 24, 2024	MGD92QJssNNcNTze4xUq7H...
sohammokal420@gma...	✉	Feb 23, 2024	Mar 12, 2024	trn3vvz3w0RXJBiDXWxwovTO...
lilski@gmail.com	✉	Feb 23, 2024	Feb 23, 2024	LbcZcBWhBhYJm3oc0GisBPE...
jaibn1234@gmail.com	✉	Feb 22, 2024	Feb 22, 2024	lqQVh1qmNMdtdhcyRH0XgV...
dgnix510@gmail.com	✉	Feb 22, 2024	Feb 22, 2024	J4qj56Q9lmPhVax18O1BMsp...
soham@123.com	✉	Feb 21, 2024	Feb 21, 2024	3uUkjGFA6IgDoYLzyfCwYKjy...


News Chatbot
Storage
Files
Rules
Usage
Extensions
Protect your Storage resources from abuse, such as billing fraud or phishing
Configure App Check
gs://news-chatbot-15410.appspot.com > profile_pictures
Upload file

Name	Size	Type	Last modified
MGD92QJssNNcNTze4xUq7HtHwlG3.jpg	376.68 KB	image/jpeg	Feb 26, 2024
Rt7Ms9iOd2hHZrYBJxFgG2elreC3.jpg	23.66 KB	image/jpeg	Feb 27, 2024
trn3vvz3w0RXJBiDXWxwovTO1YF3.jpg	85.91 KB	image/jpeg	Mar 12, 2024

Firebase Authentication:

- Admins can access Firebase Authentication to manage user accounts, monitor sign-in activities, and enforce security measures.
- Generate reports on user authentication trends, including sign-up rates, login frequency, and account status changes.
- Monitor and moderate user accounts, including banning or suspending accounts if necessary.
-

Firebase Realtime Database:

- Utilize Firebase Realtime Database to monitor and manage data stored within the app, including user-generated content and app activity logs.
- Generate reports on data usage, storage trends, and performance metrics to optimize database performance.
- Implement security rules to restrict access to sensitive data and ensure compliance with privacy regulations.

Firebase Storage:

- Admins can utilize Firebase Storage to manage and monitor file storage within the app, including images, videos, and documents uploaded by users.
- Generate reports on storage usage, file types, and upload/download activities to track resource consumption and optimize storage allocation.
- Implement access controls and permissions to protect sensitive files and ensure data security.

Razorpay Admin Panel:

- Access the Razorpay Admin Panel to manage subscriptions, payments, and transactions processed through the app's subscription gateway.
- Generate reports on subscription revenue, payment success rates, and transaction history to track financial performance and revenue growth.
- Monitor subscription plans, pricing changes, and user subscriptions to optimize monetization strategies and maximize revenue.
- Overall, these admin-side tools provide comprehensive insights and management capabilities to effectively monitor, manage, and optimize various aspects of the app's authentication, database, storage, and payment functionalities.

Chapter 6
Testing

Test Cases

6.1 Registration:

Test Case No.	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status
1	Validate Name Input	User enters only letters	Name: "Vansh"	"Vansh" in the name field	"Vansh" in the name field	Pass
2	Validate Name Input	User enters numbers and letters	Name: "Vansh123"	"Vansh" in the name field	"Vansh" in the name field	Pass
3	Validate Name Input	User enters special characters	Name: "Vansh@12!"	"Vansh" in the name field	"Vansh" in the name field	Pass
4	Email Format Validation	User enters a valid email	Email: "vansh@gmail.com"	No error message; Email accepted	No error message; Email accepted	Pass
5	Email Format Validation	User enters an invalid email format	Email: "soham"	"Email Not Valid" error message	"Email Not Valid" error message	Pass
6	Email Format Validation	User enters an email without "@" symbol	Email: "vansh.com"	"Email Not Valid" error message	"Email Not Valid" error message	Pass
7	Number Format Validation	User enter a valid number	Number:7039160722	No error message;Number accepted	No error message;Number accepted	Pass
8	Number Length Validation	User enter a only 10 digit number	Number:7039160722	No error message;Number accepted	No error message;Number accepted	Pass
9	Password Length Validation	User should enter Symbol, uppercase, lowercase, number	Password:Vansh@0000	Password accepted	Password accepted	Pass

10	Password and Confirm Password Match	Password and Confirm Password are identical	Password:Vansh@0000 Confirm Password:Vansh@0000	Password accepted	Password accepted	Pass
11	Successful Resgisteration	User fill all fields correctly and submi	Name,email,number, Pass,confirm pass	Registeration successful, redirect to homepage	Registeration successful, redirect to homepage	Pass

6.2 Login:

Test Case No.	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status
1	Validate Email Input	User enters a valid email	Email: "vansh@gmail.com"	No error message; Email accepted	No error message; Email accepted	Pass
2	Validate Email Input	User enters an invalid email format	Email: "vansh"	"Email Not Valid" error message	"Email Not Valid" error message	Pass
3	Validate Email Input	User enters an email without "@" symbol	Email: "vansh.com"	"Email Not Valid" error message	"Email Not Valid" error message	Pass
4	Password Length Validation	User should enter Symbol, uppercase, lowercase, number	Password:Vansh@0000	Password accepted	Password accepted	Pass
5	Password and Confirm Password Match	Password and Confirm Password are identical	Password:Vansh@0000 Confirm Password:Vansh@0000	Password accepted	Password accepted	Pass
6	Login with Incorrect Credentials	User enters an incorrect email or password	Email: "invalid@hdhh.com", Password: "wrongpass"	Show error message; Stay on login page	Show error message; Stay on login page	Pass
7	Login with Empty Fields	User attempts to login without entering email or password	Email: "", Password: ""	Show error message; Prevent login	Show error message; Prevent login	Pass
8	Successful Login	User fills all fields correctly and submit	Email: "valid@gmail.com", Password: "Password@123"	Login successful, redirect to homepage	Login successful, redirect to homepage	Pass

6.3 News Display:

Test Case No.	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status
1	News Display	User clicks a news overview	Click on any news in swipe refresh layout	Open detailed news display with web view	Open detailed news display with web view	Pass
2	Search Bar	User enters a news to search	User enters “Ram Mandir”	Displays news related to Ram Mandir	Displays news related to Ram Mandir	Pass
3	Category	User clicks a category tab	Click on”Sports”	Displays news related to Sports	Displays news related to Sports	Pass
4	Navbar	User clicks any logo in navbar	Click on”Main Dashboard” logo	Opens main dashboard	Opens main dashboard	Pass
5	About us	User clicks on about us button	Click on circle button	Opens about us information	Opens about us information	Pass

6.4 Chatbot:

Test Case No.	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status
1	Chatbot	User Inputs Message	Best Anime in the history	This is subjective and varies from person to person. Some popular choices for the best anime in history include:1. Fullmetal Alchemist: Brotherhood 2. Attack on Titan 3. Death Note	This is subjective and varies from person to person. Some popular choices for the best anime in history include: 1. Fullmetal Alchemist: Brotherhoo d 2. Attack on Titan 3. Death Note	Pass
2	Chatbot	No internet	User opens the interface without internet	Display "Please Check Your Internet Connection"	Display "Please Check Your Internet Connection"	Pass

6.5 Fake News Detection:

Test Case No.	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status
1	News Detection	User enters a news headline to detect	Mukesh Ambani and Nita Ambani son Anant Ambani will marry Radhika Merchant in Jamnagar Gujarat	Probably Real News	Probably Real News	Pass
2	News Detection	User enters a news headline to detect	Salman Khan is Dead	Probably Fake News	Probably Fake News	Pass

6.6 Profile:

Test Case No.	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status
1	Profile Photo Upload	User selects a new profile photo	Choose a new photo file to upload	The selected file is successfully captured in the component's state	Selected file was successfully captured and prepared for upload.	Pass
2	Submit Profile Photo Change	User submits the profile photo change form	Click "Change Profile Photo" button after selecting a new photo	The photo is uploaded, and user data is refetched	Profile photo was successfully uploaded, user data was refreshed, displaying the new photo.	Pass
3	Error Handling on Photo Upload	Error occurs during the profile photo upload process	Simulate an error response from the server on profile photo upload	An error message is displayed, indicating the profile photo upload failed	An error occurred during the photo upload, "Error during the file upload" message was displayed to the user.	Pass

6.7 Stats:

Test Case No.	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status
1	Stats	User opens statistical history	User clicks stats	Usage History Is displayed in pie chart	Usage History Is displayed in pie chart	Pass
2	Stats	No internet	User opens the interface without internet	Display "Please Check Your Internet Connection"	Display "Please Check Your Internet Connection"	Pass

6.8 Subscription:

Test Case No.	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status
1	Razorpay Payment Display	User initiates payment for Subscription	Click "Purchase"	Razorpay payment modal pops up with correct amount and subscription details	Razorpay payment modal pops up with correct amount and subscription details	Pass
2	Credit Card Payment Process	Complete payment using a test credit card	Card Number: "4111 1111 1111 1111", Expiry: "12/30", CVV: "123"	Payment is successfully processed; Razorpay provides a payment ID; subscription is activated	Payment successful, received payment ID, subscription activated	Pass
3	Debit Card Payment Process	Complete payment using a test debit card	Card Number: "5555 5555 5555 4444", Expiry: "12/30", CVV: "123"	Payment is successfully processed; Razorpay provides a payment ID; subscription is activated	Payment successful, received payment ID, subscription activated	Pass
4	Net Banking Payment Process	Complete payment using net banking option	Select bank from net banking options	Payment is successfully processed; Razorpay provides a payment ID; subscription is activated	Payment is successfully processed; Razorpay provides a payment ID; subscription is activated	Pass
5	UPI Payment Process	Complete payment using UPI	UPI ID for test: "test@okicici"	Payment is successfully processed; Razorpay provides a payment ID; subscription is activated	Payment successful, received payment ID, subscription activated	Pass
6	Payment Failure Due to Incorrect CVV	Attempt to pay with incorrect CVV	Card Number: "4111 1111 1111 1111", Expiry: "12/30", CVV: "000"	Payment fails due to incorrect CVV; user is prompted to check their card details	Payment failed, user prompted to check CVV	Pass

7	Payment Failure Due to Expired Card	Attempt to pay with an expired card	Card Number: "4111 1111 1111 1111", Expiry: "01/20", CVV: "123"	Payment fails due to expired card; user is prompted to use a different card	Payment failed, user prompted to use a different card	Pass
8	Payment Failure Due to Network Issues	Attempt to pay during a network issue	Simulate network failure	Payment fails due to network issues; user receives an appropriate error message	Payment failed, network error message displayed	Pass
9	Receipt Upon Successful Payment	Complete any payment successfully	Use any successful payment method	User receives payment confirmation and a detailed receipt	Receipt received	Pass

Test Summary Report:

Number of Test Passed:	42
Number of Test Failed:	0

Chapter 7
Conclusion & Future Enhancement

Conclusion:

The "News Chatbot" project is a user-friendly Android application developed to combat the growing concern of fake news in today's digital age. The project utilizes machine learning algorithms and natural language processing techniques to analyze news articles and identify potential indicators of fake news, such as misleading information, biased language, and unreliable sources. By integrating with reputable fact-checking services and APIs, the application provides users with additional information about the credibility of news sources and articles, empowering them to make informed decisions about the information they consume. The project aims to promote media literacy, enhance information trustworthiness, and contribute to a safer online environment by reducing the spread of misinformation.

Motive for Building the App:

The primary motive for developing this app is to address the growing concern of misinformation and disinformation prevalent in today's digital age. By providing users with a platform that offers reliable news sources, implements fake news detection, and fosters critical thinking, the app aims to empower individuals to make informed decisions and combat the spread of misinformation.

Services Provided by the App:

1. News Aggregation: Aggregate news articles from various reliable sources to provide users with a diverse range of current and relevant information.
2. Fake News Detection: Utilize advanced algorithms to detect and flag potentially misleading or fabricated news articles, helping users identify trustworthy sources.
3. Chatbot Assistance: Offer an interactive chatbot feature to provide users with real-time assistance, answer queries, and offer personalized recommendations.
4. Subscription Management: Enable users to subscribe to premium features, such as ad-free usage and unlimited access to news and chatbot functionalities.
5. User Profile Management: Allow users to create and manage their profiles, customize settings, and update personal information, including profile pictures.
6. Usage Statistics: Provide users with detailed analytics and insights into their app usage, including reading habits, interaction frequency, and subscription activity.

7. Secure Payment Gateway: Integrate a secure payment gateway, such as Razorpay, for seamless subscription management and payment processing.
8. Content Storage: Enable users to upload and store multimedia content, such as images and videos, securely within the app using Firebase Storage.
9. Authentication and Security: Implement Firebase Authentication to ensure secure user authentication and enforce access controls to protect user data and privacy.
10. Real-time Database: Utilize Firebase Realtime Database to store and manage user-generated content, app activity logs, and other dynamic data in real-time.

Limitations:

The limitations of the data used in the "News Chatbot" project could have several potential implications for future research:

1. Data Quality Improvement: Addressing the limitations of the data used in the project can lead to future research focusing on enhancing data quality by incorporating more diverse and reliable sources.
2. Algorithm Enhancement: Future research may explore ways to improve machine learning algorithms to compensate for any shortcomings in the data used, ensuring better accuracy in detecting fake news.
3. User Engagement Strategies: Understanding the limitations of the data can prompt research into developing strategies to engage users effectively in providing feedback and insights to enhance the accuracy and relevance of news identification.
4. Privacy and Security Measures: Research may focus on implementing robust privacy and security measures to protect user data, especially when users contribute their knowledge and opinions to distinguish between reliable and unreliable sources or identify bias in news articles.
5. Collaborative Data Collection: Future studies could investigate collaborative approaches to data collection, where users actively participate in verifying news credibility, thus enriching the dataset with diverse perspectives and insights.

Future Enhancements:

To make the database of the news detection real-time, future enhancements of the "News Chatbot" app could include the following:

1. Real-time Data Integration: Incorporate real-time data integration from reputable news sources and fact-checking services to ensure the most up-to-date information is available to users. This can be achieved by establishing APIs and webhooks that allow for seamless data exchange.
2. User-Generated Content: Implement a system for users to report potential fake news articles directly within the app. This will allow for a crowdsourced approach to identifying fake news and can help improve the accuracy of the app's detection capabilities.
3. Artificial Intelligence and Machine Learning: Continuously update the machine learning algorithms used in the app to improve their accuracy in detecting fake news. This can be achieved by incorporating new data sources, refining the training process, and utilizing advanced AI techniques such as deep learning.
4. Push Notifications: Implement push notifications to alert users when a news article they are viewing has been flagged as potentially fake. This will help users make informed decisions in real-time and reduce the spread of misinformation.
5. Social Media Integration: Integrate the app with popular social media platforms to allow users to share and discuss news articles with their friends and followers. This will help promote media literacy and encourage users to critically evaluate the information they consume.

By implementing these enhancements, the "News Chatbot" app can provide users with a more comprehensive and real-time solution for detecting and combating fake news.

BIBLIOGRAPHY

Reference:

https://www.snopes.com/about/

https://arxiv.org/pdf/1707.03264v2.pdf

https://play.google.com/store/apps/details?id=io.scal.oigetit

https://www.kaggle.com/datasets/shivamtaneja2304/inshorts-dataset-english

https://newsapi.org/

https://www.ndtv.com/

https://timesofindia.indiatimes.com/

https://www.hindustantimes.com/

https://www.indiatoday.in/

https://indianexpress.com/

https://www.thehindu.com/

https://www.livemint.com/