REAL TIME AI POWERED SOCIAL MEDIA

A PROJECT REPORT

Submitted by

PRITHVIRAJ M

RANJITH S

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report "**REAL TIME AI POWERED SOCIAL MEDIA**" is the bonafide work of **PRITHVIRAJ M [211420104204], RANJITH S [211420104221]** who carried out the project work under my supervision.

Signature of the HOD with date
Dr L. JABASHEELA M.E., Ph.D.,
Professor and Head,

Department of Computer Science and Engineering,
Panimalar Engineering College
Chennai - 123

Signature of the Supervisor with date
Mr. D. ELANGOVAN, M.E.
Associate Professor

Department of Computer Science and Engineering,
Panimalar Engineering College
Chennai - 123

Submitted for the Project Viva – Voce examination held on ________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **RANJITH S [211420104221], PRITHVIRAJ M [211420104204]** hereby declare that thisproject report titled “**REAL TIME AI POWERED SOCIAL MEDIA”**, under the guidance of **Mr.D ELANGOVAN** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

Prithviraj M [211420104204]
Ranjith S [211420104221]

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his benevolent words and fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our directors **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express my heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Project Coordinator Dr. V. SUBEDHA, M.E., Ph.D** and **Project Guide Mr. D ELANGOVAN, M.E, Ph.D**and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

Prithviraj M [211420104204]
Ranjith S [211420104221]

ABSTRACT

An AI-powered social media with sentiment analysis is a platform that leverages advanced algorithms and techniques to help individuals and organizations to understand the sentiment and mood of social media data. The platform uses natural language processing, machine learning and mood detection to analyze the tone, sentiment and emotion expressed in social media content. This enables organizations to make informed decisions, respond effectively to customer needs and concerns, and identify emerging trends and opportunities .One of the key benefits of an AI-powered social media with sentiment analysis is the ability to improve the accuracy and speed of sentiment analysis. Traditional sentiment analysis methods often struggle with accuracy due to the complexity of language and the wide range of emotions and opinions expressed in social media content. However, by leveraging advanced algorithms such as deep learning and natural language processing, AI-powered sentiment analysis systems can achieve much higher accuracy rates .Another benefit of an AI-powered social media with sentiment analysis is the ability to handle large volumes of data. Social media generates massive amounts of data every day, making it difficult for organizations to manually analyze and extract insights from this data. However, an AI-powered platform can analyze social media data in real-time, providing organizations with valuable insights that can be used to make informed decisions and take action.An AI-powered social media with sentiment analysis is a powerful tool that can provide valuable insights into the sentiments and moods of social media data. The platform can help organizations make informed decisions, respond effectively to customer needs and concerns and identify emerging trends and opportunities

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The AI-powered social media app with sentiment analysis is a platform designed to help individuals and organizations analyze social media content in order to better understand the sentiments and moods of their audience. The app leverages advanced algorithms such as natural language processing, machine learning, and mood detection to analyze the tone, sentiment, and emotion expressed in social media content.The platform provides a user-friendly interface that allows users to easily navigate and analyze social media data. Users can specify the social media channels they want to monitor, and the app will collect and analyze data from those channels in real-time. The sentiment analysis features of the app allow users to quickly and accurately identify positive, negative, and neutral sentiment in social media content.The app is designed to be customizable to fit the specific needs of different organizations. Users can set up custom alerts and notifications to stay informed about changes in sentiment or emerging trends. The platform also provides robust reporting capabilities, allowing users to generate detailed reports on sentiment trends, audience engagement, and other key metrics.Overall, the AI-powered social media app with sentiment analysis is a powerful tool that can provide valuable insights into the sentiments and moods of social media data. It can help organizations make informed decisions, improve customer engagement, and identify new opportunities.

1.2 PROBLEM DEFINITION

- The problem that the AI-powered social media app with sentiment analysis addresses is the difficulty that organizations face in understanding the sentiment and mood of their audience on social media.
- Traditional sentiment analysis methods struggle with accuracy and scalability due to the complexity of language and the vast amount of data generated by social media.
- The app improves upon traditional methods by leveraging advanced algorithms such as deep learning and natural language processing, enabling it to achieve higher accuracy rates and handle large volumes of data.
- This allows organizations to gain valuable insights into audience sentiment and make informed decisions based on these insights.

1.3 LITERATURE SURVEY

INTRODUCTION

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources.

LITERATURE SURVEY

[1] Title: social media with machine

learning Introduction:

The study presented a comprehensive approach to analyze VKontakte data for the purpose of identifying effective teams and potential deviant behavior. By leveraging machine learning techniques, particularly neural networks and the k-means algorithm, the research demonstrated the feasibility of extracting valuable insights from social media data. The process involved data collection, preprocessing, training models, and clustering users based on their interests.Through this analysis, the study showcased the potential applications of such insights in real-world scenarios, including business companies and educational institutions. By understanding user behavior and interests, organizations can optimize team formation and mitigate potential risks associated with deviant behavior.However, while the system shows promise in providing valuable insights, it also raises important considerations regarding privacy, data ethics, and the potential for algorithmic biases. Future research should continue to address these concerns while exploring further applications and refinements of the proposed methodology..

Advantages:

The system offers several significant advantages. Firstly, it enables organizations to form highly effective teams by identifying users with similar interests and compatibility. This facilitates a positive working environment and enhances team productivity. Additionally, the system's capability to predict potential deviant behavior provides organizations with the opportunity to take proactive measures to mitigate risks and maintain a harmonious workplace. Furthermore, leveraging machine learning techniques ensures that decision-making processes are data-driven, leading to more informed and effective outcomes. The scalability of the approach allows it to be applied to large datasets, making it suitable for diverse applications across various contexts. Finally, the practical applications of the system in business companies, schools, and other organizations demonstrate its potential to improve team dynamics and organizational effectiveness.

Disadvantages:

the system also presents several challenges and disadvantages. Firstly, there are significant privacy concerns associated with analyzing personal data from social media platforms. Ensuring user consent and adhering to ethical principles regarding data usage are paramount but can be difficult to navigate effectively. Moreover, preprocessing data and ensuring its quality pose challenges, especially when dealing with large volumes of noisy and unstructured data. Additionally, machine learning models may exhibit biases, leading to inaccurate or unfair outcomes, particularly if the training data is not representative. Implementing and maintaining the system may also require substantial computational resources, expertise, and infrastructure, making it inaccessible for some organizations. Finally, the complexity of machine learning models can hinder interpretability, limiting transparency and accountability in decision-making processes. Addressing these challenges is crucial to ensure responsible and ethical use of the system while maximizing its potential benefits.

[2] Title: Detecting Psychological Stress using Machine Learning over Social Media Interaction

Introduction:

In the proposed system offers a promising approach to addressing the escalating concern of psychological stress through the utilization of social media data and Convolutional Neural Networks (CNN). By analyzing user behavior on platforms like Twitter, the system effectively identifies stress levels, enabling early detection and intervention to mitigate potential health risks.

Advantages:

One of the key advantages of this system is its ability to leverage readily available social media data for stress detection, making it accessible and cost-effective. By employing CNN, the system can efficiently process and classify large volumes of user-generated content, enabling timely intervention and support for individuals experiencing stress. Moreover, the automated nature of the system reduces the need for manual intervention, allowing for scalable implementation across diverse user populations. Additionally, the system offers the potential for personalized interventions based on individual stress profiles, enhancing the effectiveness of stress management strategies..

[3] Title: An approach to analyse and Forecast Social media data using Machine Learning

Introduction:

This research underscores the significance of employing advanced data analytics techniques, such as Machine Learning (ML) and Natural Language Processing (NLP), to gauge public sentiment on social media platforms like Twitter. By leveraging web scraping technology, we collected and analyzed 10,000 tweets related to the Goods and Service Tax (GST) in India, revealing valuable insights into user opinions. Through data visualization, sentiment analysis, and statistical approaches, we were able to discern that the majority of users expressed positive sentiments toward GST.

Advantages:

One of the primary advantages of this research is its ability to harness the vast amount of user-generated content on social media platforms, providing a real-time and comprehensive understanding of public opinion. By employing ML and NLP techniques, we were able to automate sentiment analysis, allowing for efficient processing of large datasets. Additionally, data visualization techniques facilitated the identification of patterns and trends within the Twitter dataset, enhancing the interpretability of results..

Disadvantages:

Despite its merits, this research also faces certain limitations. Web scraping, while effective in collecting data, may encounter challenges such as data incompleteness or inconsistencies, which can impact the reliability of findings. Moreover, sentiment analysis algorithms may not always accurately capture the nuances of human language, leading to potential misinterpretations of user sentiments. Additionally, the scope of this study was limited to Twitter, overlooking sentiments expressed on other social media platforms, thereby potentially limiting the comprehensiveness of the analysis.

[4] Title:A Survey on Machine Learning Methodologies in Social Network Analysis

Introduction:

this paper highlights the significant role of machine learning algorithms in advancing the field of social network analysis. Through the examination of various methodologies for detecting fake profiles, recognizing personality traits, and identifying signs of depression, it becomes evident that machine learning techniques offer valuable insights into online user behavior and interactions. By leveraging the vast amount of data available on social networking platforms, these algorithms enable researchers to extract meaningful patterns and trends, ultimately contributing to a deeper understanding of online social dynamics. One notable advantage of employing machine learning in social network analysis is its ability to handle large and complex datasets efficiently. These algorithms can process vast amounts of user-generated content, extracting relevant features and patterns that may not be readily apparent through traditional methods. Additionally, machine learning enables the automation of tasks such as profile verification, sentiment analysis, and anomaly detection, thereby streamlining the analysis process and increasing productivity.

this paper highlights the significant role of machine learning algorithms in advancing the field of social network analysis. Through the examination of various methodologies for detecting fake profiles, recognizing personality traits, and identifying signs of depression, it becomes evident that machine learning techniques offer valuable insights into online user behavior and interactions. By leveraging the vast amount of data available on social networking platforms, these algorithms enable researchers to extract meaningful patterns and trends, ultimately contributing to a deeper understanding of online social dynamics.

Advantages:

One notable advantage of employing machine learning in social network analysis is its ability to handle large and complex datasets efficiently. These algorithms can process vast amounts of user-generated content, extracting relevant features and patterns that may not be readily apparent through traditional methods. Additionally, machine learning enables the automation of tasks such as profile verification, sentiment analysis, and anomaly detection, thereby streamlining the analysis process and increasing productivity.

Disadvantages:

there are also challenges and limitations associated with the use of machine learning in social network analysis. One drawback is the potential for bias in algorithms, which may lead to inaccurate or unfair results, particularly in sensitive areas such as personality recognition and depression detection. Moreover, the reliance on labeled data for training machine learning models can pose challenges, as obtaining labeled datasets for tasks like depression detection may be difficult due to privacy concerns and ethical considerations.

[5] Title: A Survey of Sentiment Analysis from Social Media Data

Introduction:

For decades, UNESCO has led the international effort to ensure that science and technology develop within strong ethical boundaries, thanks to its unique mandate. Whether it's genetic research, climate change, or scientific research, UNESCO has established worldwide standards to maximize the advantages of scientific discoveries while reducing the hazards, ensuring they contribute to a more inclusive, sustainable, and peaceful world. It has also recognized frontier concerns in neuroscience ethics, climate engineering, and the internet of things.

Advantages:

The rapid emergence of artificial intelligence (AI) has provided numerous opportunities worldwide, ranging from aiding healthcare diagnostics to enabling human relationships via social media and increasing labour efficiencies through automated jobs. However, these rapid developments present significant ethical considerations.

Disadvantages:

In no other field is the ethical compass more important than in artificial intelligence. These general-purpose technologies are transforming the way we work, engage, and live. The world is about to change at a rate unprecedented since the introduction of the printing press six centuries ago. AI technology has significant benefits in many fields, but without ethical safeguards, it risks duplicatingreal-world prejudices causing divisions and faults in the system.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

- The existing system for social media sentiment analysis relies on traditional methods such as rule-based and lexicon-based approaches.
- These methods have limitations in terms of accuracy and scalability due to the complexity of language and the vast amount of data generated by social media.
- This makes it difficult for organizations to gain valuable insights into audience sentiment and make informed decisions based on these insight

2.2 PROPOSED SYSTEM

- The proposed system for social media sentiment analysis is an AI-powered app that utilizes advanced algorithms such as deep learning and natural language processing to achieve higher accuracy rates and handle large volumes of data.
- The app also includes features such as customized alerts and notifications, reporting, and visualization that provide organizations with valuable insights into audience sentiment and engagement.
- With the proposed system, organizations can make informed decisions based on data-driven insights and improve their social media strategy to better engage with their audience. and just for it.

2.3 DEVELOPMENT ENVIROMENT

SOFTWARE REQUIREMENT

- HTML
- CSS
- Java script
- Django
- Python
- Machine learning
- VS Code

HARDWARE REQUIREMENT

- Processor: Minimum 1 GHz
- Memory (RAM): 4 GB
- Hard Drive: 32 GB
- Internet Connection

CHAPTER 3

SYSTEM DESIGN

3.1 UML DIAGRAMS

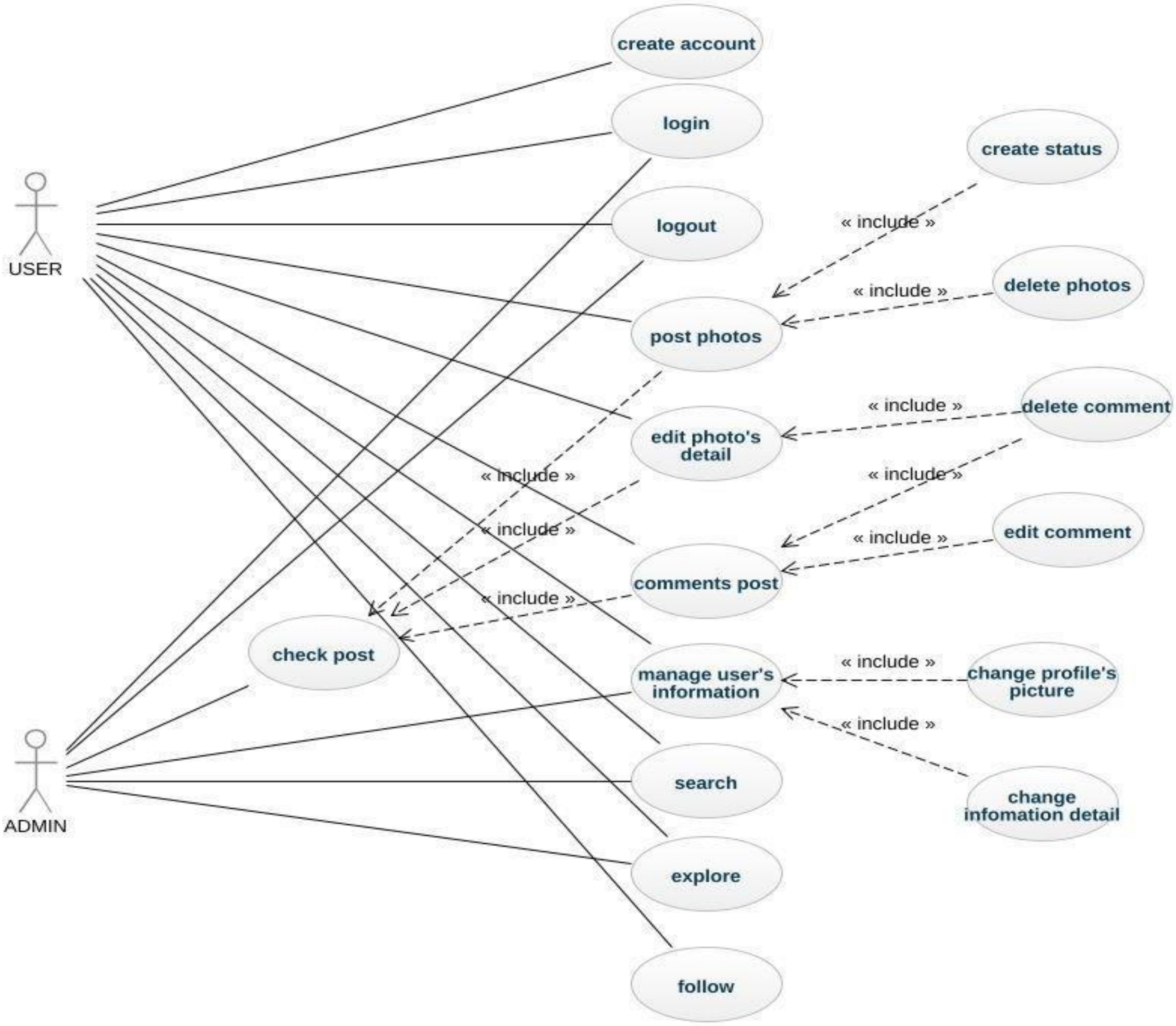


Fig 3.1.1 Use case diagram for Realtime AI powered social media app

This use case diagram refers to activities done within users and their corresponding use cases

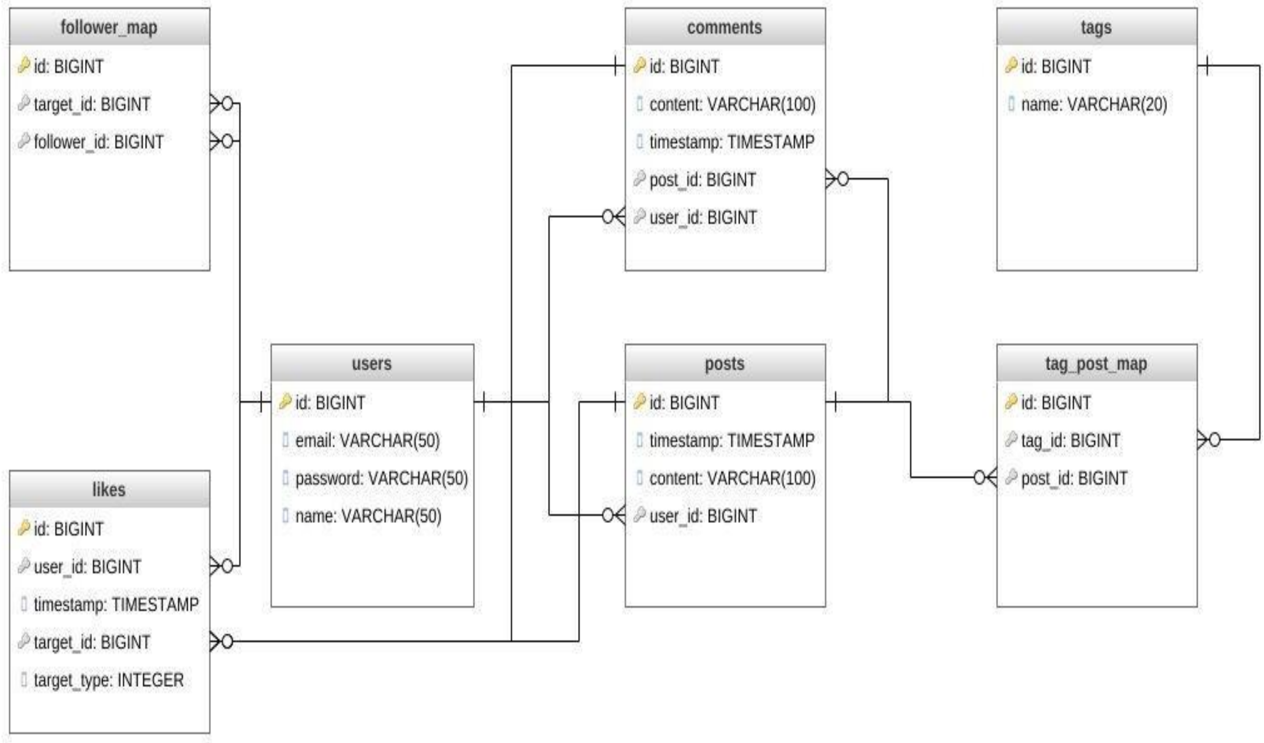


Fig 3.1.2 Class diagram for SOCIAL MEDIA

The class diagram refers to relationships between different classes that the user uses to communicate.

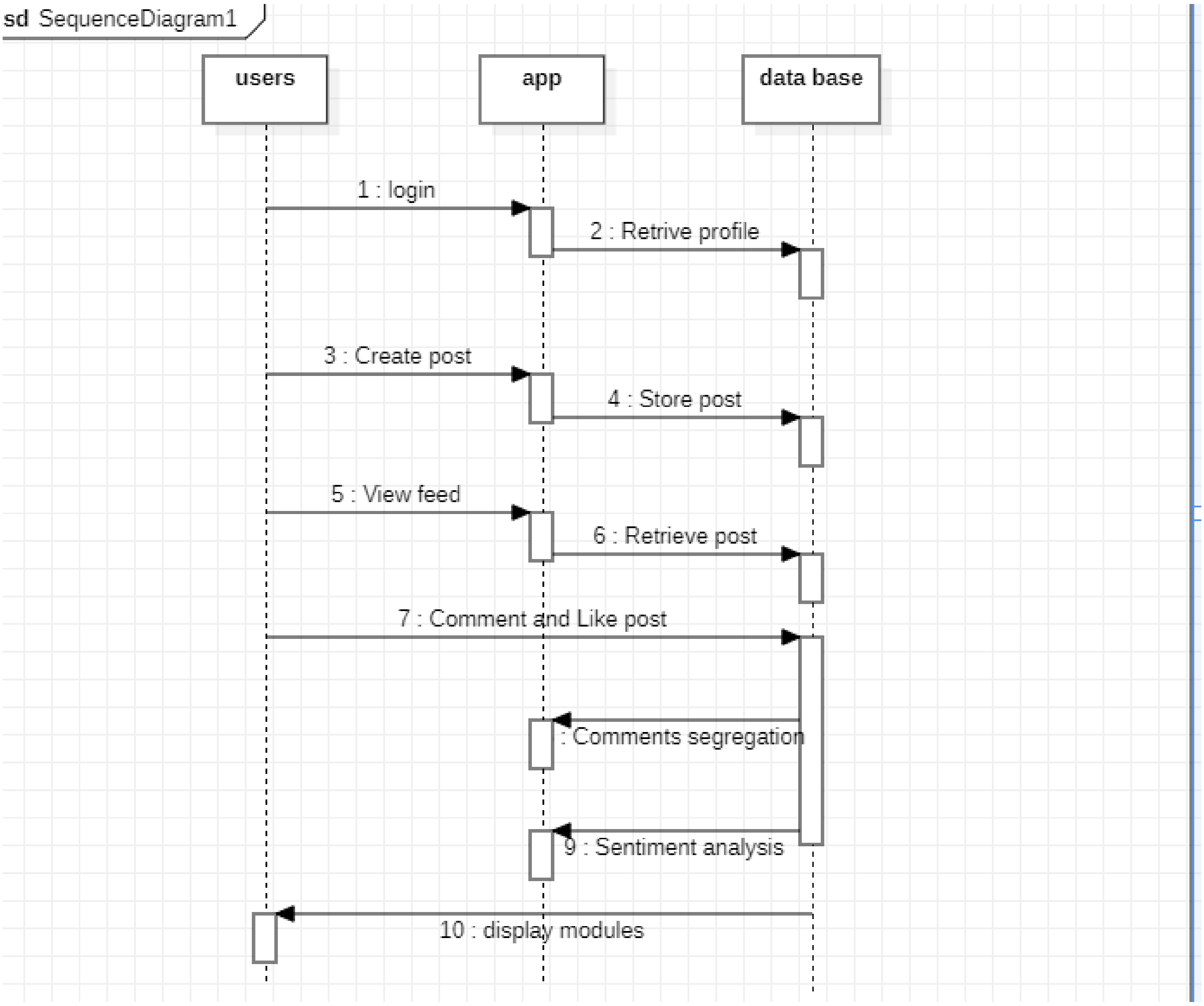


Fig 3.1.3 Sequence diagram for Realtime AI powered social media app

The sequence diagram of Realtime AI powered social media app which the users can communicate with.

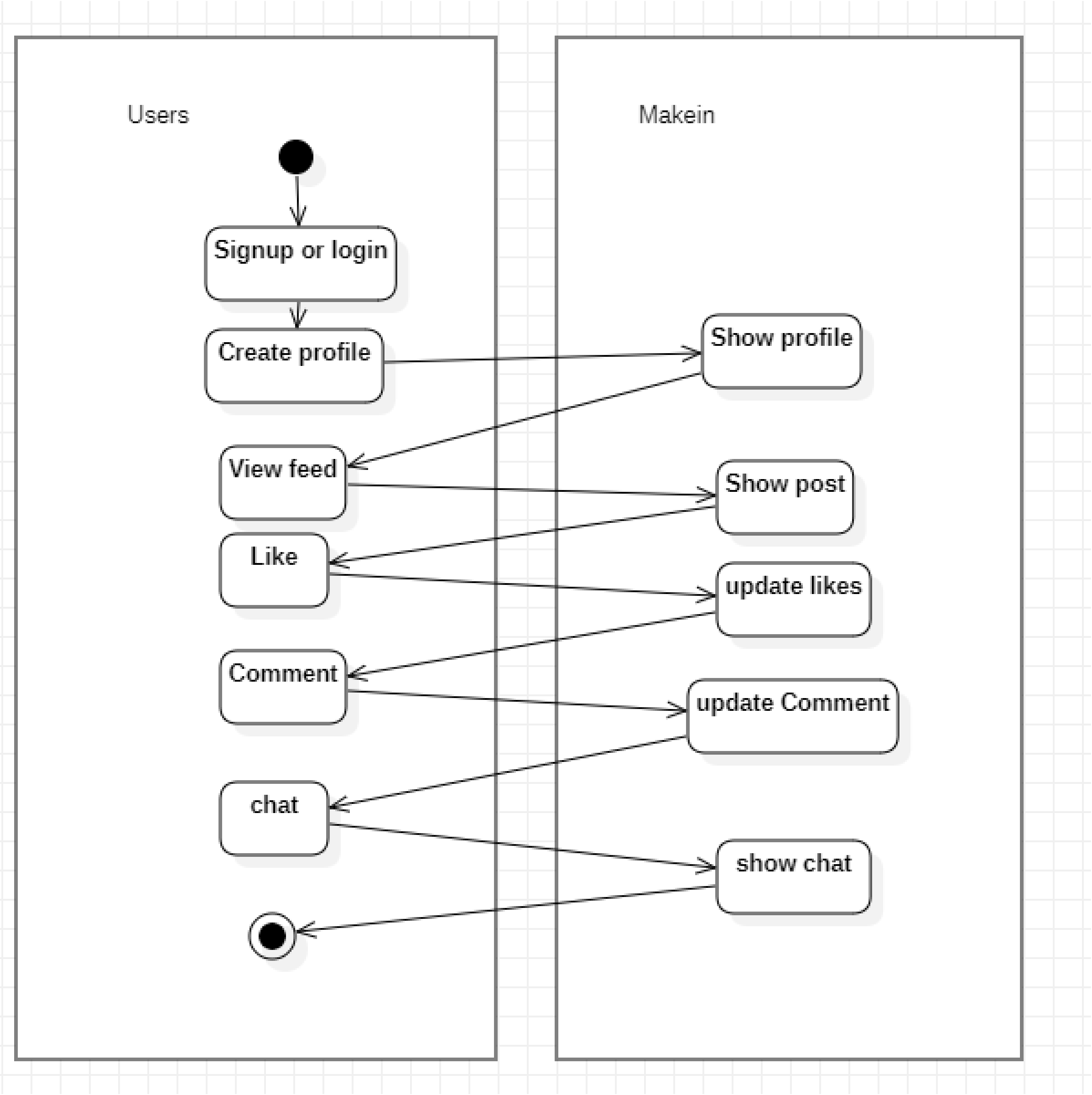


Fig 3.1.5 Activity diagram for Realtime AI powered social media app

The activity diagram of Realtime AI powered social media app shows the flow of activities of the user.

3.2 DATA DICTIONARY

This is normally represented as the data about data. It is also termed as metadata some times which gives the data about the data stored in the database. It defines each data term encountered during the analysis and design of a new system. Data elements can describe files or the processes. Following are some rules, which defines the construction of data dictionary entries:

User Data:

- UserID: Unique identifier for each user
- FirstName: User's first name
- LastName: User's last name
- Email: User's email address
- Password: User's password
- DateOfBirth: User's date of birth
- Gender: User's gender

Post Data:

- PostID: Unique identifier for each post
- UserID: Identifier for the user who created the post
- PostContent: Text content of the post
- Image: Image associated with the post
- Video: Video associated with the post
- DatePosted: Date and time when the post was created

Comment Data:

- CommentID: Unique identifier for each comment
- PostID: Identifier for the post the comment is associated with
- UserID: Identifier for the user who created the comment
- CommentContent: Text content of the comment
- DatePosted: Date and time when the comment was created

Like Data:

- LikeID: Unique identifier for each like

- PostID: Identifier for the post the like is associated with
- UserID: Identifier for the user who created the like
- LikeType: Type of like (e.g., thumbs up, heart)

Sentiment Analysis Data:

- PostID: Identifier for the post being analyzed
- SentimentScore: Numeric score representing the sentiment of the post (-1 to 1, with -1 being very negative and 1 being very positive)
- SentimentCategory: Categorical label representing the sentiment of the post (e.g., positive, negative, neutral)

Reporting and Visualization Data:

- ReportID: Unique identifier for each report
- UserID: Identifier for the user who generated the report
- ReportType: Type of report (e.g., sentiment trends, engagement metrics)
- ReportData: Data included in the report (e.g., graphs, tables).

3.2.1 SIGNUP TABLE

COLUMN NAME	DATA TYPE	DESCRIPTION	CONSTRAINT
NAME	VARCHAR	NAME OF THE USER	NOT NULL
USERNAME	VARCHAR	USERNAME OF THE USER	NOT NULL
PASSWORD	VARCHAR	PASSWORD OF THE USER	NOT NULL
REPEAT PASSWORD	VARCHAR	REPEAT THE PASSWORD OFTHE USER	NOT NULL
MAIL	VARCHAR	MAILID OF USER	NOT NULL

3.2.1 Signup table

3.2.2 SIGN IN TABLE

COLUMN NAME	DATA TYPE	DESCRIPTION	CONSTRAINT
USERNAME	VARCHAR	USERNAME OF THE USER	NOT NULL
PASSWORD	VARCHAR	PASSWORD OF THE USER	NOT NULL

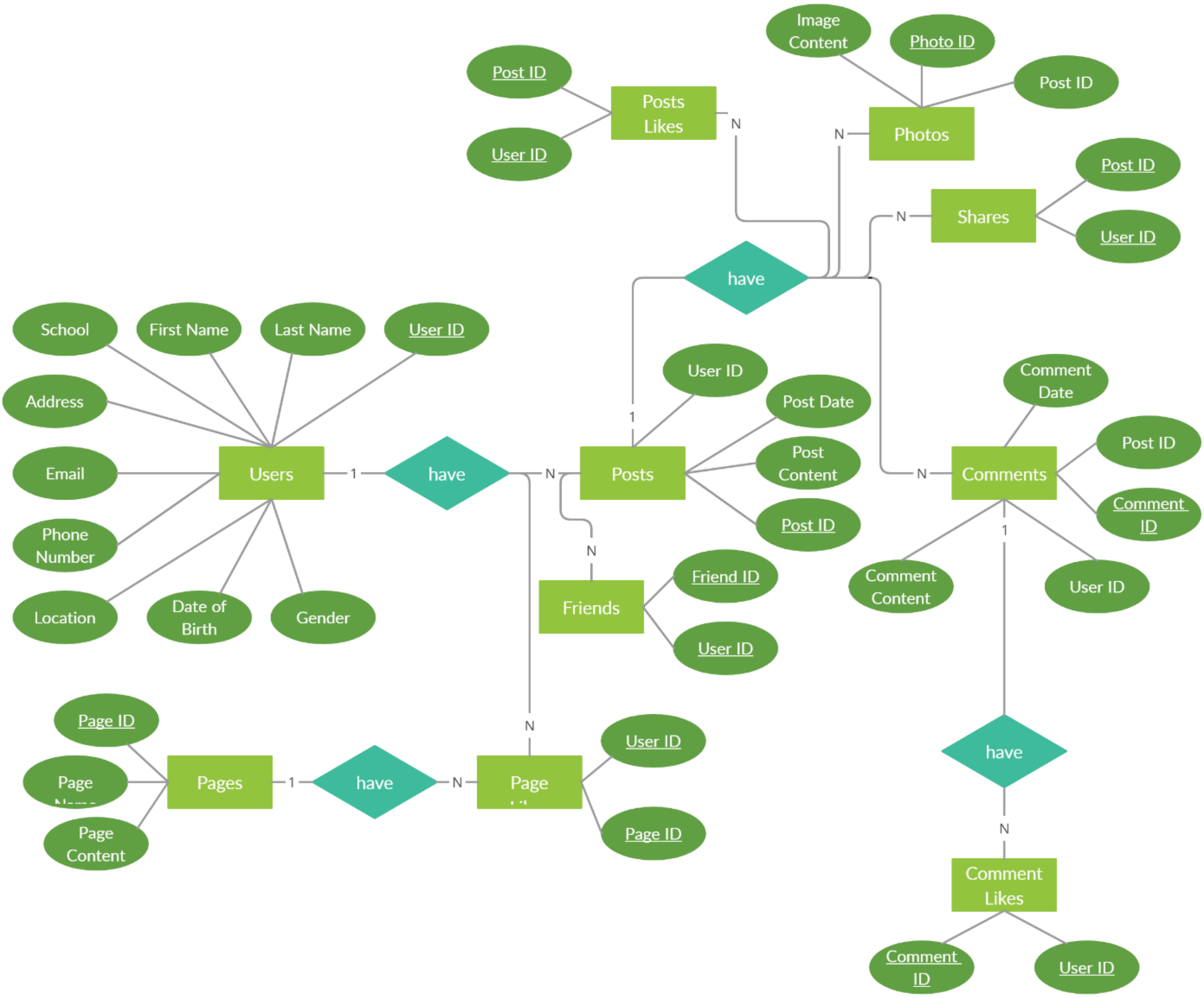
3.2.2 Sign in table

3.2.3 FUNCTION TABLE

COLUMN NAME	DATA TYPE	DESCRIPTION	CONSTRAINT
COMMENT	VARCHAR	COMMENT ON POST	NOT NULL
POST	VARCHAR	POSTING THE PICTURE	NOT NULL
EDIT	IMAGE	EDIT PROFILE	NOT NULL
PROFILE	VARCHAR	DESCRIPTION OF THE USER	NONE
LIKE	NUMBER	LIKE THE POST	NOT NULL

3.2.3 FUNCTION table

3.3 ER DIAGRAM



The E-R diagram gives the entities that is users, admin and their attributes.

3.4 DATAFLOW DIAGARM

0 LEVEL DFD

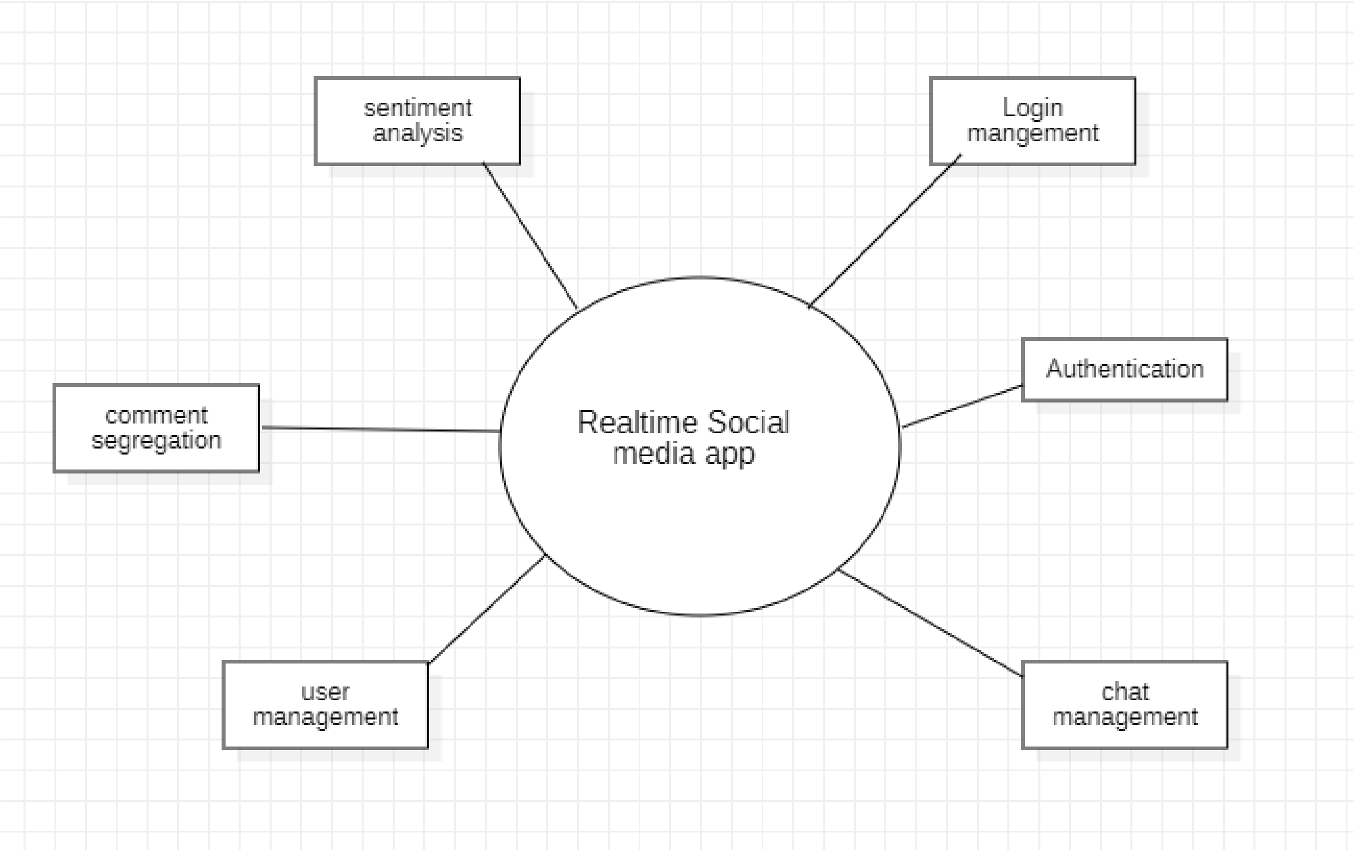


Fig 3.3.1 Dataflow diagram level 0

The zero level of data flow diagram of Realtime AI powered social media app which the users can communicate with.

FIRST LEVEL DFD

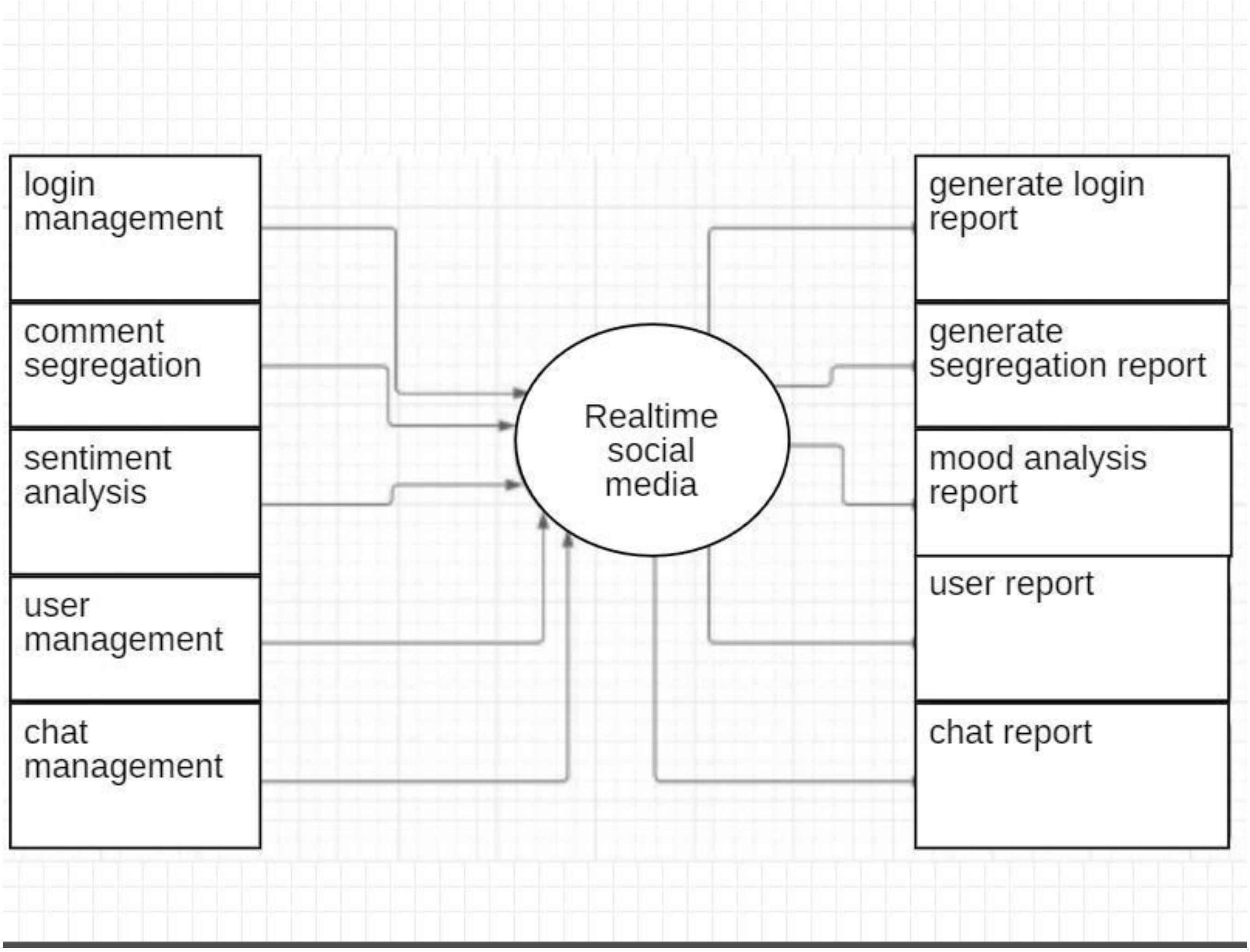


Fig 3.3.2 Dataflow diagram level 1

The first level of data flow diagram of real time AI powered social media shows the various management levels and their corresponding report

SECOND LEVEL DFD

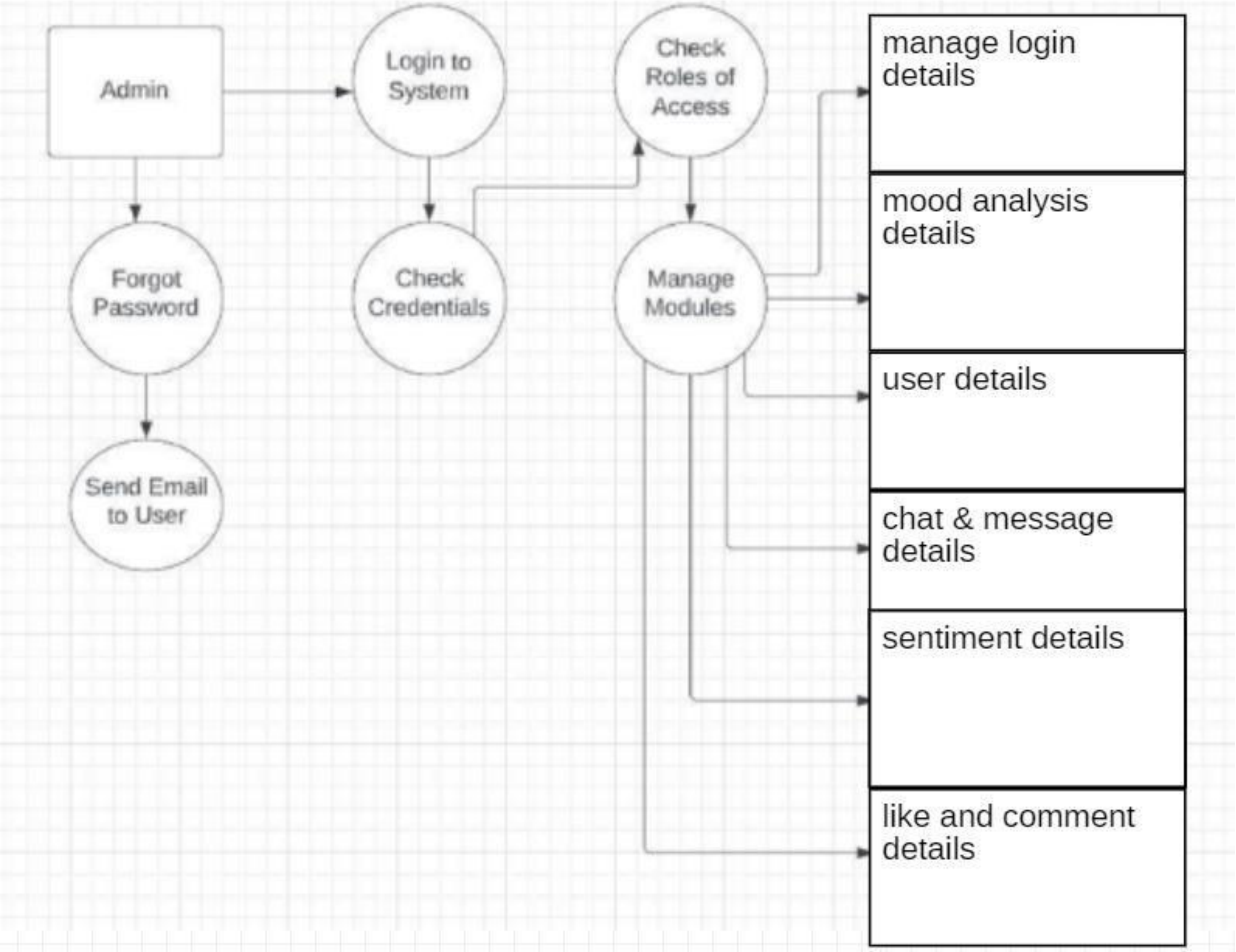


Fig 3.3.3 Dataflow diagram level 2

The second level of data flow diagram of Realtime AI powered social media app which the users can communicate with.

CHAPTER 4

SYSTEM ARCHITECTURE

4.1 ARCHITECTURE OVERVIEW

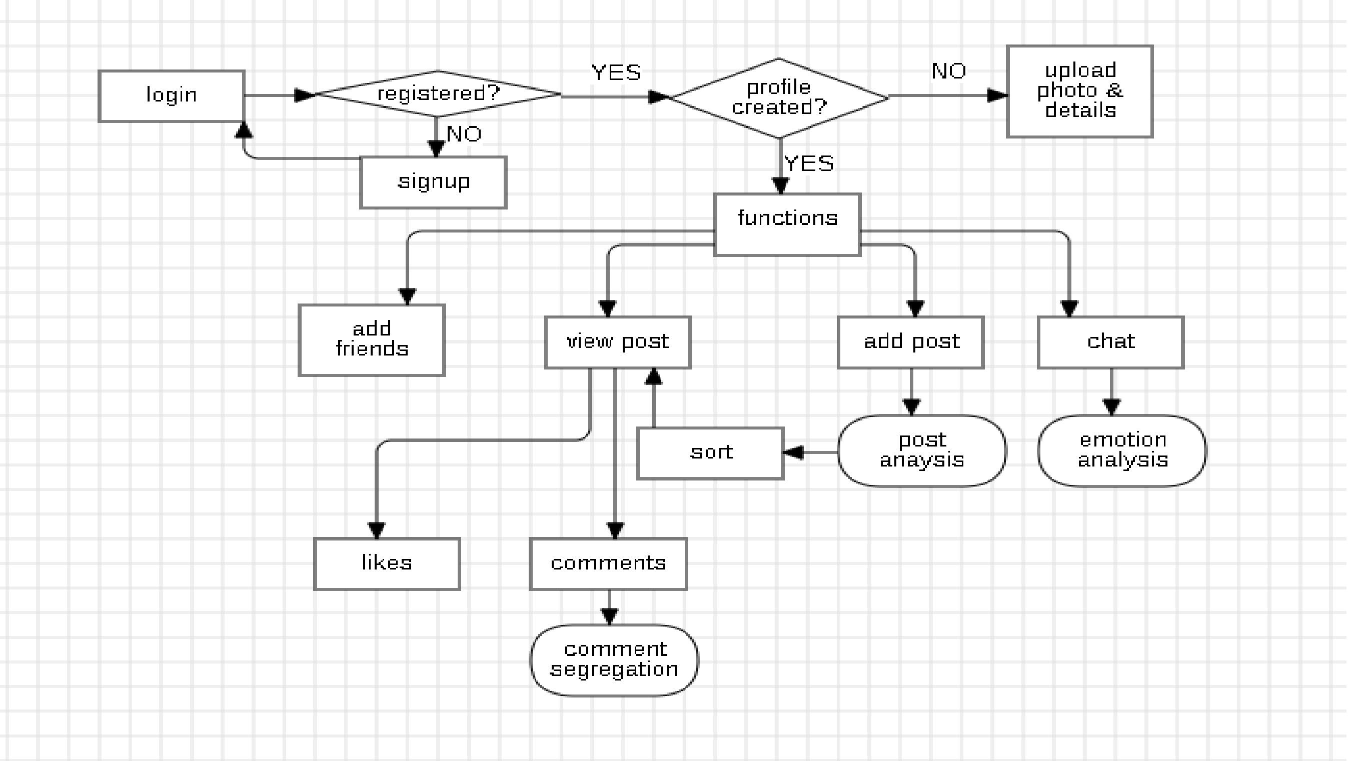


Fig 4.1 Architecture diagram for realtime social media app

The architecture of the AI-powered social media app with sentiment analysis consists of several key components, including data collection, data processing, sentiment analysis, and reporting. The data collection component collects social media data from various sources and stores it in a database. The data processing component uses natural language processing and machine learning algorithms to preprocess and transform the data into a format suitable for sentiment analysis. The sentiment analysis component applies advanced algorithms to classify the sentiment of social media content. Finally, the reporting component generates reports and visualizations that provide insights into sentiment trends, audience engagement, and other key metrics.

4.2 MODULE DESCRIPTION

Sign-up Module:

This module enables new users to create an account on the social media app. Users provide basic information such as their name, email address, and password.

Sign-in Module:

This module allows users to log in to their account and access the social media platform. Users enter their email address and password to authenticate their identity.

Post Module:

This module enables users to create new social media posts and share them with their audience. Users can write a text post, upload images or videos, and add hashtags and mentions.

Like and Comment Module:

This module enables users to interact with social media content by liking posts and leaving comments. Users can express their sentiment towards a post through likes and comments.

Sentiment Analysis Module:

This module analyzes the sentiment and mood of the social media content. It uses advanced algorithms to classify the sentiment of the content as positive, negative, or neutral.

Reporting and Visualization Module:

This module generates reports and visualizations that provide insights into sentiment trends, audience engagement, and other key metrics. The reports can be customized to fit the specific needs of different organizations and can be exported in various formats.

Overall, these modules work together to provide a user-friendly social media platform that allows users to share and interact with content while also providing valuable insights to organizations through sentiment analysis and reporting.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 FRONT-END CODING BASE.HTML

EDITPROFILE.HTML

```
{% extends 'base.html' %}
{% load static %}
{% load crispy_forms_tags %}
{% block content %}
<br><br><br><br>


  <div id="wrapper">
    <div class="container">
      <div class="phone-app-ddemo"></div>
      <div class="form-data">
        <form action="" method="POST" enctype="multipart/form-data">
                              {% csrf_token %}
          <div class="logo">
            <h1 >Edit Profile</h1>
          </div><br><br>
          {{form.image}} <br><br>
                      {{form.first_name}} <br><br>
                      {{form.last_name}} <br><br>
                      {{form.bio}} <br><br>
                      {{form.location}} <br><br>
                      {{form.url}} <br><br><br><br>
          <button class="form-btn btn-lg btn-danger" type="submit">Update
Profile</button>
          <!-- <a href="index.html" class="form-btn" type="submit">Log in</a> -->

        </form>

      </div>
    </div>
```

```
 </div>

{% endblock content %}
```

FRONTPAGE.HTML

```
{% extends 'core/base.html' %}

{% block title %}Home Page{% endblock title %}

{% block content %}
    <div class="columns is-multiline mb-6">
        <div class="column is-12 has-test-centered mt-6 mb-6">
            <h2 class="is-size-3">Latest Products</h2>
        </div>
        {% for product in newest_products %}
            {% include 'product/parts/list_item.html' %}
            {% endfor %}

          </div>

     {% endblock content %}
```

PROFILE.HTML

```
{% extends 'base.html' %}
{% load static %}
{% block content %}
<!DOCTYPE html>
<html lang="en">
<head>

    <!-- Metadata -->
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <meta name="author" content="George W. Park">
    <meta name="description" content="This project demonstrates how CSS grid (with
flexbox and float fallback) can be used to re-create the layout of an Instagram profile
page.">

    <!-- Title -->
    <title>Instagram Profile Layout</title>

    <!-- External CSS Files -->
    <link rel="stylesheet" href="{% static 'assets3/css/reset.css' %}">
    <link rel="stylesheet" href="{% static 'assets3/css/styles.css' %}">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600">
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.2.0/css/all.css">

</head>
<body>

    <header>
        <br><br><br>
        <div class="container">

            <div class="profile">
        <div class="profile-image">

          <img src="{{profile.image.url}}" alt="" style="width: 200px; height: 200px;">

        </div>

        <div class="profile-user-settings">
            {% if profile.first_name %}
            <h6 class="profile-user-name">{{profile.first_name}} {{profile.last_name}}</h6> <br>
            {% endif %}
            <h5 style="font-size: 15px;" class="profile-user-name">@{{profile.user.username}}</h5>

            {% if request.user.is_authenticated and request.user == profile.user %}
            <a style="text-decoration: none;" href="{% url 'editprofile' %}" class="btn profile-edit-btn btn-primary">Edit Profile</a>
            <a style="text-decoration: none;" href="{% url 'sign-out' %}" class="btn profile-edit-btn btn-danger ">Logout</a>



            {% endif %}
            <br><br>
            {% if request.user.is_authenticated and request.user != profile.user %}
              {% if follow_status == True %}
                <a href="{% url 'follow' profile.user 0 %}" class="btn profile-edit-btn btn-secondary" >Unfollow</a> <br><br>
              {% else %}
                <a href="{% url 'follow' profile.user 1 %}" class="btn profile-edit-btn btn-primary" >Follow</a> <br><br>
              {% endif %}
            {% endif %}

            {% if request.user.is_authenticated and request.user != profile.user %}
                <a href="{% url 'conversation' profile.user %}" class="btn profile-edit-btn btn-danger">Message</a> <br><br>
            {% endif %}
            <br>
        </div>



        <div class="profile-bio">
          <br>
          {% if profile.bio %}
          <p><span class="profile-real-name"></span> {{profile.bio}}</p>
          {% endif %}
          {% if profile.location %}
          <p><span class="profile-real-name"></span> {{profile.location}}</p>
          {% endif %}
          {% if profile.url %}
          <p><span class="profile-real-name"></span><a
href="{{profile.url}}">{{profile.url}}</a></p>
          {% endif %}
          <br><br>
          <div class="profile-user-settings">
            {% if request.user == profile.user %}
            <a href="{% url 'profilefavourite' profile.user %}" class="btns profeile-
edit-btn"> <svg xmlns="http://www.w3.org/2000/svg" width="40" height="40"
style="color:blue;" fill="currentColor" class="bi bi-bookmark-star-fill" viewBox="0 0 16
16">
                <path fill-rule="evenodd" d="M2 15.5V2a2 2 0 0 1 2-2h8a2 2 0 0 1 2
2v13.5a.5.5 0 0 1-.74.439L8 13.069l-5.26 2.87A.5.5 0 0 1 2 15.5zM8.16 4.1a.178.178 0 0
0-.32 0l-.634 1.285a.178.178 0 0 1-.134.098l-1.42.206a.178.178 0 0 0-.098.303L6.58
6.993c.042.041.061.1.051.158L6.39 8.565a.178.178 0 0 0 .258.187l1.27-.668a.178.178 0
0 1 .165 0l1.27.668a.178.178 0 0 0 .257-.187L9.368 7.15a.178.178 0 0 1 .05-.158l1.028-
1.001a.178.178 0 0 0-.098-.303l-1.42-.206a.178.178 0 0 1-.134-.098L8.16 4.1z"/>
              </svg></a><span class="profile-stat-count" style="font-style: bold;font-
size: 20px;">Tagged</span>
            {% endif %}
            <br>
          </div>
        </div>



        <div class="profile-stats">
          <br><br>

          <ul>
```

```
                  <li><span class="profile-stat-count" style="font-style: bold;font-size: 35px;">{{posts_count}}</span><a href="{% url 'profile' profile.user %}" ><svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" style="color:blue;" fill="currentColor" class="bi bi-images" viewBox="0 0 16 16">
                    <path d="M4.502 9a1.5 1.5 0 1 0 0-3 1.5 1.5 0 0 0 0 3z"/>
                    <path d="M14.002 13a2 2 0 0 1-2 2h-10a2 2 0 0 1-2-2V5A2 2 0 0 1 2 3a2 2 0 0 1 2-2h10a2 2 0 0 1 2 2v8a2 2 0 0 1-1.998 2zM14 2H4a1 1 0 0 0-1 1h9.002a2 2 0 0 1 2 2v7A1 1 0 0 0 15 11V3a1 1 0 0 0-1-1zM2.002 4a1 1 0 0 0-1 1v8l2.646-2.354a.5.5 0 0 1 .63-.062l2.66 1.773 3.71-3.71a.5.5 0 0 1 .577-.094l1.777 1.947V5a1 1 0 0 0-1-1h-10z"/>
                   </svg> </a></li>
                  <li><span class="profile-stat-count" style="font-style: bold;font-size: 35px;">{{followers_count}}</span><svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" style="color:blue;" fill="currentColor" class="bi bi-people-fill" viewBox="0 0 16 16">
                    <path d="M7 14s-1 0-1-1 1-4 5-4 5 3 5 4-1 1-1 1H7Zm4-6a3 3 0 1 0 0-6 3 3 0 0 0 0 6Zm-5.784 6A2.238 2.238 0 0 1 5 13c0-1.355.68-2.75 1.936-3.72A6.325 6.325 0 0 0 5 9c-4 0-5 3-5 4s1 1 1 1h4.216ZM4.5 8a2.5 2.5 0 1 0 0-5 2.5 2.5 0 0 0 0 5Z"/>
                   </svg></li>
                  <li><span class="profile-stat-count" style="font-style: bold;font-size: 35px;">{{following_count}}</span><svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" style="color:blue;" fill="currentColor" class="bi bi-people" viewBox="0 0 16 16">
                    <path d="M15 14s1 0 1-1-1-4-5-4-5 3-5 4 1 1 1 1h8Zm-7.978-1A.261.261 0 0 1 7 12.996c.001-.264.167-1.03.76-1.72C8.312 10.629 9.282 10 11 10c1.717 0 2.687.63 3.24 1.276.593.69.758 1.457.76 1.72l-.008.002a.274.274 0 0 1-.014.002H7.022ZM11 7a2 2 0 1 0 0-4 2 2 0 0 0 0 4Zm3-2a3 3 0 1 1-6 0 3 3 0 0 1 6 0ZM6.936 9.28a5.88 5.88 0 0 0-1.23-.247A7.35 7.35 0 0 0 5 9c-4 0-5 3-5 4 0 .667.333 1 1 1h4.216A2.238 2.238 0 0 1 5 13c0-1.01.377-2.042 1.09-2.904.243-.294.526-.569.846-.816ZM4.92 10A5.493 5.493 0 0 0 4 13H1c0-.26.164-1.03.76-1.724.545-.636 1.492-1.256 3.16-1.275ZM1.5 5.5a3 3 0 1 1 6 0 3 3 0 0 1-6 0Zm3-2a2 2 0 1 0 0 4 2 2 0 0 0 0-4Z"/>
                   </svg></li>

                </ul>
                <br>

              </div>

            </div>
            <!-- End of profile section -->

          </div>
          <!-- End of container -->
```

```
    </header>

    <main>

        <div class="container">

            <div class="gallery">

                {% for post in posts %}
                <a href="{{post.get_absolute_url}}"><div class="gallery-item" tabindex="0">

                    <img src="{{ post.picture.url }}" class="gallery-image" alt="" style="width: 500px; height: 300px; object-fit: cover;">

                    <div class="gallery-item-info">

                        <ul>
                            <li class="gallery-item-likes"><span class="visually-hidden">Likes:</span><i class="fas fa-heart" aria-hidden="true"></i> {{post.likes}}</li>
                            <li class="gallery-item-comments"><span class="visually-hidden">Comments:</span><i class="fas fa-comment" aria-hidden="true"></i> {{post.comment.count}}</li>
                        </ul>

                    </div>

                </div></a>
                {% endfor %}





            </div>
            <!-- End of gallery -->


        </div>
        <!-- End of container -->

    </main>
    {% endblock content %}
```

5.2 BACK-END CODING

MODEL.PY

```
from django.db import models
from django.contrib.auth.models import User
import PIL
from PIL import Image
from django.db.models.base import Model
from django.db.models.fields import DateField
from django.urls import reverse
from django.db.models.signals import post_save
import uuid
from django.utils import timezone
from post.models import Post


class Profile(models.Model):
    user = models.OneToOneField(User, related_name='profile',
on_delete=models.CASCADE)
    image = models.ImageField(upload_to="profile_pciture", null=True,
default="default.jpg")
    first_name = models.CharField(max_length=200, null=True, blank=True)
    last_name = models.CharField(max_length=200, null=True, blank=True)
    bio = models.CharField(max_length=200, null=True, blank=True)
    location = models.CharField(max_length=200, null=True, blank=True)
    url = models.URLField(max_length=200, null=True, blank=True)
    favourite = models.ManyToManyField(Post, blank=True)



    def save(self, *args, **kwargs):
        super().save(*args, **kwargs)

    def __str__(self):
        return f'{self.user.username} - Profile'

    def save(self, *args, **kwargs):
        super().save(*args, **kwargs)

        img = Image.open(self.image.path)
        if img.height > 300 or img.width > 300:
            output_size = (300, 300)
            img.thumbnail(output_size)
            img.save(self.image.path)


def  create_user_profile(sender, instance,  created,  **kwargs):
    if created:
            Profile.objects.create(user=instance)

def save_user_profile(sender, instance, **kwargs):
    instance.profile.save()

post_save.connect(create_user_profile, sender=User)
post_save.connect(save_user_profile, sender=User)
```

URLS.PY

```
from django.urls import path
from . import views
from django.contrib.auth import views as auth_views
from django.urls import reverse_lazy
from authy.views import UserProfile, EditProfile

urlpatterns = [
   # Profile Section
   path('profile/edit', EditProfile, name="editprofile"),

   # User Authentication
   path('sign-up/', views.register, name="sign-up"),
   path('sign-in/', auth_views.LoginView.as_view(template_name="sign-in.html",
redirect_authenticated_user=True), name='sign-in'),
   path('sign-out/', auth_views.LogoutView.as_view(template_name="sign-out.html"),
name='sign-out'),
]
```

VIEWS.PY

```
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth.decorators import login_required
from django.urls import reverse
from django.http import HttpResponseRedirect
from django.core.paginator import Paginator
from django.db import transaction
from django.contrib.auth.models import User
from django.core.exceptions import ObjectDoesNotExist
from django.contrib import messages
from django.contrib.auth import authenticate, login


from post.models import Post, Follow, Stream
from django.contrib.auth.models import User
from authy.models import Profile
from .forms import EditProfileForm, UserRegisterForm
from django.urls import resolve
from comment.models import Comment

def UserProfile(request, username):
    Profile.objects.get_or_create(user=request.user)
    user = get_object_or_404(User, username=username)
    profile = Profile.objects.get(user=user)
    url_name = resolve(request.path).url_name
    posts = Post.objects.filter(user=user).order_by('-posted')

    if url_name == 'profile':
        posts = Post.objects.filter(user=user).order_by('-posted')
    else:
        posts = profile.favourite.all()

    # Profile Stats
    posts_count = Post.objects.filter(user=user).count()
    following_count = Follow.objects.filter(follower=user).count()
    followers_count = Follow.objects.filter(following=user).count()
    # count_comment = Comment.objects.filter(post=posts).count()
    follow_status = Follow.objects.filter(following=user, follower=request.user).exists()

    # pagination
    paginator = Paginator(posts, 8)
    page_number = request.GET.get('page')
    posts_paginator = paginator.get_page(page_number)

    context = {
        'posts': posts,
        'profile':profile,
        'posts_count':posts_count,
        'following_count':following_count,
        'followers_count':followers_count,
        'posts_paginator':posts_paginator,
        'follow_status':follow_status,
        # 'count_comment':count_comment,
    }
    return render(request, 'profile.html', context)

def EditProfile(request):
    user = request.user.id
    profile = Profile.objects.get(user__id=user)

    if request.method == "POST":
        form = EditProfileForm(request.POST, request.FILES, instance=request.user.profile)
        if form.is_valid():
            profile.image = form.cleaned_data.get('image')
            profile.first_name = form.cleaned_data.get('first_name')
            profile.last_name = form.cleaned_data.get('last_name')
            profile.location = form.cleaned_data.get('location')
            profile.url = form.cleaned_data.get('url')
            profile.bio = form.cleaned_data.get('bio')
            profile.save()
            return redirect('profile', profile.user.username)
    else:
        form = EditProfileForm(instance=request.user.profile)

    context = {
        'form':form,
    }
    return render(request, 'editprofile.html', context)

def follow(request, username, option):
    user = request.user
    following = get_object_or_404(User, username=username)

    try:
        f, created = Follow.objects.get_or_create(follower=request.user, 
following=following)

        if int(option) == 0:
            f.delete()
            Stream.objects.filter(following=following, user=request.user).all().delete()
        else:
            posts = Post.objects.all().filter(user=following)[:25]
            with transaction.atomic():
                for post in posts:
                    stream = Stream(post=post, user=request.user, date=post.posted, 
following=following)
```

```
                    stream.save()
        return HttpResponseRedirect(reverse('profile', args=[username]))

    except User.DoesNotExist:
        return HttpResponseRedirect(reverse('profile', args=[username]))


def register(request):
    if request.method == "POST":
        form = UserRegisterForm(request.POST)
        if form.is_valid():
            new_user = form.save()
            # Profile.get_or_create(user=request.user)
            username = form.cleaned_data.get('username')
            messages.success(request, f'Hurray your account was created!!')

            # Automatically Log In The User
            new_user = authenticate(username=form.cleaned_data['username'],
                                    password=form.cleaned_data['password1'],)
            login(request, new_user)
            # return redirect('editprofile')
            return redirect('index')



    elif request.user.is_authenticated:
        return redirect('index')
    else:
        form = UserRegisterForm()
    context = {
        'form': form,
    }
    return render(request, 'sign-up.html', context)
```

FORMS.PY

```
from django import forms

from authy.models import Profile

from django.contrib.auth.models import User

from django.contrib.auth.forms import UserCreationForm
```

```
class EditProfileForm(forms.ModelForm):
    image = forms.ImageField(required=True)
    first_name = forms.CharField(widget=forms.TextInput(attrs={'class': 'input',
'placeholder': 'First Name'}), required=True)
    last_name = forms.CharField(widget=forms.TextInput(attrs={'class': 'input',
'placeholder': 'Last Name'}), required=True)
    bio = forms.CharField(widget=forms.TextInput(attrs={'class': 'input', 'placeholder':
'Bio'}), required=True)
    url = forms.CharField(widget=forms.TextInput(attrs={'class': 'input', 'placeholder':
'URL'}), required=True)
    location = forms.CharField(widget=forms.TextInput(attrs={'class': 'input', 'placeholder':
'Address'}), required=True)

    class Meta:
        model = Profile
        fields = ['image', 'first_name', 'last_name', 'bio', 'url', 'location']



class UserRegisterForm(UserCreationForm):
    username = forms.CharField(widget=forms.TextInput(attrs={'placeholder': 'Username',
'class': 'prompt srch_explore'}), max_length=50, required=True)
    # username = forms.EmailInput(widget=forms.TextInput(attrs={'placeholder':
'Username'}), max_length=50, required=True)

    email = forms.EmailField(widget=forms.TextInput(attrs={'placeholder': 'Email', 'class':
'prompt srch_explore'}))
```

```
    password1 = forms.CharField(widget=forms.PasswordInput(attrs={'placeholder': 'Enter
Password', 'class': 'prompt srch_explore'}))
    password2 = forms.CharField(widget=forms.PasswordInput(attrs={'placeholder':
'Confirm Password', 'class': 'prompt srch_explore'}))
    # email = forms.EmailField()

    class Meta:
        model = User
        fields = ['username', 'email', 'password1', 'password2']
```

CHAPTER 6

SYSTEM TESTING

6.1 TEST CASES & REPORTS

TEST CASE ID	TESTCASE/ ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAIL
1.	Selecting “CREATE PROFIE” button	Display the created profle	Display the created profle	Pass
2.	Selecting “SIGN IN” button	Display sign in page	Display sign in page	Pass
3.	Select “EDIT PROFILE”button	Display EDIT profile page	Display EDIT profile page	Pass
4.	Select “POST” button	Display Enter details for post	Enter details for post	Pass
5.	Selecting “LIKE” button	Increases like count	Increases like count	Pass
6.	Selecting “COMMENT” button	Entering the comment	Entering the comment	Pass

TEST CASE ID	TESTCASE/ ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAIL
7.	Selecting “VIEW PROFILE”	Display the user details	Display the user details	Pass
8	Selecting “ADDFRIENDS”	Display add friends form	Display add friends form	Pass
9.	Clicking the post	Display full post	Displays the post	Pass
10.	Clicking the “CHAT” button	Display chat window	Display chat window	Pass

CHAPTER 7

7.1 CONCLUSION

AI-powered social media with sentiment analysis can provide valuable insights into the emotions and opinions expressed in social media content. The platform can help organizations make informed decisions, respond to customer needs and concerns, and identify emerging trends and opportunities. The system requirements for such a platform are significant, but the improvements in performance can be significant when compared to traditional sentiment analysis methods.

.

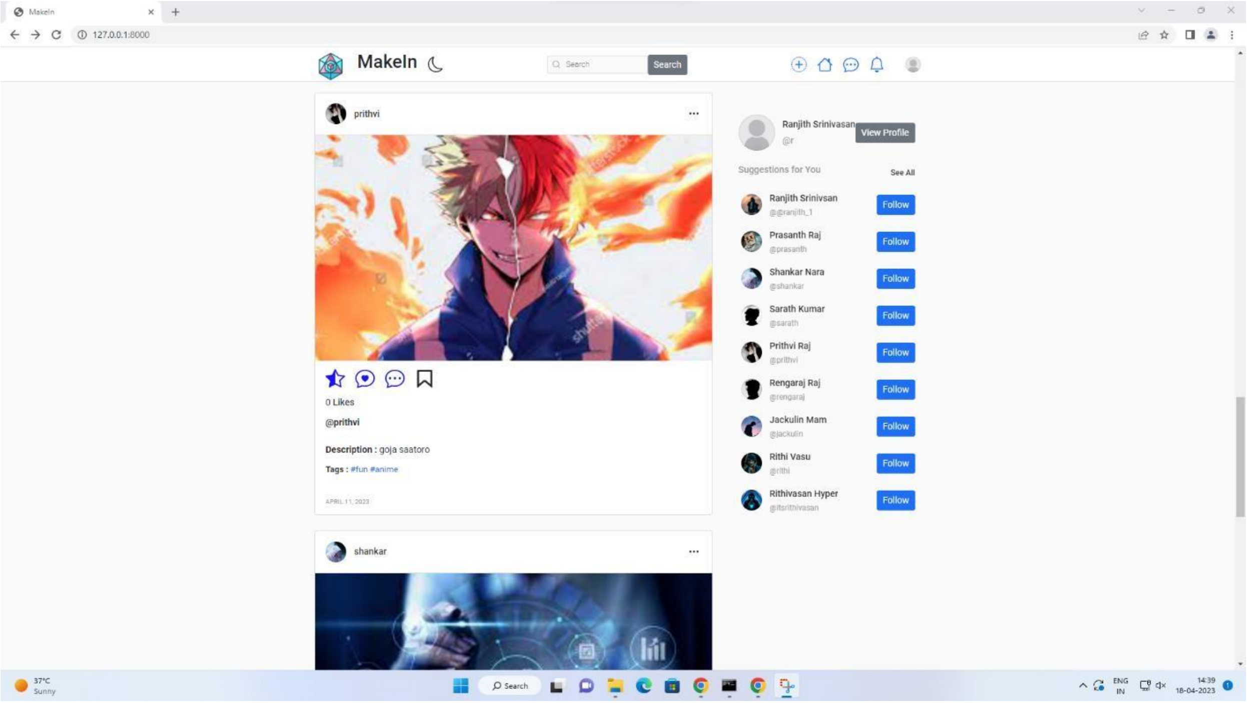
7.2 FUTURE ENHANCEMENTS

1. Content recommendation system
2. Post verification system using CNN
3. Applying advanced deep learning techniques in content creations.

The above mentioned are the future enhancements that can be done to make this project much more dynamic.

CHAPTER 8

APPENDICES



prithvi

0 Likes

@prithvi

Description : goja saatoro

Tags : #fun #anime

APRIL 11, 2023

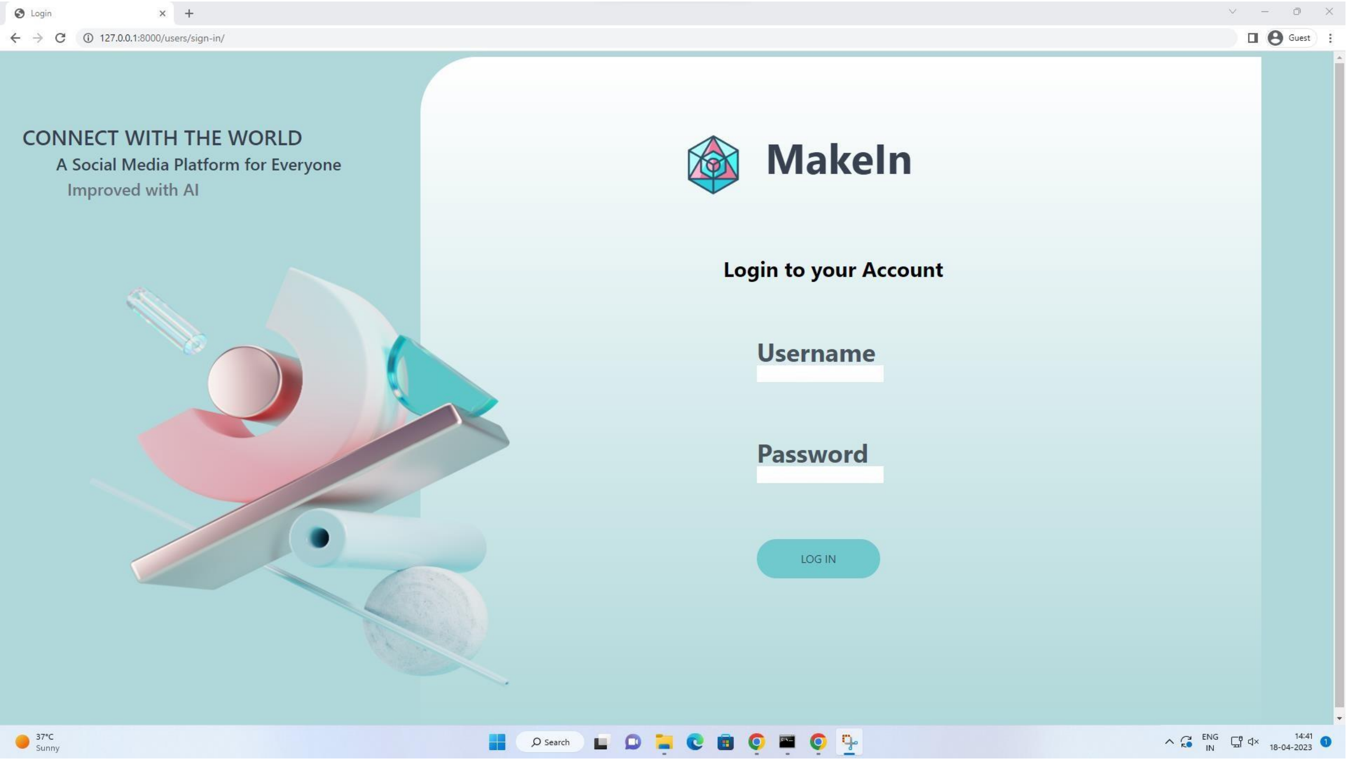
shankar

Ranjith Srinivasan
@r

Suggestions for You

Ranjith Srinivsan	@@ranjith_1	Follow
Prasanth Raj	@prasanth	Follow
Shankar Nara	@shankar	Follow
Sarath Kumar	@sarath	Follow
Prithvi Raj	@prithvi	Follow
Rengaraj Raj	@rengaraj	Follow
Jackulin Mam	@jackulin	Follow
Rithi Vasu	@rithi	Follow
Rithivasan Hyper	@itsrithivasan	Follow

Fig 8.1 Home Page



CONNECT WITH THE WORLD

A Social Media Platform for Everyone

Improved with AI

MakeIn

Login to your Account

Username

Password

Fig 8.2 Login Page

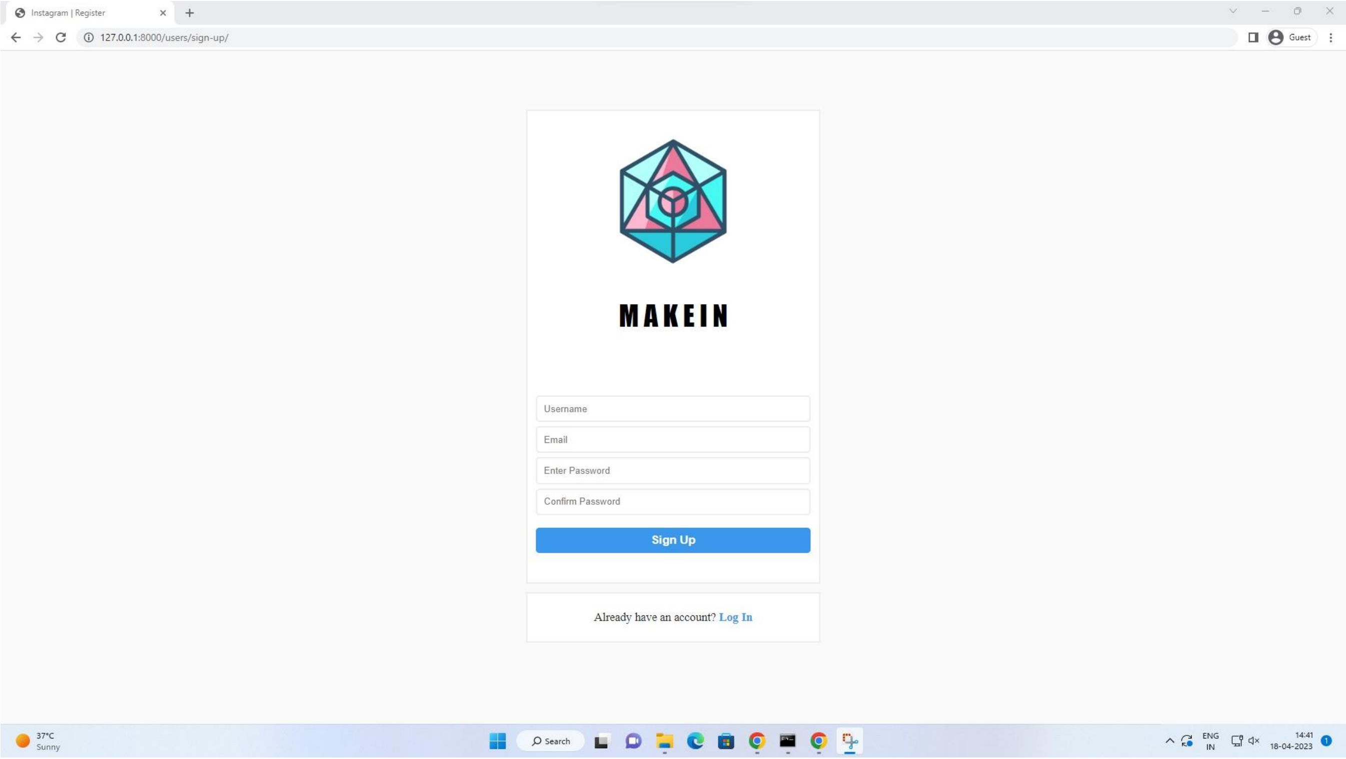
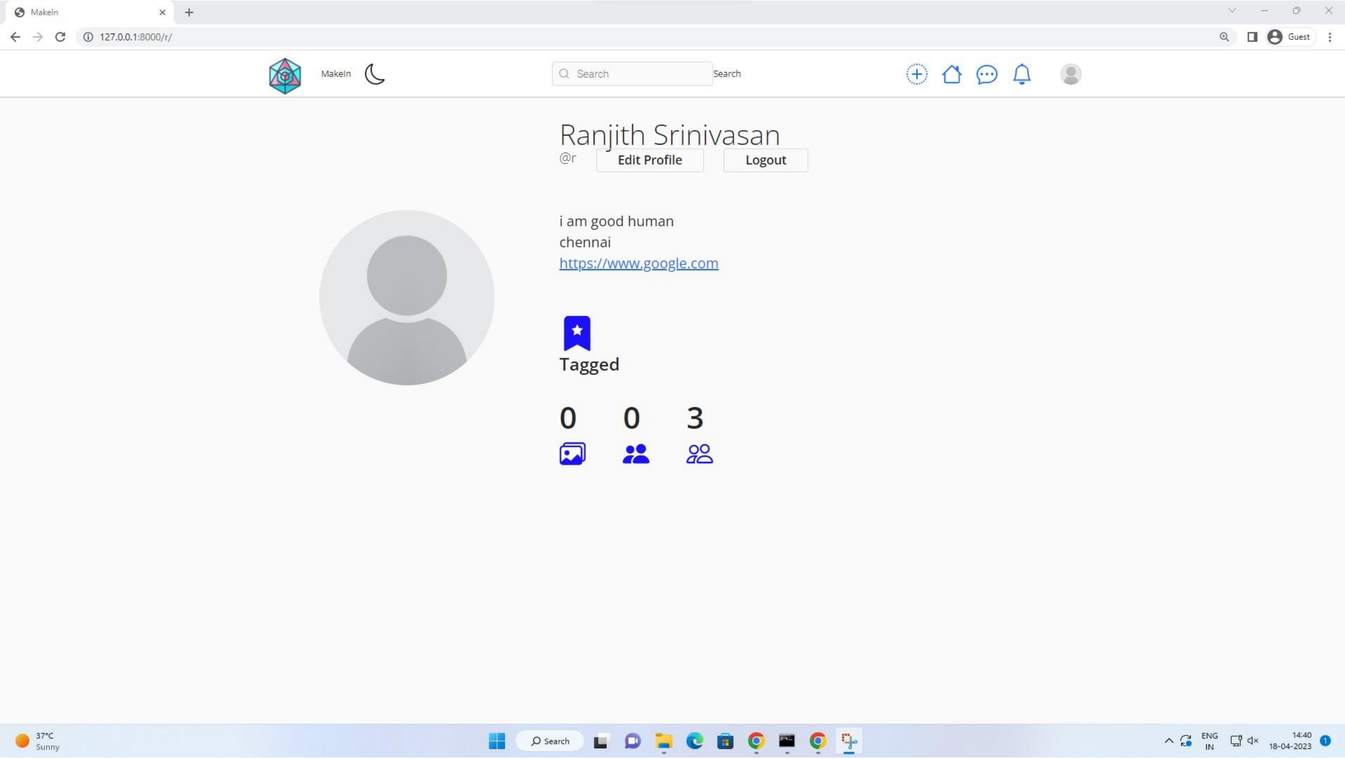


Fig 8.3 Signup Page

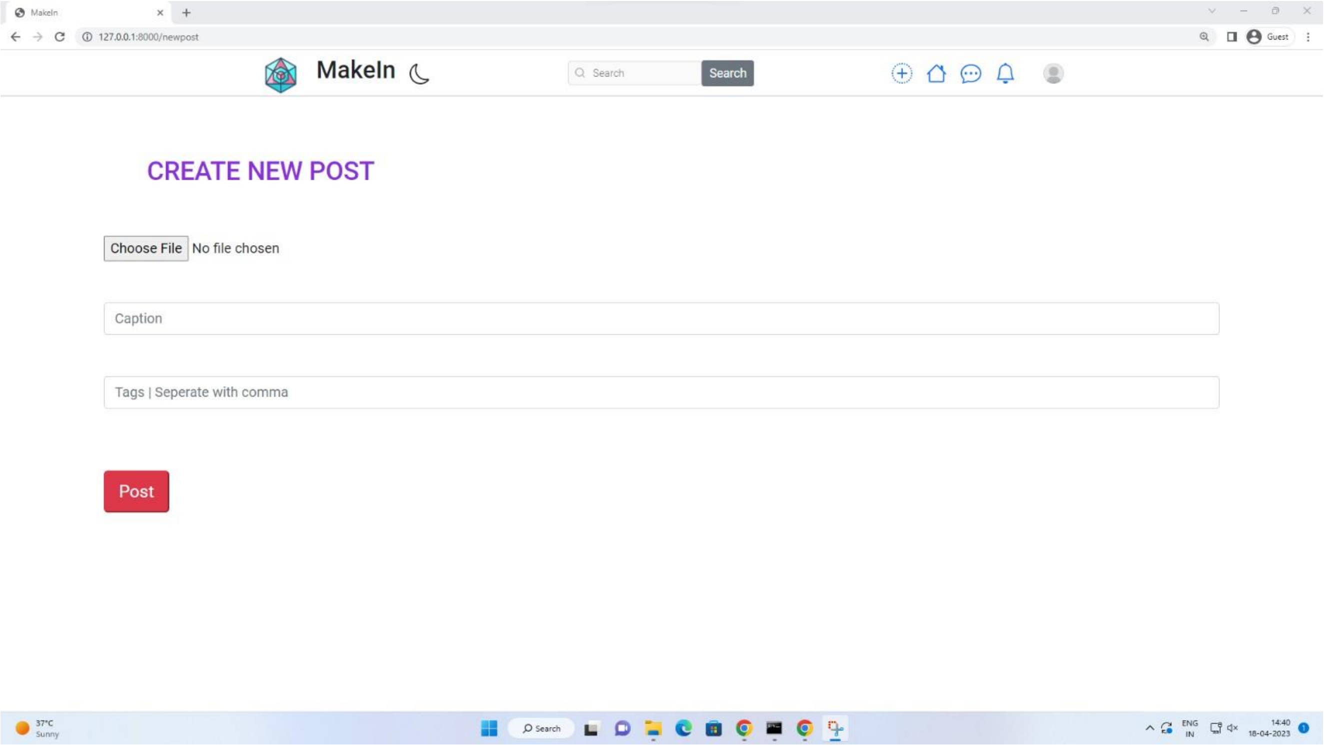


Ranjith Srinivasan
@r

i am good human
chennai
https://www.google.com

0 0 3

Fig 8.4 Profile Page



CREATE NEW POST

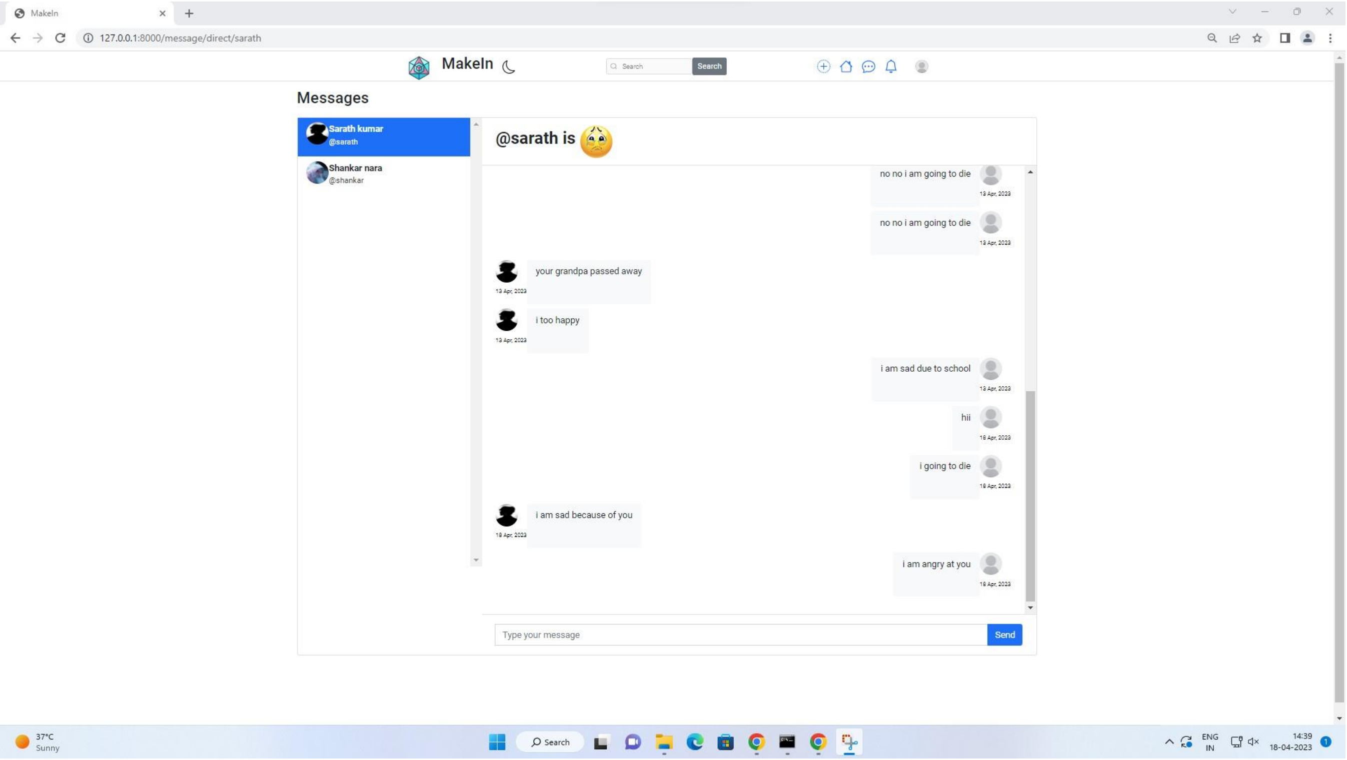
Choose File No file chosen

Caption

Tags | Seperate with comma

Post

Fig 8.5 Add post Page



Messages

Sarath kumar
@sarath

Shankar nara
@shankar

@sarath is

no no i am going to die
13 Apr, 2023

no no i am going to die
13 Apr, 2023

your grandpa passed away
13 Apr, 2023

i too happy
13 Apr, 2023

i am sad due to school
13 Apr, 2023

hii
18 Apr, 2023

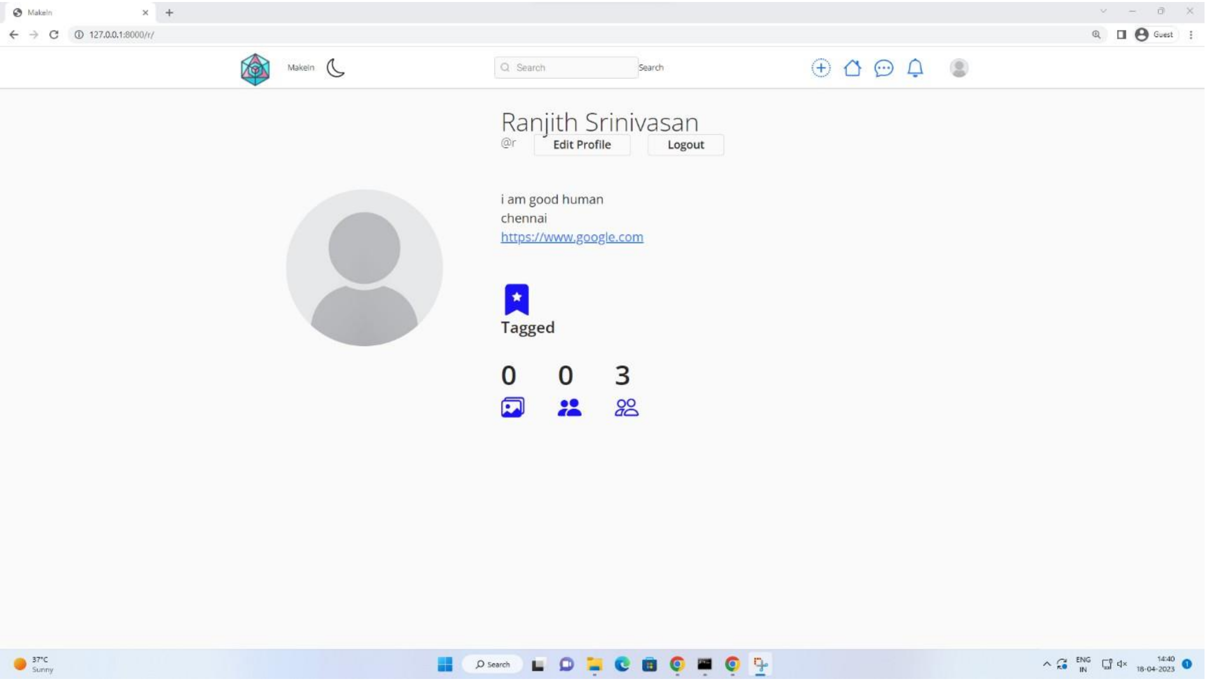
i going to die
18 Apr, 2023

i am sad because of you
18 Apr, 2023

i am angry at you
18 Apr, 2023

Type your message
Send

Fig 8.6 Chat Page



MakeIn

127.0.0.1:8000/r/

Ranjith Srinivasan

@r Edit Profile Logout

i am good human

chennai

https://www.google.com

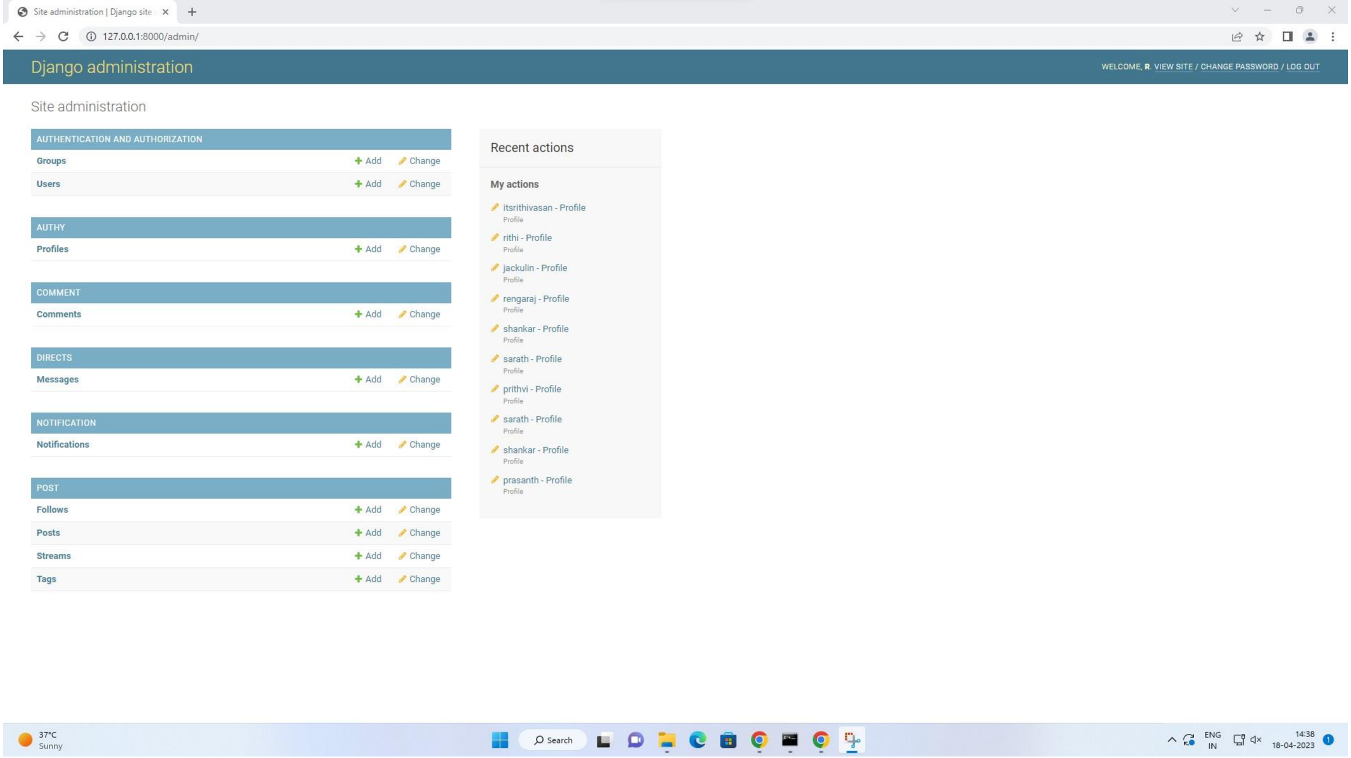
Tagged

0 0 3

37°C Sunny

14:40 18-04-2023

Fig 8.7 Post detail Page



Site administration | Django site

127.0.0.1:8000/admin/

Django administration

WELCOME, R. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change
AUTHY		
Profiles	+ Add	Change
COMMENT		
Comments	+ Add	Change
DIRECTS		
Messages	+ Add	Change
NOTIFICATION		
Notifications	+ Add	Change
POST		
Follows	+ Add	Change
Posts	+ Add	Change
Streams	+ Add	Change
Tags	+ Add	Change

Recent actions

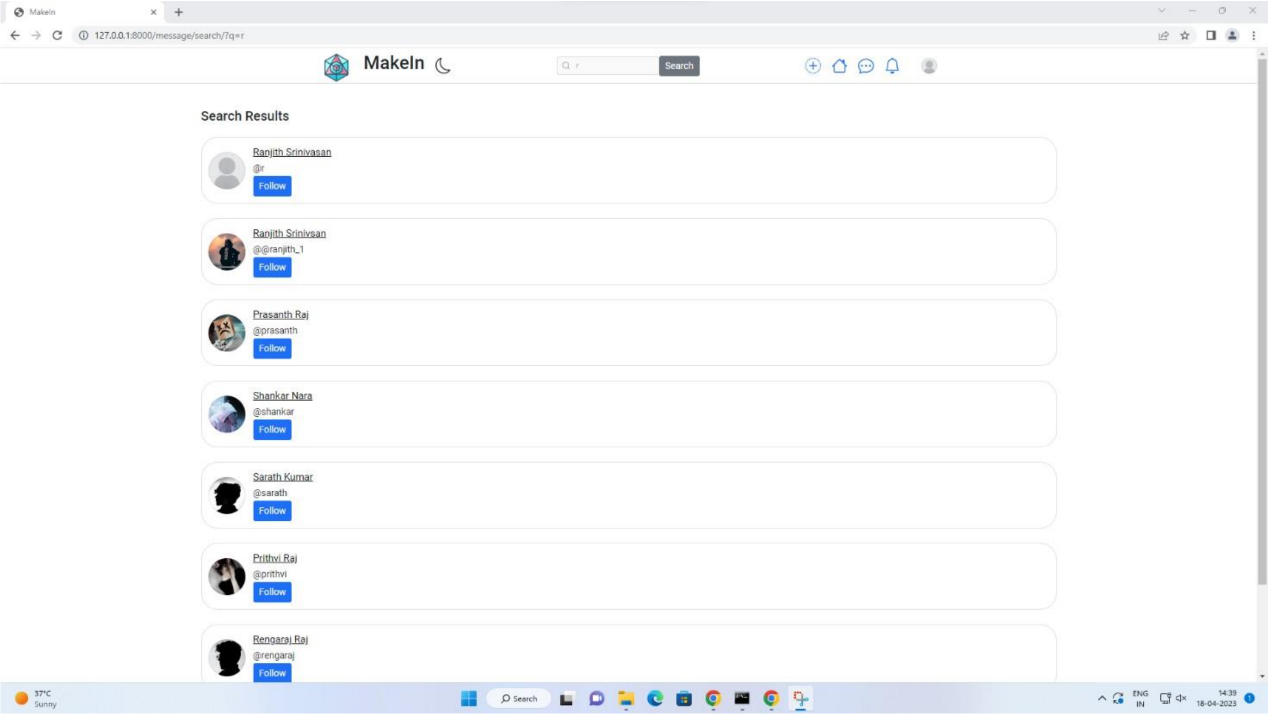
My actions

- itsrithivasan - Profile
Profile
- rithi - Profile
Profile
- jackulin - Profile
Profile
- rengaraj - Profile
Profile
- shankar - Profile
Profile
- sarath - Profile
Profile
- prithvi - Profile
Profile
- sarath - Profile
Profile
- shankar - Profile
Profile
- prasanth - Profile
Profile

37°C Sunny

14:38 18-04-2023

Fig 8.11 Admin Module



Search Results

Ranjith Srinivasan
@r
Follow

Ranjith Srinivsan
@@ranjith_1
Follow

Prasanth Raj
@prasanth
Follow

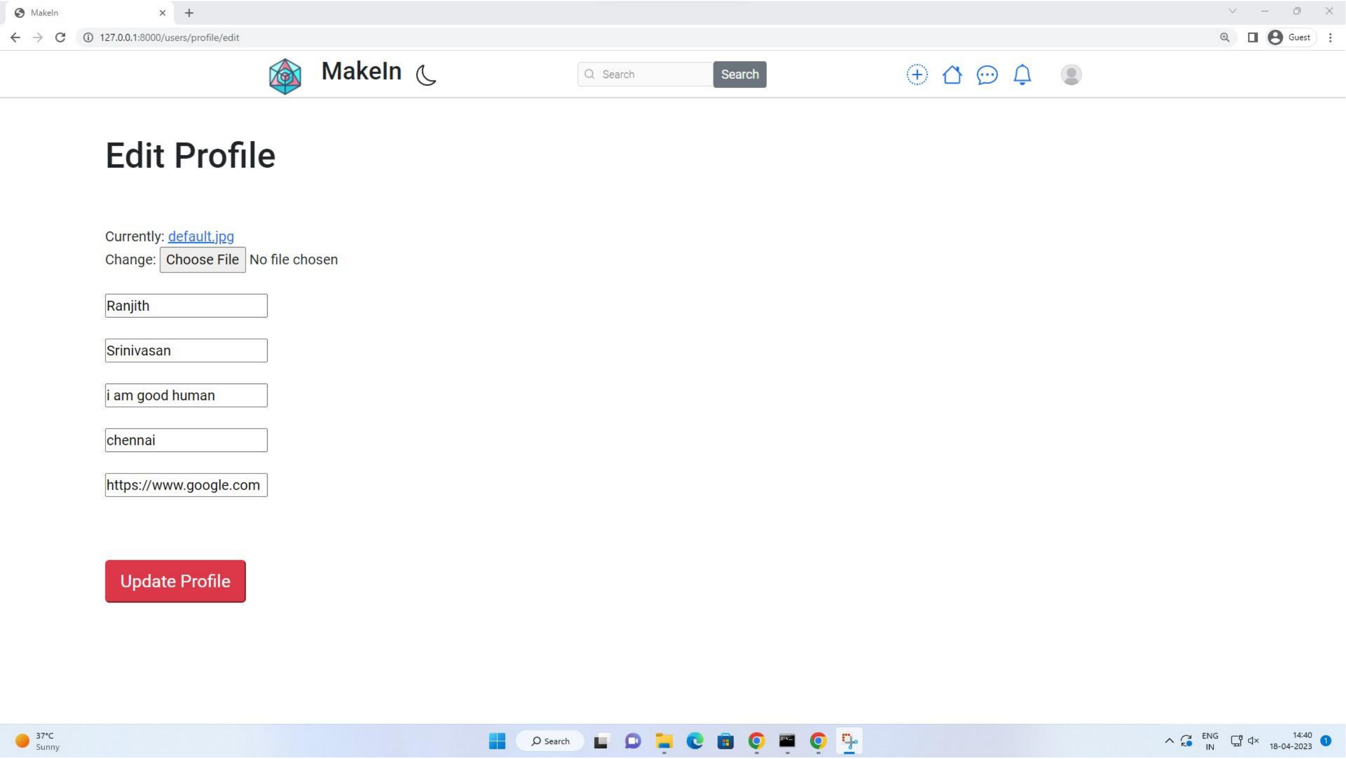
Shankar Nara
@shankar
Follow

Sarath Kumar
@sarath
Follow

Prithvi Raj
@prithvi
Follow

Rengaraj Raj
@rengaraj
Follow

Fig 8.9 Search Page



Edit Profile

Currently: default.jpg
Change: Choose File No file chosen

Ranjith

Srinivasan

i am good human

chennai

https://www.google.com

Update Profile

Fig 8.10 Edit profile Page

CHAPTER 9

REFERENCES

[1] “DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation” Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett,Xiang Gao, Jianfeng Gao, Jingjing Liu, Bill Dolan.

[2] Chao-Chun Chen; Min-Hsiung Hung; Kuan-Chou Lai; Yu-Chuan Lin, "Docker andKubernetes," in *Industry 4.1: Intelligent Manufacturing with Zero Defects* , IEEE, 2022, pp.169-213, doi: 10.1002/9781119739920.ch5.

[3] N. Marathe, A. Gandhi and J. M. Shah, "Docker Swarm and Kubernetes in CloudComputing Environment," *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI),* Tirunelveli, India, 2019, pp. 179-184, doi: 0.1109/ICOEI.2019.8862654.

[4] D. De Silva, N. Mills, M. El-Ayoubi, M. Manic and D. Alahakoon, "ChatGPT and Generative AI Guidelines for Addressing Academic Integrity and Augmenting Pre- Existing Chatbots," *2023 IEEE International Conference on Industrial Technology (ICIT),* Orlando, FL, USA, 2023, pp. 1-6, doi: 10.1109/ICIT58465.2023.10143123.

[5] E. T. G. Wang, A. P. S. Chen and C. W. Liu, "A Hybrid Evaluation of AI Chatbots in Taiwan Agriculture Services," *2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI),* Taichung, Taiwan, 2021, pp. 112-118, doi: 10.1109/TAAI54685.2021.00029.

[6] M. S. Ali, F. Azam, A. Safdar and M. W. Anwar, "Intelligent Agents inEducationalInstitutions: NEdBOT - NLP-based Chatbot for Administrative Support Using DialogFlow," *2022 IEEE International Conference on Agents(ICA),* Adelaide, Australia, 2022, pp. 30-35, doi: 10.1109/ICA55837.2022.00012.

[7] Abedi, Mahyar & Alshybani, Ibrahem & Shahadat, MRB & Murillo,Michael. (2023). Beyond Traditional Teaching: The Potential of Large Language Models andChatbots in Graduate Engineering Education. 10.48550/arXiv.2309.13059**.**

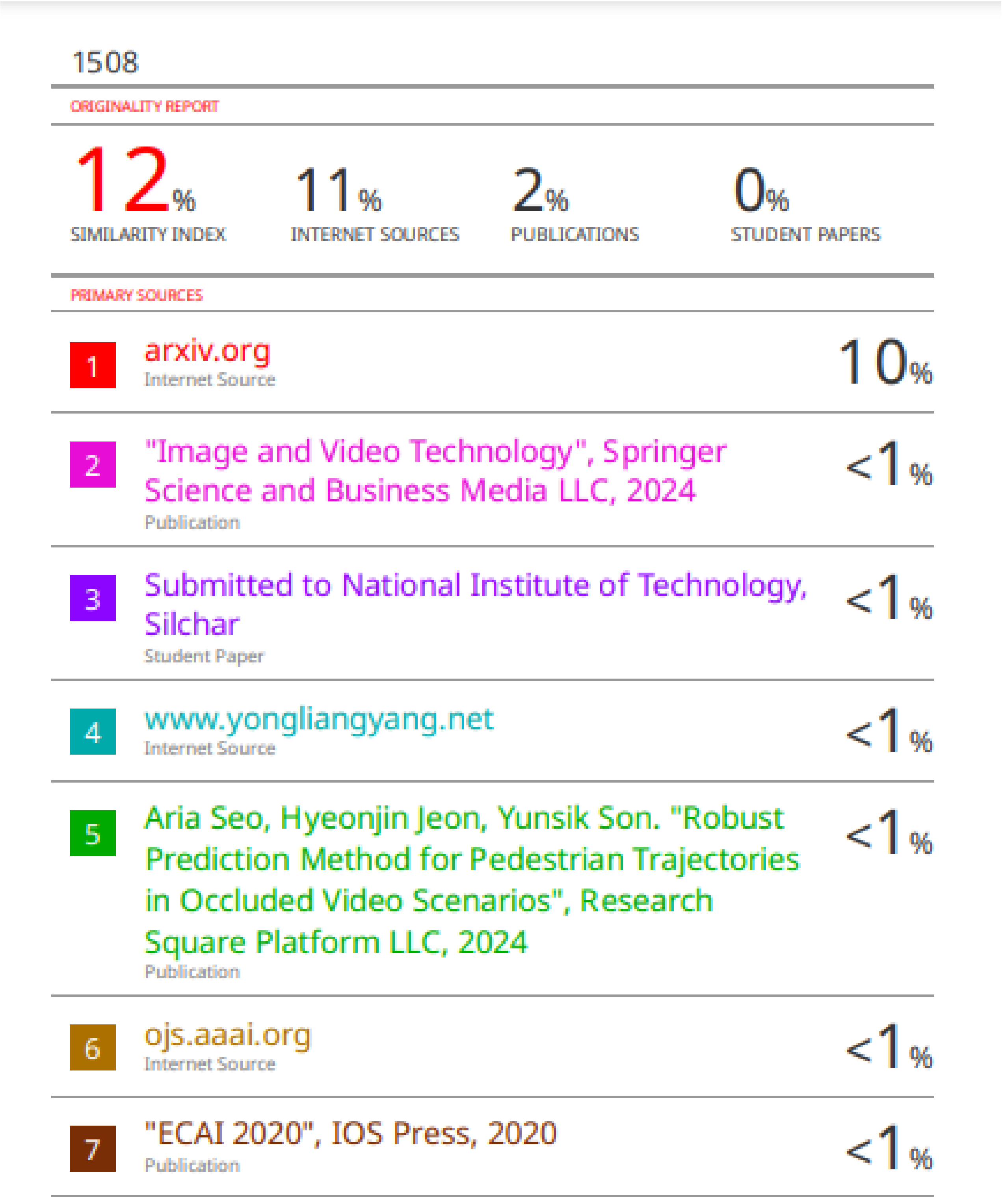
[8] A. Tesliuk, S. Bobkov, V. Ilyin, A. Novikov, A. Poyda and V. Velikhov, "Kubernetes Container Orchestration as a Framework for Flexible and EffectiveScientific Data Analysis," *2019 Ivannikov IsprasOpen Conference (ISPRAS),* Moscow, Russia, 2019, pp. 67-71, doi: 10.1109/ISPRAS47671.2019.00016.

[9] W. Wang, "Research on Using Docker Container Technology toRealize RapidDeployment Environment on Virtual Machine," *2022 8thAnnual International Conference on Network and Information Systemsfor Computers (ICNISC),* Hangzhou, China, 2022, pp. 541-544, doi: 10.1109/ICNISC57059.2022.00112**.**

[10] M. Moravcik and M. Kontsek, "Overview of Docker container orchestration tools," *2020 18th International Conference on Emerging eLearning Technologiesand Applications (ICETA),* Košice, Slovenia, 2020,pp. 475-480, doi: 10.1109/ICETA51985.2020.9379236.

CHAPTER 10

PLAGARISM REPORT



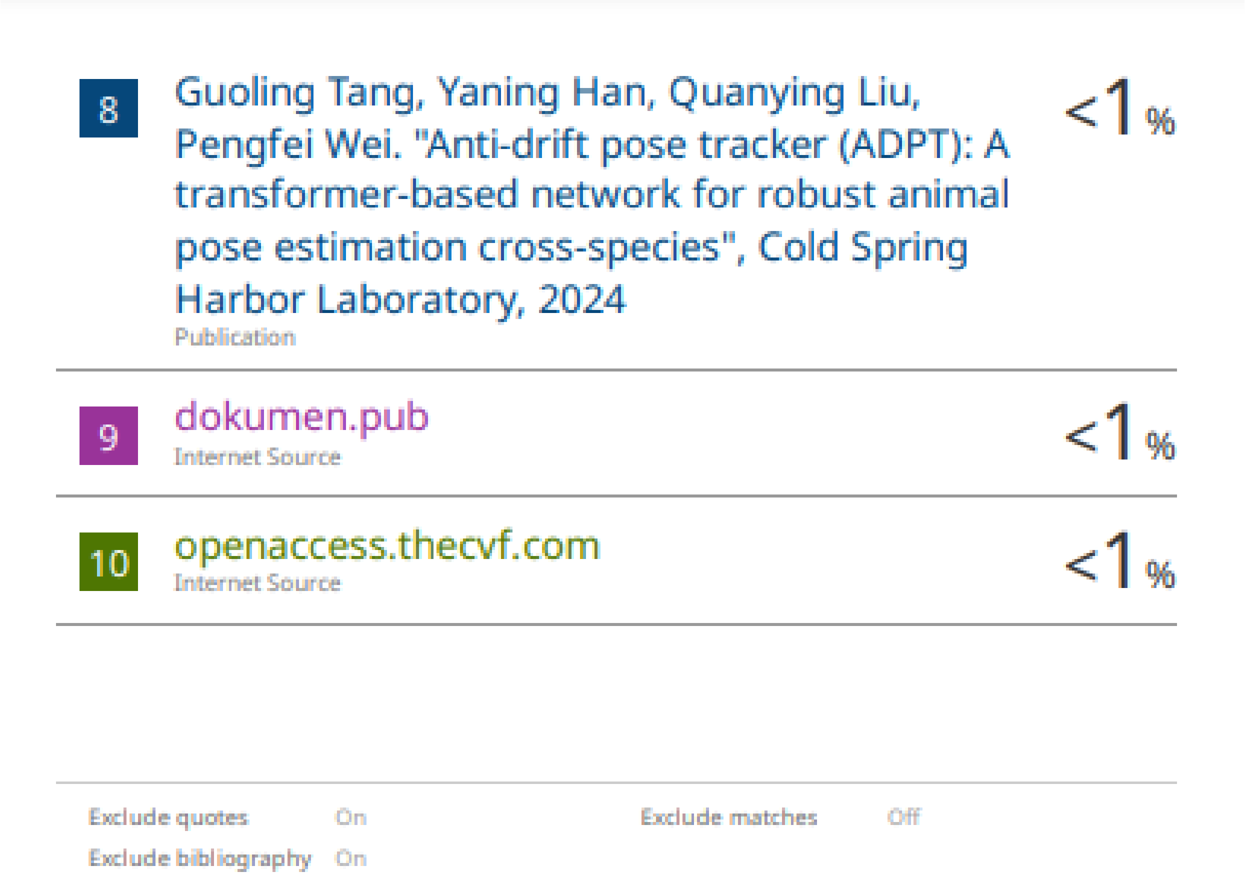
1508

ORIGINALITY REPORT

12%	11%	2%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	arxiv.org Internet Source	10%
2	"Image and Video Technology", Springer Science and Business Media LLC, 2024 Publication	<1%
3	Submitted to National Institute of Technology, Silchar Student Paper	<1%
4	www.yongliangyang.net Internet Source	<1%
5	Aria Seo, Hyeonjin Jeon, Yunsik Son. "Robust Prediction Method for Pedestrian Trajectories in Occluded Video Scenarios", Research Square Platform LLC, 2024 Publication	<1%
6	ojs.aaai.org Internet Source	<1%
7	"ECAI 2020", IOS Press, 2020 Publication	<1%



8	Guoling Tang, Yaning Han, Quanying Liu, Pengfei Wei. "Anti-drift pose tracker (ADPT): A transformer-based network for robust animal pose estimation cross-species", Cold Spring Harbor Laboratory, 2024 Publication	<1 %
9	dokumen.pub Internet Source	<1 %
10	openaccess.thecvf.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		