

**Kathmandu University**  
**Department of Computer Science and Engineering**  
**Dhulikhel, Kavre**



**A Project Report**

**on**

**“HashTag”**

**[Code No: COMP 206]**

**(For partial fulfillment of Second Year/First Semester in Computer  
Engineering)**

**Submitted by:**

**Abiral Adhikari (Roll No. 2)**

**Mahip Adhikari (Roll No. 3)**

**Ranjan Lamsal (Roll No. 26)**

**Roshan Sahani (Roll No. 42)**

**Submitted to**

**(Dr. Rajani Chulyadyo)**

**Department of Computer Science and Engineering**

**Submission Date: 23<sup>rd</sup> November ,2022**

## **Bona fide Certificate**

**This project work on  
“HASHTAG”  
is the bona fide work of  
“Abiral Adhikari, Mahip Adhikari, Ranjan Lamsal and Roshan Sahani”  
who carried out the project work under my supervision.**

**Project Supervisor**

---

**Dr. Prakash Poudyal**

**Assistant Professor & Graduate Coordinator**

**Department of Computer Science and Engineering (DoCSE)**

**Date: 23<sup>rd</sup> November 2022**

## **AKNOWLEDGEMENT**

We would like to express our sincere appreciation to our supervisor, Dr. Prakash Poudyal for his constant encouragement, guidance, understanding and suggestions during the advancement of our project, without his valuable help this project would not have been possible. We were able to elevate the idea due to his guidance. We are highly indebted to him for believing in us and for being a constant source of motivation.

We would also like to thank our project coordinator Dr. Rajani Chulyadyo for approving our project proposal, and assigning appropriate supervisor for the project.

We extend our gratitude to Kathmandu University Department of Computer Science and Engineering for providing us a platform where students can work on a topic as a semester long project and demonstrate our findings.

Abiral Adhikari

Mahip Adhikari

Ranjan Lamsal

Roshan Sahani

Date: 23<sup>rd</sup> November 2022

## **Abstract**

The contemporary system with prerequisite to follow a person behind the content to just enjoy the ideas and feed, has resulted the unhealthy trend of amassing larger number of followers for exposure, in social media. And the algorithm of social media of making trivial posts trending based on the personality creating it, has been degrading the quality of usage of social media sites like twitter, Instagram etc. So we created this project with the aim to make a social media site for sharing content, where people follow the tag, they like not the person behind the keyboard. The system of searching and following the tag they like have been implemented with hashtags. This project “Hashtag” has been developed for web using HTML, CSS, Django (Web Framework for Python), PostgreSQL (Database), React (Library of JavaScript) with Django REST API. We expect to be able to recreate “an impartial content sharing medium neutral to identity of the creator”. This report contains the detailed information regarding development and implementation of the project.

**Keywords:** Django, React, Content Sharing, Hashtag, OOP, Full stack, Social Media etc.

# Table of Contents

AKNOWLEDGEMENT	iii
Abstract	iv
List of Figures	vii
Acronyms/Abbreviations	viii
Chapter 1: Introduction	1
1.1 Background	1
1.2 Objectives	2
1.3 Motivation and Significance	2
Chapter 2: Related Works	3
Chapter 3: Planning, Design and Implementation	6
3.1 Research and Planning	6
3.2 Design	6
3.3 Implementation (Coding)	7
3.4 API and Merging:	12
3.5 Debugging and Documentation	13
3.5 User Case Scenario	13
3.6 System Requirement Specifications	14
3.6.1 Software Specifications	14
3.6.1.1 Front End Tools: React, Bootstrap	14
3.6.1.2 Back End Tools: Django, PostgreSQL	15
3.6.1.3 User /Client Software Requirements	15
3.6.2 Hardware Specifications	15
Chapter 4: Discussion on the achievements	16
Chapter 5: Features	18
Chapter 6: Conclusion and Recommendation	19

6.1 Limitations	19
6.2 Future Enhancement	19
References	20
APPENDIX	21
APPENDIX-1: GANTT CHART	21
APPENDIX-2: PREVIEW	22

## List of Figures

Figure 2.1: Twitter Home Page.....	3
Figure 2.2: Reddit Home Page.....	4
Figure 2.3: Quora Home Page .....	4
Figure 3.2.1: Workflow Design .....	7
Figure 3.3.1: System Flowchart .....	9
Figure 3.3.2: HashTag Login Page .....	10
Figure 3.3.3: HashTag Signup Page .....	11
Figure 3.3.4: Backend Code in IDE.....	12
Figure 3.5.1: User Case Scenario Diagram.....	14
Figure 7.1: Gantt Chart .....	21
Figure 8.1: Homepage 1.....	22
Figure 8.2: Homepage 2.....	22
Figure 8.3: Homepage 3.....	23
Figure 8.4: Homepage 4.....	23
Figure 8.5: Follow HashTag Page .....	24

## **Acronyms/Abbreviations**

HTML:	Hyper Text Markup Language
CSS:	Cascaded Style Sheet
SQL:	Structured Query Language
REST:	Representational State Transfer
API:	Application Programming Interface
OOP:	Object-Oriented Programming
UI:	User Interface
JS:	JavaScript
URL:	Uniform Resource Locator
JSON:	JavaScript Object Notation
IDE:	Integrated Development Environment
PC:	Personal Computer
CRUD:	Create, Read, Update, and Delete
JWT:	JSON Web Token



# **Chapter 1: Introduction**

Project 'Hashtag' is a social networking site for content sharing which aims "to bridge the gap between like-minded people based on their thoughts" expressed in the form of tags.

## **1.1 Background**

One of the earliest iterations of knowledge sharing took form of caved drawings in 15,000 BC [1]. Concept of sharing ideas and arts have existed since ancient time. In ancient Greece: philosophers held open ideological debates in senates and amphitheaters. Ancient China and India held events for idea discussions in courts too. This can be considered the earliest form of content sharing.

The recognizable social media site, Six Degrees was created in 1997. It enabled users to upload a profile and make friends with other users. After, 1999 the blogging sites became popular creating social media sensation. In 2006, with YouTube and Twitter becoming popular the new era of social media networking started. The social networking industry boomed with new additions like Facebook, Pinterest etc. Nowadays the social media sites like Instagram and Twitter are on the top of the game which enables posting of videos, photos, comments etc. along with communication services.

But even though these social networking sites have developed over the course of time increasing the reach of human speech and thoughts, it isn't without its own set of problems. In recent years these sites have become platform for unhealthy tussle between people, spreading propaganda, inciting masses with hate speech and misinformation. Though the technology behind the industry is well developed but the major drawback of the existing work lies in the fact: "The reach of content in the existing sites are affected by in life media presence of the account holder creating the content rather than quality of content." Moreover, existing algorithm shows trending content to user rather than what user wants at that instance. With

this project we are trying to give users a stock minimalistic social media experience. We aim to bring out the unappreciated beautiful content into the limelight often untouched due to terrible algorithms.

## **1.2 Objectives**

The objectives of this project work are:

- To create a proper minimalistic social media platform allowing users to post the content and ideas.
- To create a creator independent content following mechanism.
- To create the search and feed algorithm based on hashtags only.

## **1.3 Motivation and Significance**

During my visits and searches in social media I found that posts of the certain celebrities had massive views just because of the massive fan following. Sometimes I was even shown posts that I had no interest in just because they were trending. Time and again I found I had no options to choose what I want to see as the algorithm will just show me content, I based on my previous consumption of content either keeping me in a bubble or showing me entirely irrelevant content. So, with this project we aim to create a system where people are shown content of their choosing based on the tags they are following. So, as that only tags, they follow will tailor their experience. This project gets rid of the follower system for person and create a platform for sharing views impartially. The project will use tags created by user for searching, viewing and creating post.

After discussing this problem, we came to conclude that one of the efficient way to solve is it “Making communication between content and consumer independent of creator’s charisma” via the chain of #hashtags. Thus protecting the quality of content from undue influence and interference.

## Chapter 2: Related Works

There are social media that use hashtags and have content sharing features like Facebook, TikTok, Instagram, Reddit etc. and our project has subtle resemblance to Twitter, Reddit and Instagram more towards Twitter and Reddit.

Twitter is a microblogging and social networking site that started in 2006[2]. It is the precursor to the present-day hashtag culture. But Twitter has failed to explore the possibilities with their use of hashtags which we want to work upon. Twitter has similar mechanism of posting tweets which we will be improvising.

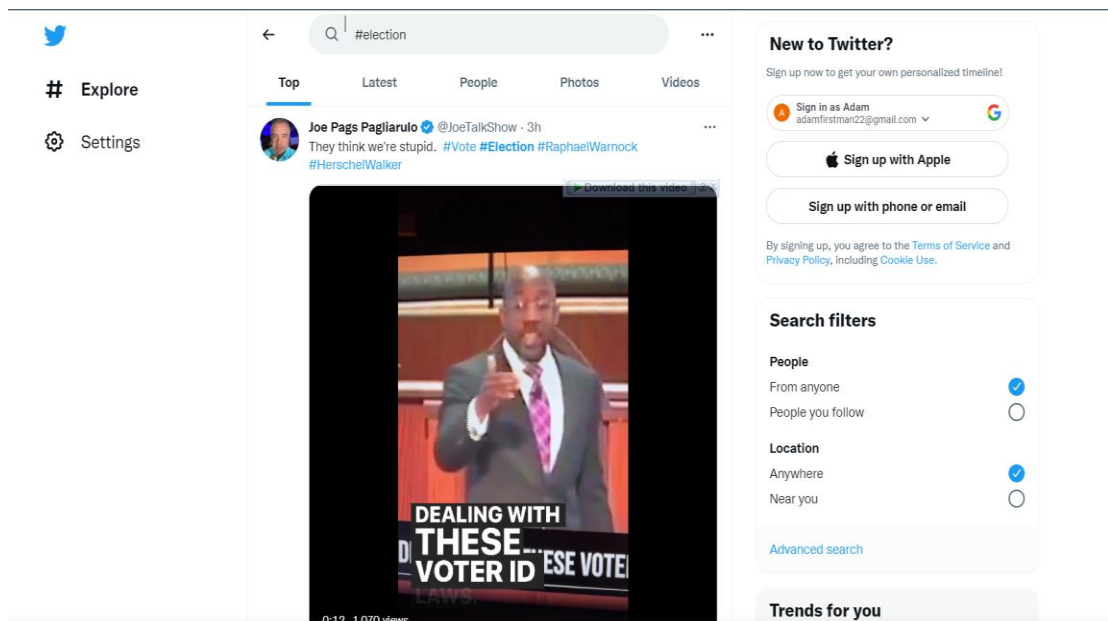


Figure 2.1: Twitter Home Page

Reddit is an American social news aggregation, content rating, and discussion website established in 2005 [3]. The community of Reddit and its diversified application was an eye catcher. Reddit has feature comment threading and compartmentalization of content using subreddits.

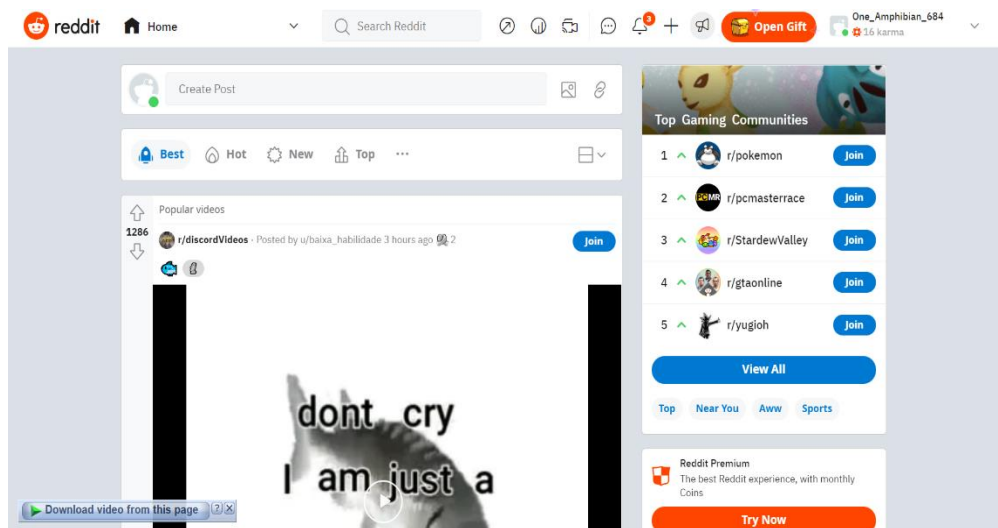


Figure 2.2: Reddit Home Page

Moreover, its minimalistic UI with system threads based on SQLite and Python has acted as a point of interest for us. The design of the project has vague similarity to Quora and its user feed UI.

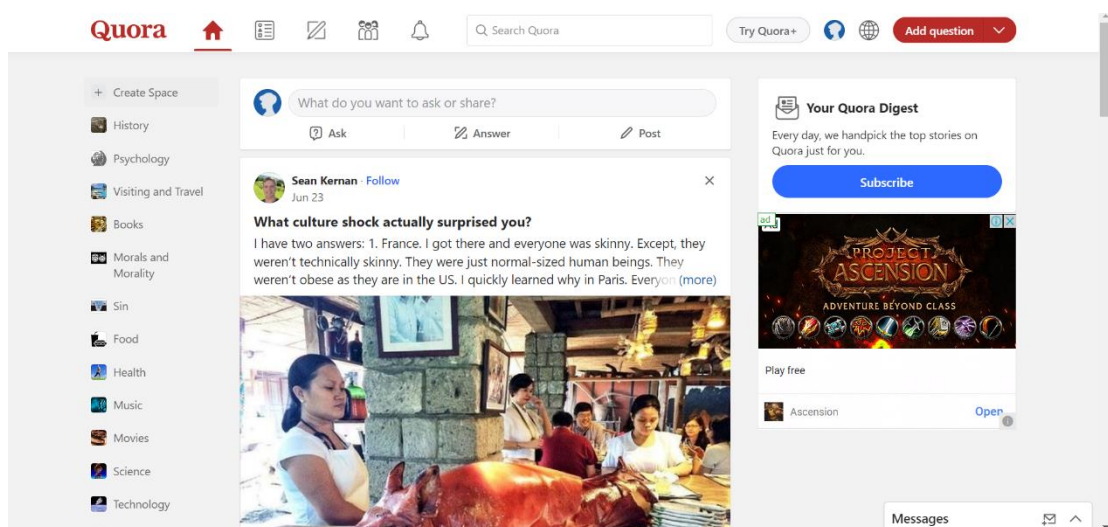


Figure 2.3: Quora Home Page

All these blogging, networking and posting sites have few of the features we want but not all of them integrated in a single platform, which we want to achieve from our site.

## **Chapter 3: Planning, Design and Implementation**

The project was to be completed in below mentioned stages:

### **3.1 Research and Planning**

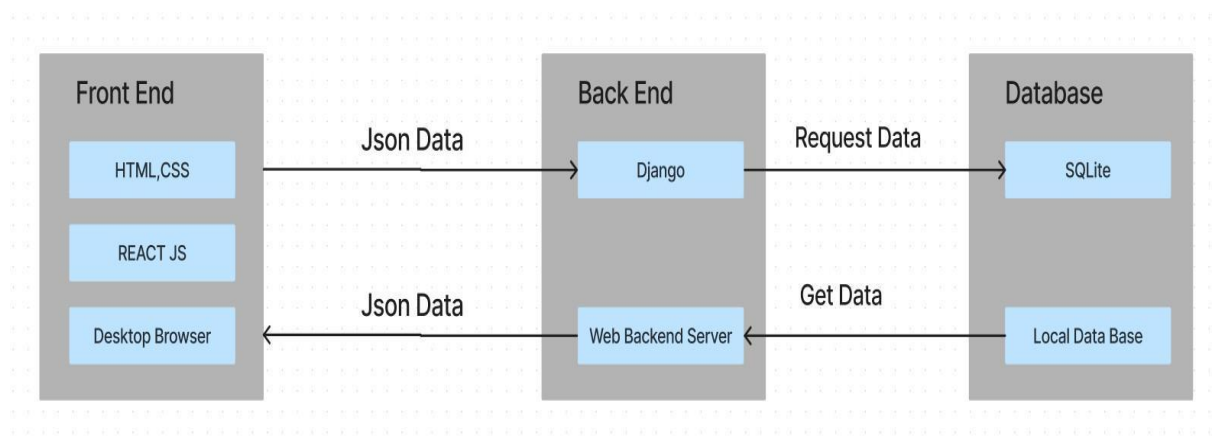
After the first meeting with our supervisor Mr. Prakash Poudyal we discussed possibilities for this project and usability of tools we mentioned in the proposal. After a week of research on web and on recommendation of our supervisor and seniors, we decided to complete this project using HTML, CSS and JavaScript's REACT library for frontend web development, Python's Django Web Framework for Backend, Django REST framework to create API and PostgreSQL for database. We initially planned to deploy the database on cloud using Heroku.

### **3.2 Design**

This project is a full stack web application. The design for the working of the system was as following divided as:

- Frontend
- Backend
- API and Database

The basic diagram on design of the system is provided below:



**Figure 3.2.1: Workflow Design**

The procedure/ methods which was implemented in the project are described below and the detailed discussion on the completion of the project is provided in next section.

### 3.3 Implementation (Coding)

While coding we found multiple ways of implementing the backend logic using Django like ModelForms, ModelClass, etc. We decided to use Model (Class) method of code in Django for the reusability purpose along with serializers and deserializers. As the language and framework were new to us the working were full of hit and trials. We started with basic concepts of designing in front end using Figma, HTML, CSS and JS(REACT). In the backend we utilized Django class and templates.

Our system has two servers running one for the frontend and another for backend which communicate with one another using endpoints. Endpoints are the address (URLs) in the system to request data between two ends. The data packages used in the projects are JSON and JSON strings, which are converted one to another using deserializes and serializers respectively. The Authentication and validation modules were used as provided by Django.

The procedure for implementation didn't go as smoothly as planned. We faced a few hiccups along the way especially in regard to database usage. We were unable to use Heroku as previously planned due to free services of Heroku being cut down. So we settled with the local database server of Django. We have used the Serializers for ingesting and getting data instead of forms. We are using a token for authentication in our project.

The model system flow chart is provided in next page:



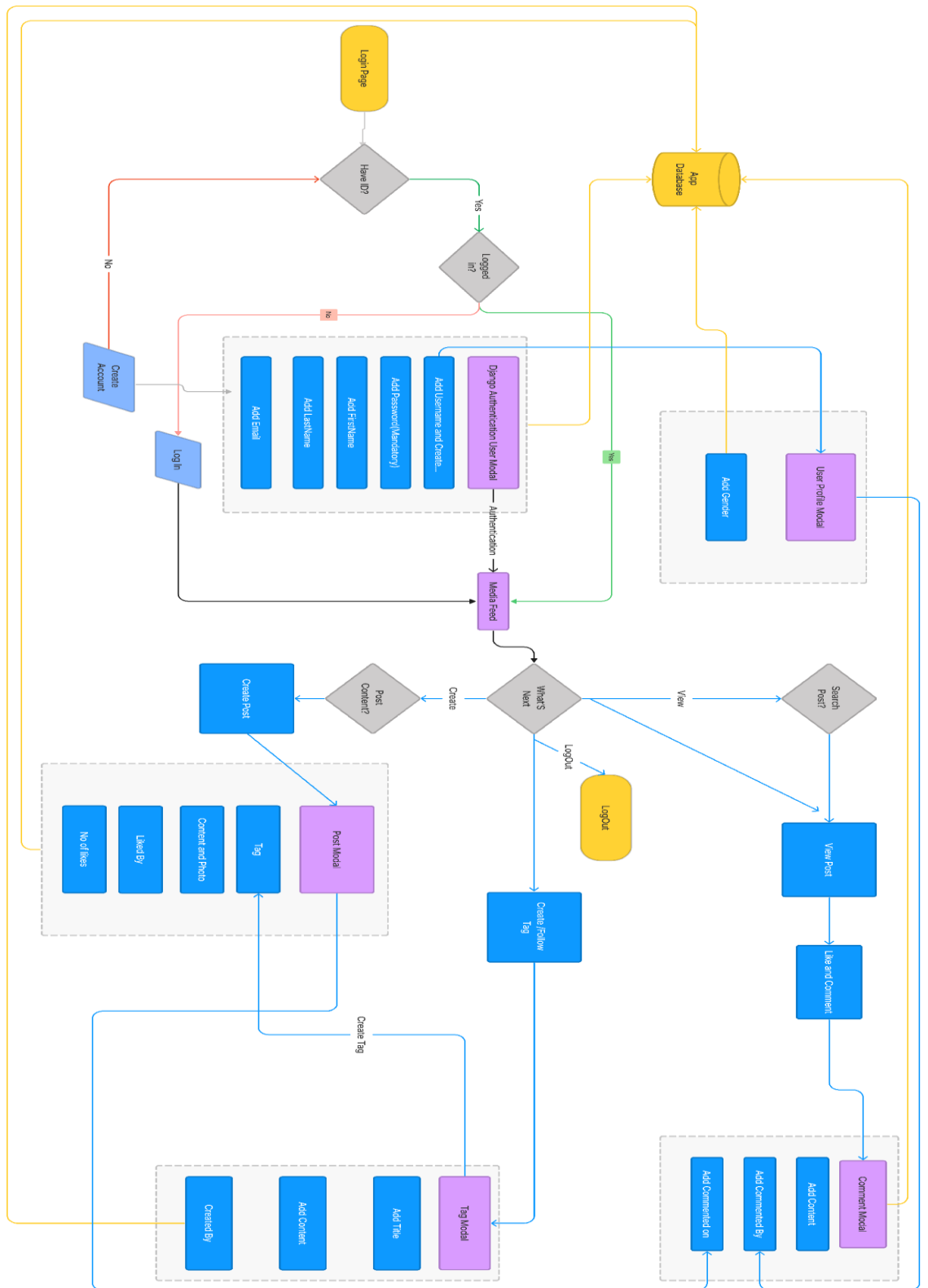
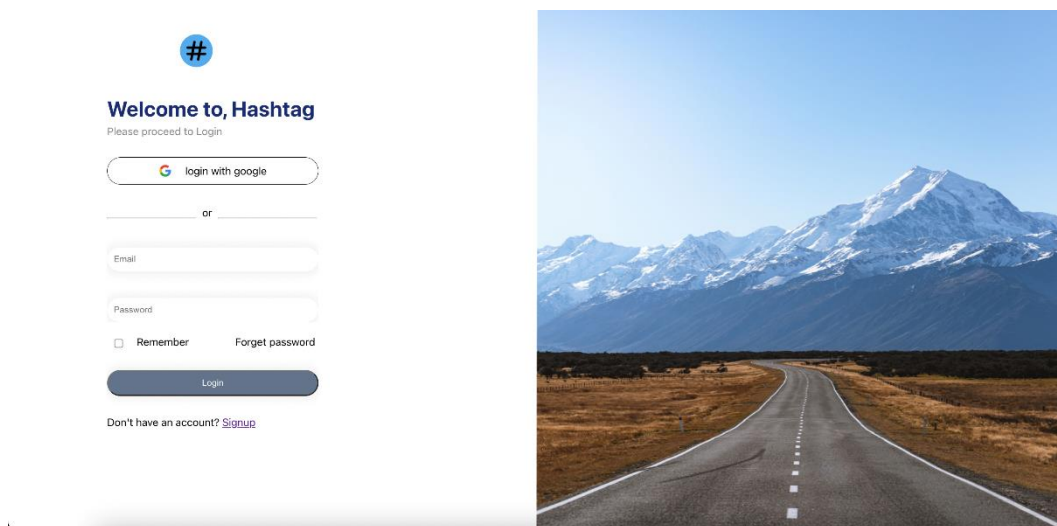
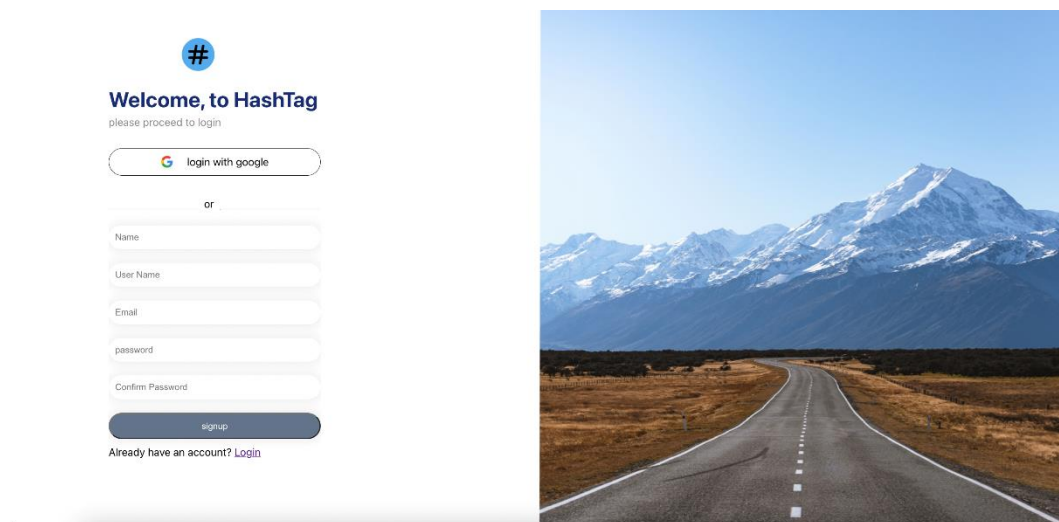


Figure 3.3.1: System Flowchart

Client Side of the web development refers to everything that is displayed on the screen of the client (user end device). The client end design is done for creating UI, and all the other things that users see. The client side of our project includes a login page for signup and login, a home page for viewing feed and posting and other pages generated via templates as per the need of the user. The snippets of Client Side Development of project is shown in figure below:



**Figure 3.3.2: HashTag Login Page**



**Figure 3.3.3: HashTag Signup Page**

The Server Side of Web Development refers to all the work done on data that is displayed to the user as asked by the user. It includes storing, deleting and updating the information in the database; the backend logic controlling permissions and validators, and all other shenanigans from and until the task requested by client end is completed. Our Django project contains apps like: Users (for maintaining information of user), PostApp (for posting a new content by user), Tag (for creating tag and storing information of followers of tag), Comments (for keeping information regarding comments like what, where and by whom) etc. We have used validators provided by Django to validate username and password. The snippet of IDE of Server Side Coding is provided below:

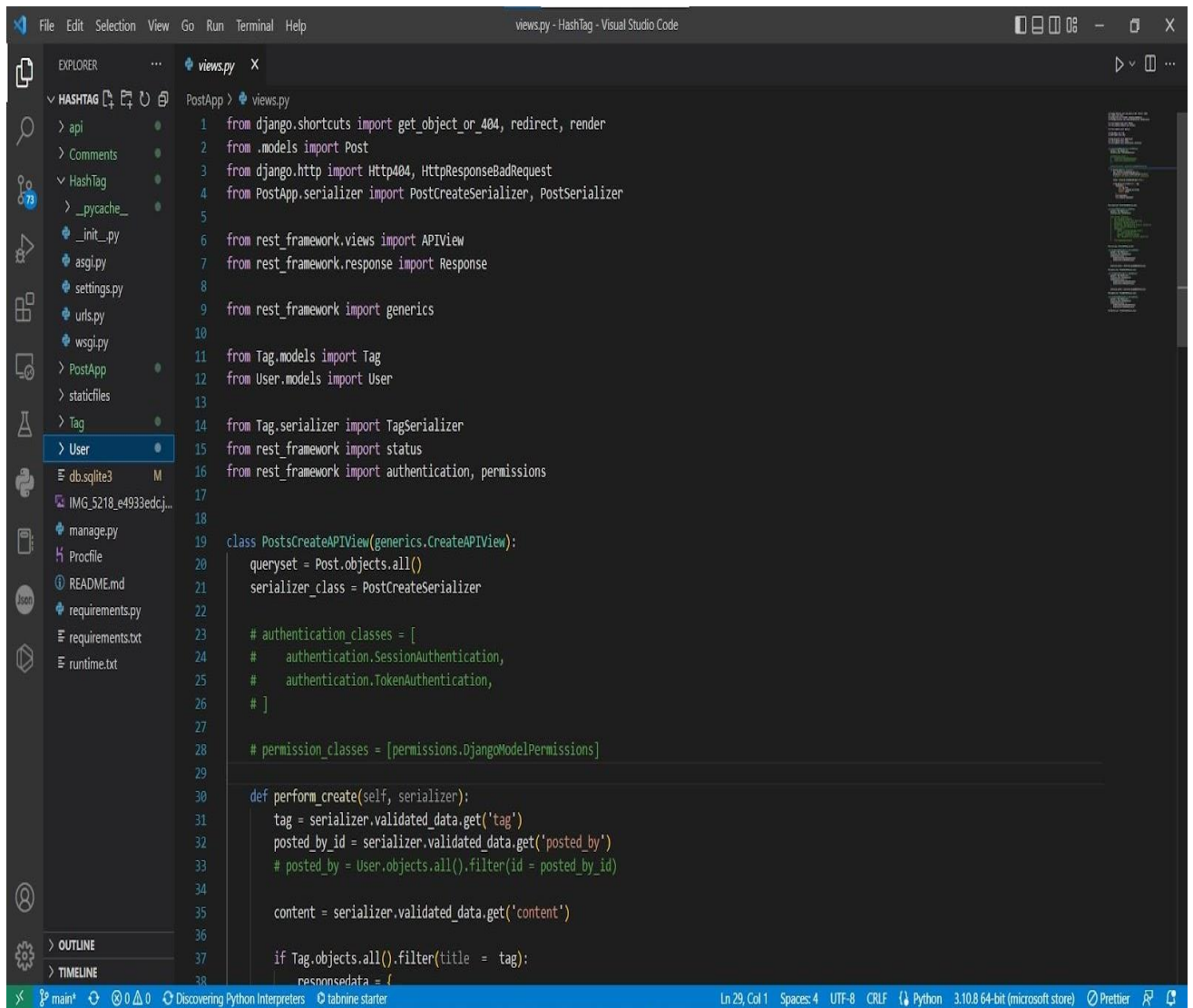


Figure 3.3.4: Backend Code in IDE

### 3.4 API and Merging:

We have used REST framework for building web API for our project. The REST framework is used along the blueprint of Django. For the API building we made use of the tools like: Serializers, Validators, Authentication, Permissions and Generic Views. Serializers allow complex data such as query sets and model instances to be converted to native Python data types that can then be easily

rendered into JSON. Validators of REST framework help in field validation. Authentication is the mechanism of associating an incoming request with a set of identifying credentials, such as the user the request came from, or the token that it was signed with. Together with authentication, permissions determine whether a request should be granted or denied access. The Generic Views provided by REST framework allowed us to quickly build API views that map closely to your database models and also being class based views they were reusable.

### **3.5 Debugging and Documentation**

After the merging of the frontend and backend code with the help of API in JSON format, we started debugging and documentation of application simultaneously in later weeks. We found few bugs in the signup and login validation code which was debugged later on. We started the documentation of code and report writing as the standard template provided.

### **3.5 User Case Scenario**

In the application, the user is at first asked to login, if not logged in. The user either creates a new id to login or uses the previous username and password to login. While creating the new user account the user is asked for valid unique username and password along with email, first name and other details. The user after logging in is welcomed with a home page containing feed based in his follows. Then the user can either see, like and comment on the post created by other users; or create their own post by creating a new tag or using an already existing tag. The user can further search or follow tags to view related content. And at the end of the session they logout.

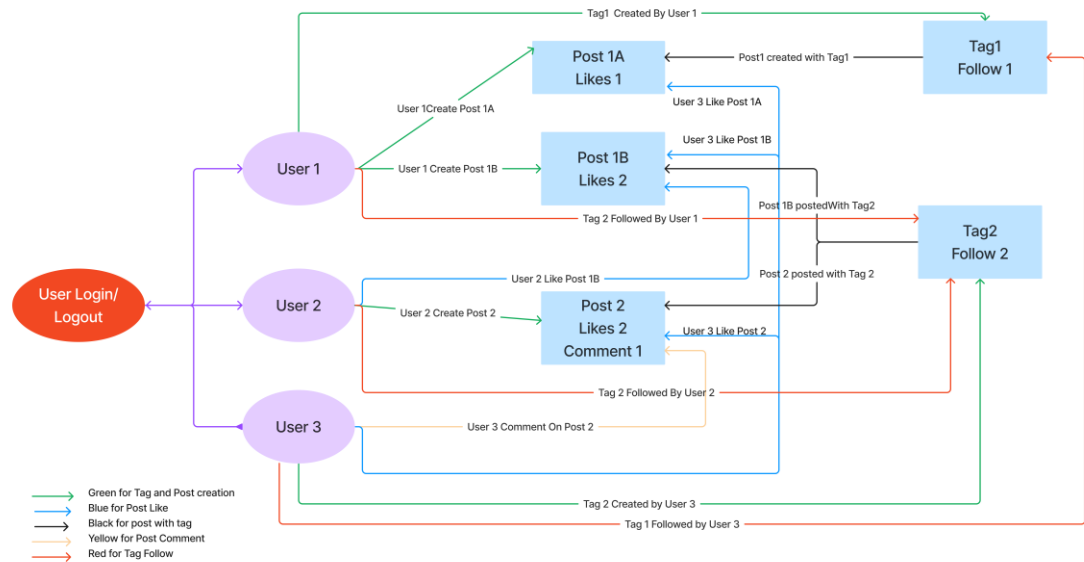


Figure 3.5.1: User Case Scenario Diagram

## 3.6 System Requirement Specifications

### 3.6.1 Software Specifications

The software specification is different for client and administrator(developer).

#### 3.6.1.1 Front End Tools: React, Bootstrap

React is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single page applications. [4]

Bootstrap is a free and open-source CSS framework directed at responsive, mobile first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. [5]

### **3.6.1.2 Back End Tools: Django, PostgreSQL**

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. [6] Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

PostgreSQL is a general purpose and object-relational database management system, the most advanced open-source database system. PostgreSQL is free and open-source software.[7] PostgreSQL requires very minimum maintained efforts because of its stability. Therefore, if you develop applications based on PostgreSQL, the total cost of ownership is low in comparison with other database management systems.

### **3.6.1.3 User /Client Software Requirements**

For this web application to run the user will need a web browser in a computer connected to the server hosting the site. This section presents the description of the developed software system. While the requirements for the provider run the code is provided within the project folder in 'requirements.txt' file.

### **3.6.2 Hardware Specifications**

The only hardware required to run this application is a well-functioning computer, though the quality of use may depend upon the condition of the PC.

## **Chapter 4: Discussion on the achievements**

Even though we tried our best to complete the project as stated on the proposal we weren't able to do so as we faced a few problems along the way. The initial problem that we faced was we were all new to the programming language used. And even though there were numerous tutorials and documentation available to us we were not able to fully utilize them at a high level of efficiency due to our inexperience. In the second week of the project, we studied the design of static pages Using CSS and HTML. In this week we also learned about the working of the web system and call methods especially get and post. We also found that the way of accessing and retrieving data using URLs on the web was different from that of the local file system which we used in the previous semester project.

In the next few weeks, we completed the design of the login page and the homepage of the application. On the other hand, we also worked on CRUD (Create, Read, Update and Delete) / data manipulation through function calls Views. Furthermore, we started learning about models in Django and began the creation of user models. After this we learned about generic views in the Django REST framework. We also found out the use of generic views provided by Django to be better practice after spending some time on exploring other options. We then implemented the use of validators for registration of new users. During this period, we faced a few problems related to the database as few errors arose on the database model.

Here we deviated from our original objective and were not able to deploy databases online using Heroku. There was a problem with Heroku ending its free database deployment services which it had been providing for some time. We were incapable of using cloud storage. Though use of the local database is a downgrade, we used what was available to us.

After this deviation, we began creating the generic class views used for creating, updating, retrieving and deleting data. We also created custom validators redefining them for user login and signup. The next step in our work was creation of a Tag



model that allowed the user to create tags on which content would be posted. This time we are finished the design of navigation bar of the webpage. After this week, the month-long vacation started and slowing us down as we were not able to do parallel work due to our lack of availability at the same time.

Now with responsive and dynamic pages remaining in the design front, we increased our pace. In the server side we created the comment and post models using the previously developed user and Tag model. Begin working on permission via user authentication based on the View Class. Finally, during the last leg of the time provided to us who designed a responsive and dynamic feed bar of the homepage. We were finally working on generation and utilization of JWT token thus redefining user permissions. We then started debugging of the project solving bugs in the login and signup function. Finally, we started the standard documentation and report writing as per the template provided to us. The debugging of project is still on the go.

At the end of this project we are able to learn the new programming languages JS and Django(Python) after using C++ in the previous semester, for which we are thankful to the university for including this subject and project in our syllabus. Upon the discussion with Prakash Sir we found suggestion to use machine learning algorithms in the project to be productive but we were not able to utilize it due to lack of sufficient time. During this entire time the availability of multiple solutions to a problem sometimes made our work easier while at other times made it more confusing. As, we were constantly facing new problems after solving the previous one; we were also improving our skill and proficiency. So, all in all were able to benefit a lot from this project.

## Chapter 5: Features

The features of our project are discussed below:

- User can create an account with a unique username and alphanumeric password.
- Password is tested with validators to have at least 8 characters, uncommon and non-fully numerical. The validators prevent the user from having password similar to user name.
- User can create tags relevant to a topic used to create posts either by themselves or other user using the tag.
- User can post, edit and delete their post which can be liked and commented upon by other user.
- There is dynamic admin area to manage the staff accounts and inappropriate posts
- User can further search and find relevant posts based upon the tags they are following.

## **Chapter 6: Conclusion and Recommendation**

At the end of the project, we managed to learn the new programming languages Python and JavaScript's along with their selected frameworks and libraries: Django and REACT. We have designed a content sharing site where users can create and follow topics they like in the form of tags. We have made it possible for users to create posts and share content specifying the relevant tags. Thus we have managed to achieve the above mentioned objectives.

### **6.1 Limitations**

Even Though we have achieved our main objectives our project does have some limitations and room for improvement. First of such limitation being the local hosted database rather than a cloud based database. Other limitations of our project include no filtration algorithm for comment and content posting which might be misused and lack of a well-developed recommendation algorithm.

### **6.2 Future Enhancement**

In the coming future if we were to work on this project we plan on using Machine Learning algorithm in conjunction with Prakash sir to further enhance the user experience. We also plan to let users know who viewed their content, not just the information of who is following their tag. There many more room for improvement which we can incorporate after further discussion in the time to come.

## References

- [1]. Hill, J. (2017, November 20). The History of Knowledge Sharing. Bloomfire. Retrieved November 23, 2022, from <https://bloomfire.com/blog/history-of-knowledge-sharing/>
- [2]. Twitter.com (2006). About Twitter | Our company and priorities. Retrieved November 23, 2022, from <https://about.twitter.com/en>
- [3]. Reddit.com (2005). Reddit: Homepage. Retrieved November 23, 2022, from <https://www.redditinc.com/>
- [4]. React JS. *Getting Started – React*. React. Retrieved November 23, 2022, from <https://reactjs.org/docs/getting-started.html>
- [5]. *Get started with Bootstrap · Bootstrap v5.2*. Bootstrap. Retrieved November 23, 2022, from <https://getbootstrap.com/docs/5.2/getting-started/introduction/>
- [6]. Django documentation | Django documentation. Retrieved November 23, 2022, from <https://docs.djangoproject.com/en/4.0/>
- [7]. PostgreSQL. *About PostgreSQL*. Retrieved November 23, 2022, from <https://www.postgresql.org/about/>

## APPENDIX

### APPENDIX-1: GANTT CHART

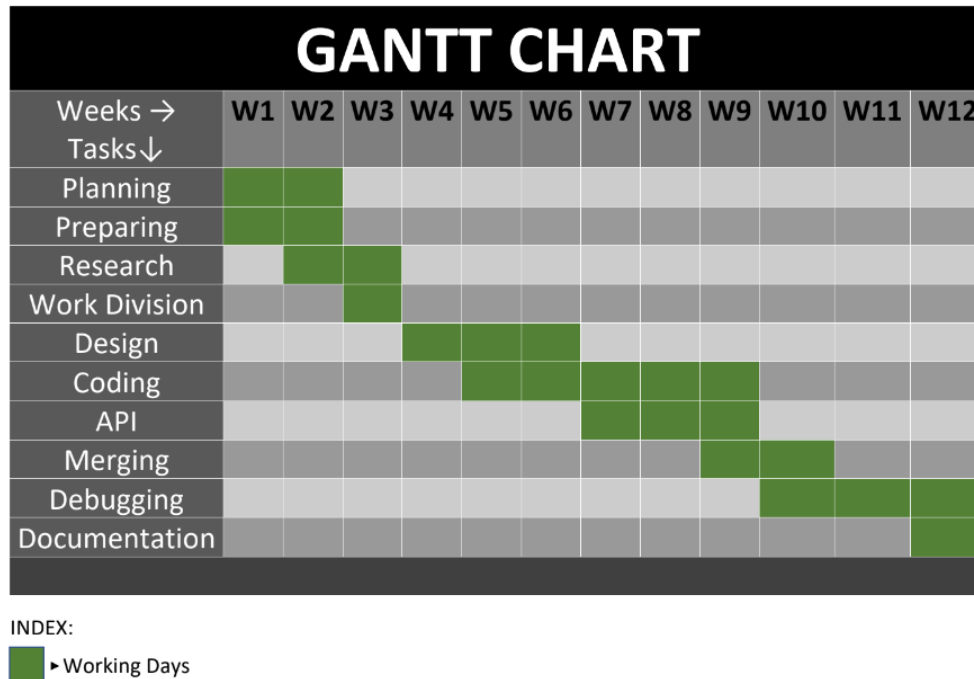


Figure 7.1: Gantt Chart

## APPENDIX-2: PREVIEW

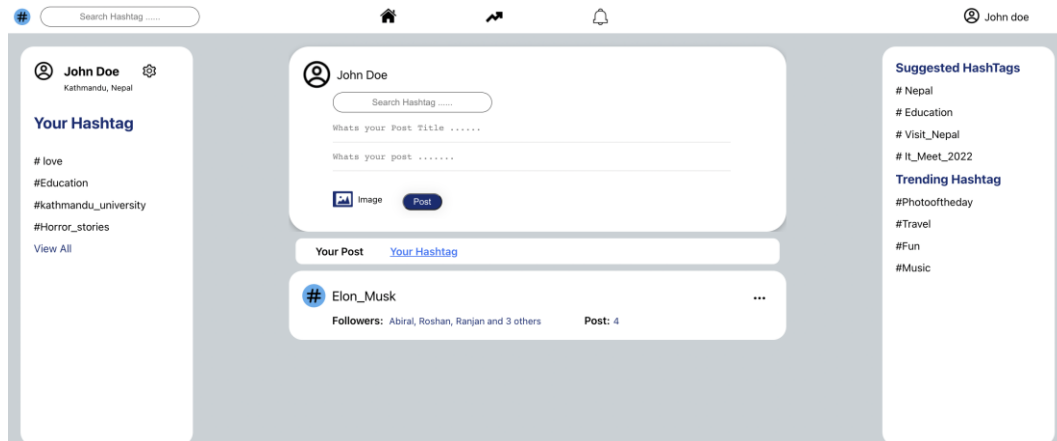


Figure 8.1: Homepage 1

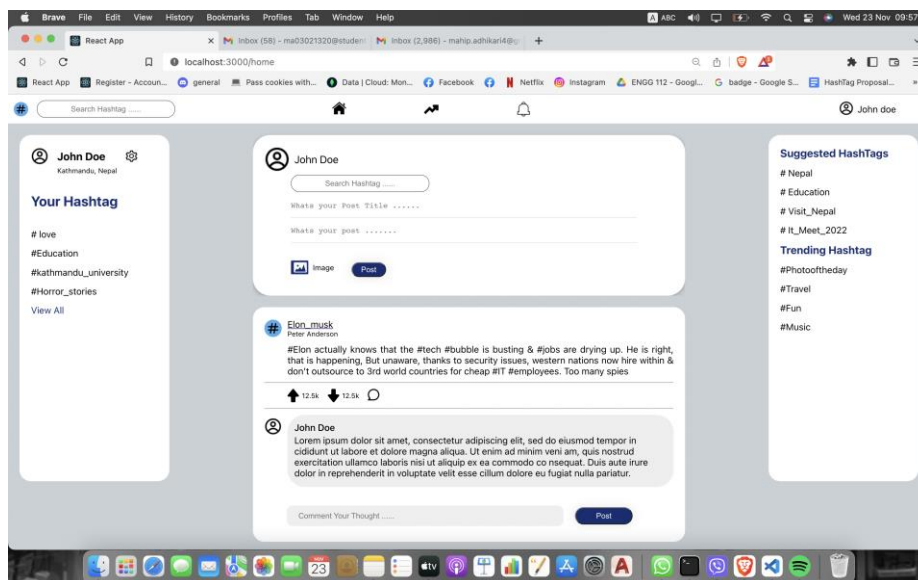


Figure 8.2: Homepage 2

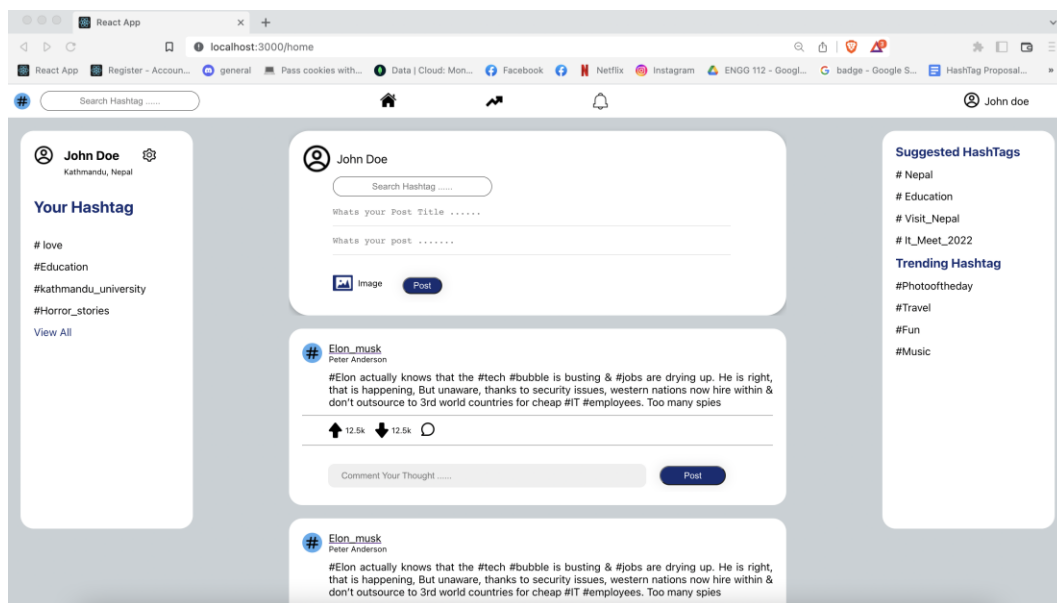


Figure 8.3: Homepage 3

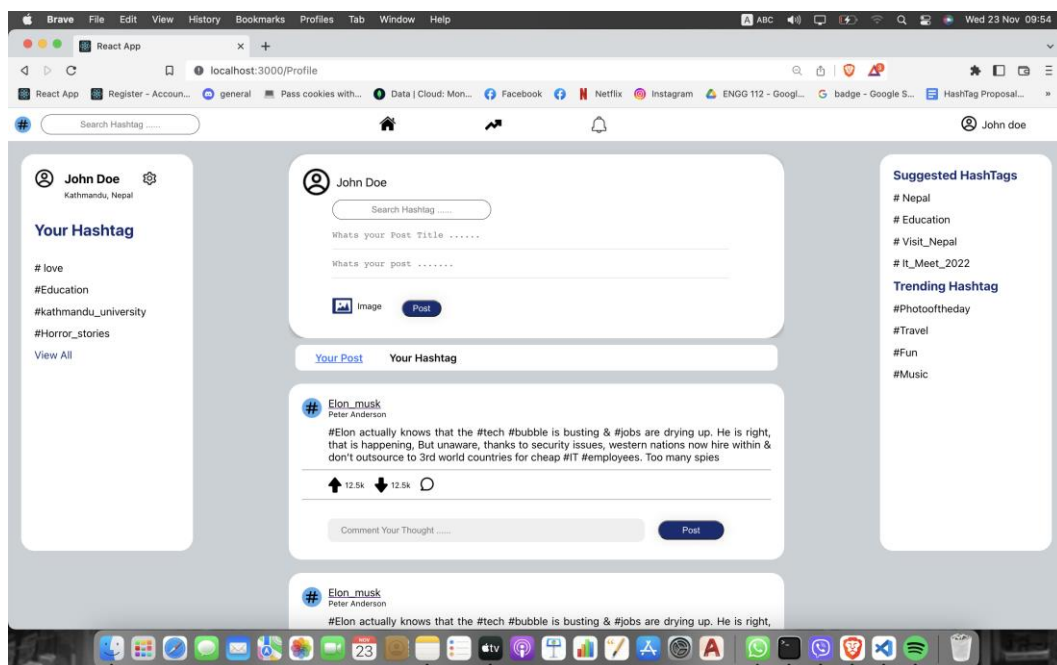


Figure 8.4: Homepage 4

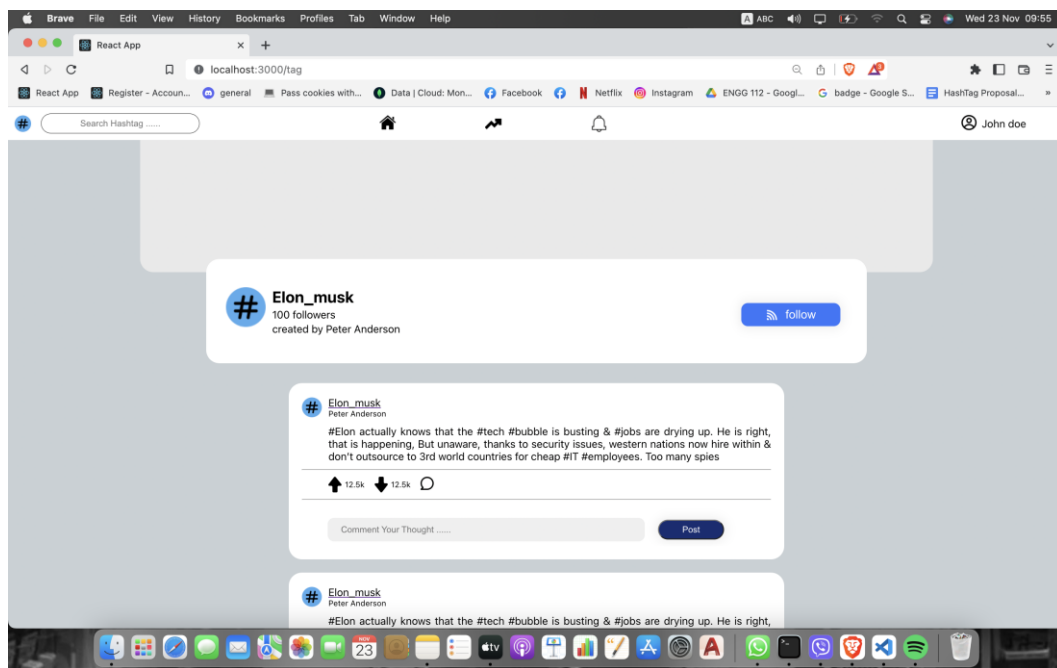


Figure 8.5: Follow HashTag Page