Django Blog Web-Application

Project Report Submitted in Partial Fulfilment of the Requirements for
the Degree of

Bachelor of Engineering
in
Computer Science & Engineering

Submitted by

Raghav Khandelwal: (Roll No. 19UCSE4015)

Sagar Sharma: (Roll No. 19UCSE4018)

Under the Mentorship of
Dr. Shrawan Ram
Assistant Professor

&

Under the Guidance of
Mr. Abhishek Gaur
Assistant Professor

Department of Computer Science & Engineering
MBM University, Jodhpur

July, 2022

This page was intentionally left blank.

Django Blog Web-Application

Project Report Submitted in Partial Fulfilment of the Requirements for the Degree of

Bachelor of Engineering
in
Computer Science & Engineering

Submitted by

Raghav Khandelwal: (Roll No. 19UCSE4015)

Sagar Sharma: (Roll No. 19UCSE4018)

Under the Mentorship of
Dr. Shrawan Ram
Assistant Professor

&

Under the Guidance of
Mr. Abhishek Gaur
Assistant Professor

Department of Computer Science & Engineering
MBM University, Jodhpur

July, 2022

This page was intentionally left blank.

Department of Computer Science & Engineering

M.B.M. University,
Ratanada, Jodhpur, Rajasthan, India –342011

CERTIFICATE

This is to certify that the work contained in this report entitled **"Django Blog Web-Application**" is submitted by the group members Mr. Raghav Khandelwal (Roll. No: 19UCSE4015) and Mr. Sagar Sharma, (Roll. No: 19UCSE4018) to the Department of Computer Science & Engineering, M.B.M. University, Jodhpur, for the partial fulfillment of the requirements for the degree of **Bachelor of Engineering** in **Computer Science & Engineering**.

They have carried out their work under my guidance. This work has not been submitted elsewhere for the award of any other degree or diploma.

The project work in our opinion has reached the standard fulfilling the requirements for the degree of Bachelor of Engineering in Computer Science in accordance with the regulations of the Institute.

Mr. Abhishek Gaur
Assistant professor
(Guide)
Dept. of Computer Science & Engg.
M.B.M. University, Jodhpur

Dr. Shrawan Ram
(Mentor)
Dept. of Computer Science & Engg.
M.B.M. University, Jodhpur

This page was intentionally left blank.

DECLARATION

we, ***Raghav Khandelwal and Sagar Sharma,*** hereby declare that this project titled **"*Django Blog Web-Application*"** is a record of original work done by us under the supervision and guidance of ***Mr. Abhishek Gaur***.

We further certify that this work has not formed the basis for the award of the Degree/Diploma/Associateship/Fellowship or similar recognition to any candidate of any university and no part of this report is reproduced as it is from any other source without appropriate reference and permission.

raghav

SIGNATURE OF STUDENT

(Raghav Khandelwal)
8th Semester, CSE
Enroll. - 18r/22913
Roll No. - 19UCSE4015

sagar

SIGNATURE OF STUDENT

(Sagar Sharma)
8th Semester, CSE
Enroll. - 18R/06201
Roll No. - 19UCSE4018

This page was intentionally left blank.

ACKNOWLEDGEMENT

We take the opportunity to express our gratitude to all who have provided their immense support and their valuable time in guiding us. Due to their support, Guidance, Supervision, and Encouragement, we have successfully completed our Project Report on "***Django Blog Web-Application***". We are highly indebted to our guide "***Mr. Abhishek Gaur, Assistant Professor"*** and mentor "***Dr. Shrawan Ram, Assistant Professor"***, for their guidance and constant supervision as well as for providing necessary information regarding this Project.

This page was intentionally left blank.

ABSTRACT

This Blog Web-App project is implemented in the Django framework. The main aim of this project is to create an online blogging application like Google blogs …etc. Using this application, users can create a blog with the tag name of the blogger. In this Web-App you can find all features that are present in existing blogging software. Users can design modify by adding new templates and add posts. Initially, users need to register with an application like in google blog sites we log with Gmail user id similarly users need to get a unique user id and password. These posts which are posted in blogger sites will be available in search results and users who visit blogs can post a comment. Time permitting, you may want to allow for more sophisticated models of querying, perhaps exploiting some of the SQL features to, e.g., query for the most recent entries by a particular blog author, etc.

Blog Application:-The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach. User friendliness is provided in the application with various controls. The system makes the overall project management much easier and flexible. Readily uploading the latest updates allows users to download the alerts by clicking the URL. There is no risk of data mismanagement at any level while the project development is under process. It provides a high level of security with different levels of authentication.

This page was intentionally left blank.

Table of Contents

List of Figures

Chapter 1: INTRODUCTION

1.1 Overall Description

The main aim of this Blog Web-Application is to provide hassle-free access to the posted blogs, entries, topics, etc. It is also used for posting blogs, editing blogs, deleting the posted blogs, etc. It is also used for viewing and posting other people's blogs/posts. Every organization, whether big or small, has challenges to overcome and manage the information of Ideas, Blogs, Entries, Content, and Views. Every Online Blogging System has different Blog needs therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

1.2 Purpose

The main purpose of the Project on Online Blogging System is to manage the details of Blogs, Ideas, topics, Entries, and Views. It manages all the information about Blogs, Content, Views, and Blogs. The project is totally built at the administrative end and thus only the administrator is guaranteed the access. The purpose of the Blogs, Ideas, Content, and Topic. It tracks all the details about the Topics, Entries, and Views.

1.2.1 Functionalities

Functionalities provided by the Online Blogging System are as follows:

- Provides the post-viewing facility for everyone.
- The Online Blogging System also manages the Content details online for Entry details, Views details, and Blogs.
- It tracks all the information on Content, Entries, etc.
- Manage the information of Ideas.

- Shows the information and description of the Blogs, and Topics.
- To increase the efficiency of managing the Blogs, Topics.
- Manage the information on the Blogs.
- Editing, adding, posting, and updating records.

1.3 Motivation and Scope

The gist of the idea is to digitalize the process of manually driven ideas, topics, and blog sharing among the users, small or big organizations as well as small forums And then bake it into a web application to make it available for common people even to speed up the process reducing further hassle and issues. It may help collect perfect management details. In a very short time, the collection will be obvious, simple, and sensible. It will help a person to know the management of the past year perfectly and vividly. It also helps in all current work relative to the Online Blogging System. It will also reduce the cost of collecting and the management & collection procedure will go on smoothly.

1.3.1 Aim

Our project aims at Business process automation, i.e. we have tried to computerize various processes of the Online Blogging System.

- To utilize resources in an efficient manner by increasing their productivity through automation.
- It satisfies the user requirement.
- Be easy to understand by the user and operator
- Be easy to operate.
- Have a good user interface
- Be expandable
- Delivered on schedule within the budget.

1.3.2 System Requirements

The proposed system has the following requirements

- System needs to store information about new blog entries.
- System needs to help the internal staff to keep informed of ideas and find them as per various queries.
- System needs to maintain a quantity record.
- System needs to update and delete the record.
- System also needs a search area.
- It also needs a security system to prevent data.

1.4 Problem Statement

The old manual ways were suffering from a series of drawbacks. Since the whole of the work was to be maintained with hands, the process of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order. There used to be lots of difficulties in associating any particular transaction with a particular context. There would always be unnecessary consumption of time while entering records and retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records. The reason behind it is that there is a lot of information to be maintained and has to be kept in mind while running the business. For these reasons we have provided features Present system is partially automated, actually existing system is quite laborious as one has to enter the same information at three different places

1.5 Project Requirements

The classification of the blog's application into different components provided a framework for the categorization of features of the blog, from the user's perspective. To understand the requirements of the blog's application, we performed a study of three popular blog publishing applications. For our study, we considered Wordpress, Blogger.com and blog.com. We found that in addition to the key functionality of the blog, the different blog publishing applications provide functionality that is unique to

their application. For arriving at the requirement checklist, we classified the features provided by these blog publishing applications, under the two components

(i) Functional Requirements

(ii) Non Functional Requirements

1.5.1 Functional Requirements

- Secure Registration and Login of different users
- Users can view all the blogs approved by admin
- It should provide all the users which are registered can post blog
- It should provide admin to approve,delete,review and create all blogs
- Any registered user can comment on any blog posted on website

1.5.2 Non Functional Requirements

- **Safety Requirements**

  If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

- **Security**

  Security requirements ensure that the software is protected from unauthorized access to the system and its stored data. Access permissions for the software information may only be changed by the system's data administrator.

- **Reliability**

  The database update process must roll back all related updates when any update fails.

- **Performance**

  The front-page load time must be no more than 2 seconds for users that access the website using an LTE mobile connection.

Chapter 2: Technology Used

2.1 Key Technology: Django Framework

Choosing a web development framework is a big deal. There are quite a few frameworks on the market, each designed to address different project needs. For many companies and independent projects, the Django framework is an easy choice — it's one of the most popular web development tools.

Django is an open-source python web framework used for rapid development, pragmatic, maintainable, clean design, and secure websites. A web application framework is a toolkit of all components needed for application development.

The main goal of the Django framework is to allow developers to focus on components of the application that are new instead of spending time on already developed components. Django is more fully-featured than many other frameworks on the market. It takes care of a lot of hassles involved in web development; enables users to focus on developing components needed for their application.

2.2 Project Structure

A Django Project when initialized contains basic files by default such as manage.py, view.py, etc. A simple project structure is enough to create a single-page application. Here are the major files and their explanations.

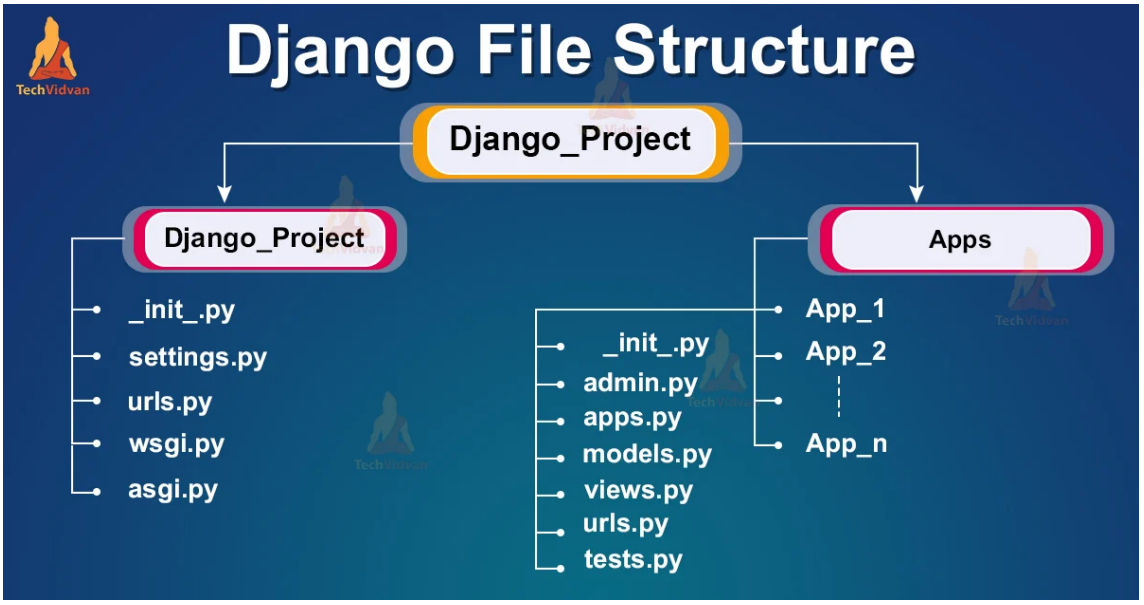

Fig. 2-1: Django File Structure

Inside the Django_site folder (project folder) there will be the following files-

1. **manage.py-** This file is used to interact with your project via the command line(start the server, sync the database… etc). For getting the full list of commands that can be executed by manage.py type this code in the command window-

```
python manage.py help
```

2. **folder (Django_site) –** This folder contains all the packages of your project. Initially, it contains four files –
    1. **_init_.py –** It is a python package. It is invoked when the package or a module in the package is imported. We usually use this to execute package initialization code, for example for the initialization of package-level data.
    2. **settings.py –** As the name indicates it contains all the website settings. In this file, we register any applications we create, the location of our static files, database configuration details, etc.

3. **urls.py –** In this file, we store all links of the project and functions to call.
4. **wsgi.py –** This file is used in deploying the project in WSGI. It is used to help your Django application communicate with the webserver.

2.3 Past Related Work with Django

Learning the concept of Python and Django is a great experience. But only theoretical learning is not enough. It is very important to implement our theoretical knowledge on some real-time projects and for that, we need projects.

There is no better way to learn Django or any other framework than by working on some real-world projects. Projects are essential to make learning easy for us. These projects helped me gain real-world experience and made me job-ready.

Today, I am sharing some interesting Django projects I developed to gain practical knowledge of the framework and helped in developing this project.

2.3.1 Sending Emails with Python

Fig 2-3: Sending Emails with Python

Automate the process of sending emails with some customized features based on business requirements. You can have a list of email addresses and their respective names. Then you can modify the message and send emails to the target audience automatically.

2.3.2 Login System in Django

Fig. 2-4: Login System in Django

Implement a basic template of a login system and then you can use the template in any web app with just minimal changes to quickly build the web apps. Nowadays every

website requires their customers to create accounts therefore this template will be very useful.

2.3.3 Notes App

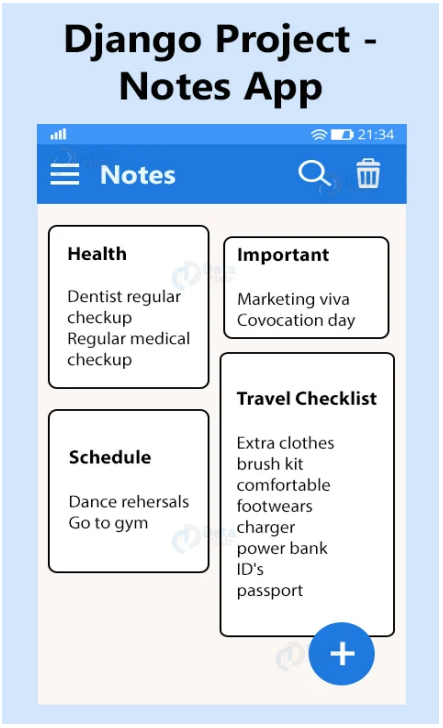

Fig. 2-4: Notes App

In this project, you have to build a nice interface for a notes application. You should provide functionalities for a user to add, edit and delete notes from the application. Also, provide a feature to add images to the notes.

Chapter 3: Project Details

In this Blog Web-App, we have used the Django framework to develop this project. The Django framework is based on MVT architecture.

3.1 Project Design

3.1.1 Level-0 Data Flow Diagram

This is the Zero Level DFD of the system, where we have elaborated the high-level process of Online Blogging. It's a basic overview of the whole Online Blogging system or process being analyzed or modeled. It's designed to be an at-a-glance view of Comment, Technology Blog, and Web Page showing the system as a single high-level process, with its relationship to external entities of Blog and Create Blog. It should be easily understood by a wide audience, including Blog, Create Blog, and Comment

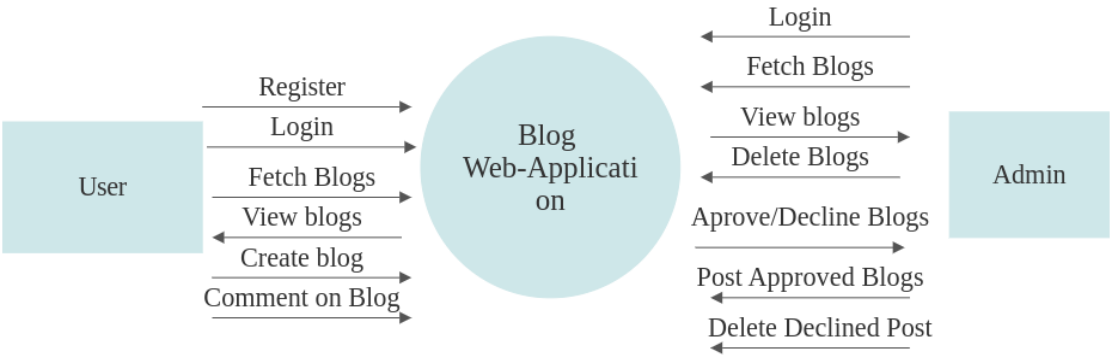

Fig 3-1: Level-0 Data Flow Diagram

3.1.2 Level-1 Data Flow Diagram

Level 1 DFDs are still a general overview, but they go into more detail than a context diagram. In a level 1 data flow diagram, the single process node from the context diagram is broken down into subprocesses. As these processes are added, the diagram will need additional data flows and data stores to link them together.

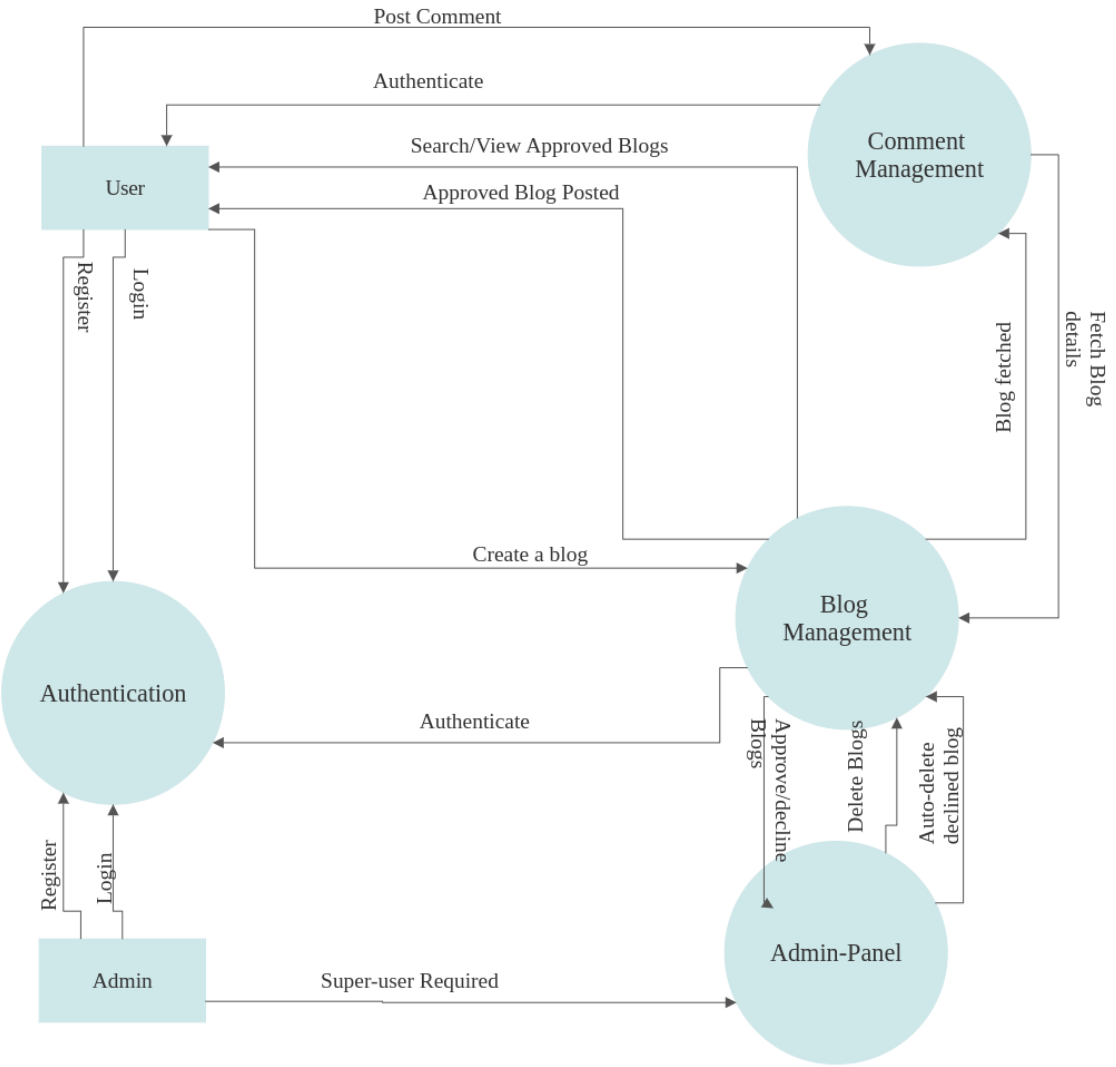

Fig 3-2: Level-1 Data Flow Diagram

3.1.3 Use Case Diagram

This Use Case Diagram is a graphic depiction of the interactions among the elements of a Blogging System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of the Blogging System. The main actors of the Blogging System in this Use Case Diagram are: Admin, User, super user, Anonymous Users, who perform the different type of use cases such as Manage Blog, Manage Blog Category, Manage Create Blog, Manage Blog Type, Manage Comment, Manage Technology Blog, Manage Web Page, Manage Users and Full Blogging System Operations. Major elements of the UML use case diagram of the Blogging System are shown in the picture below.

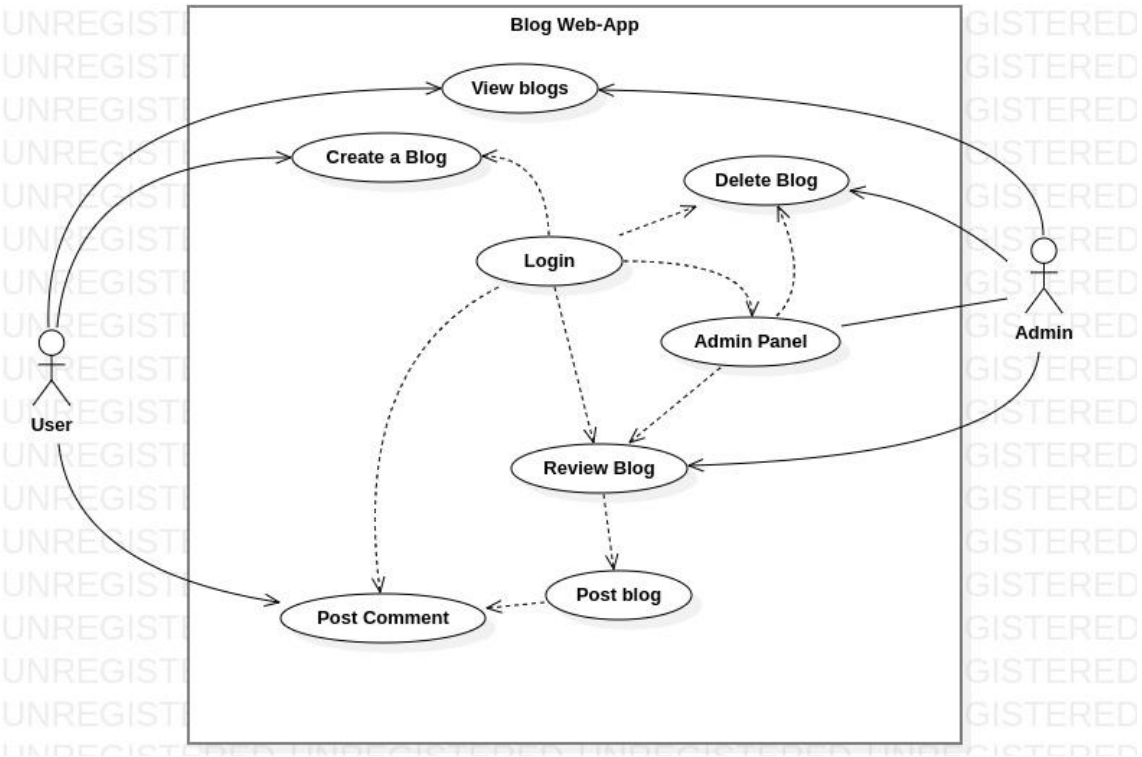

Fig 3-3: Use Case Diagram

3.2 Model-View-Template Architecture

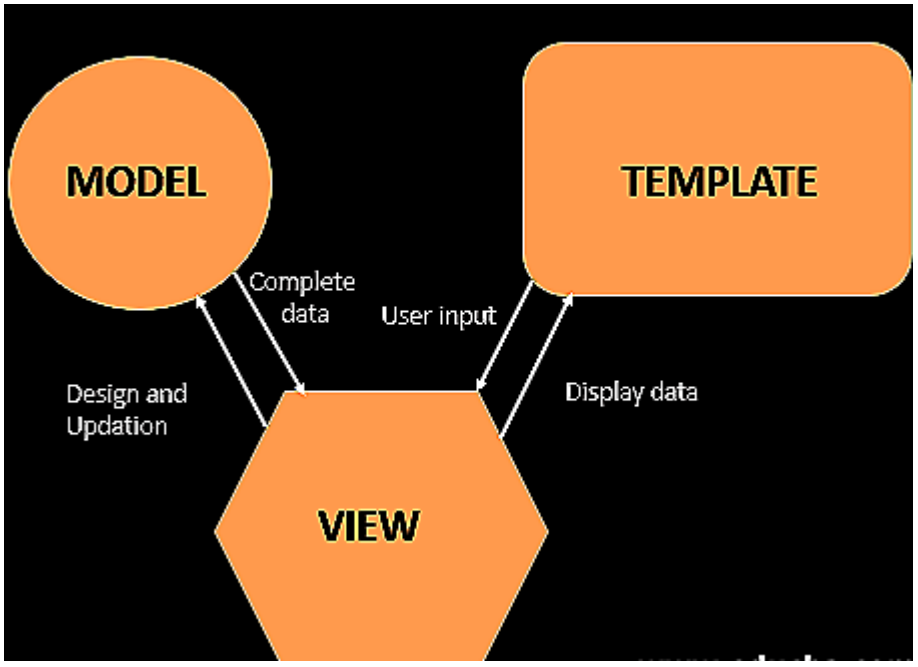

Fig. 3-4: Django Architecture

3.2.1 Models

The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySql, or Postgres).

3.2.2 View

The View is the user interface — what you see in your browser when you render a website. It is represented by HTML/CSS/Javascript and Jinja files.

3.2.3 Template

A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

3.3 Project Components

This project contains various models, views and templates, along with a sqlite3 database.

3.3.1 Models

We have 3 models in this project:

1. **User Model:**

   The User model consists of 4 fields:

   1. username
   2. email - here email is taken as USERNAME_FIELD as well as REQUIRED_FIELD
   3. password
   4. profile_pic

```
class User(AbstractUser):
   username = models.CharField(max_length=100, unique=True,
blank=False, null=False)
   email = models.EmailField(max_length=100, blank=True, unique=True)
   password = models.CharField(max_length=100, blank=True)
   profile_pic = models.ImageField(default="default.jpg",
upload_to="pictures/")
```

```
   USERNAME_FIELD = "email"
   REQUIRED_FIELDS = [email]

   def __str__(self) -> str:
       return self.email
```

2. **Blog Model:**

   The Blog Model consists of 10 fields:

   1. author
   2. title
   3. desc
   4. blog_content
   5. thumbnail
   6. read_more
   7. category
   8. id
   9. created
   10. approved_status

```
class Blog(models.Model):
   author = models.TextField(
       max_length=100, blank=True, null=True, default="unknown user",
editable=False
   )
   title = models.CharField(
       max_length=100,
       unique=False,
   )
   desc = models.TextField(max_length=200, blank=False)
   blog_content = RichTextField(blank=True, null=True)
   thumbnail = models.URLField(max_length=400)
   read_more = models.URLField(max_length=400, blank=True)
   category = models.CharField(max_length=10, blank=True)
   id = models.AutoField(primary_key=True, editable=False)
   created = models.DateTimeField(auto_now_add=True)
   approve_status = models.BooleanField(default=False)

   def __str__(self) -> str:
```

```
        return self.title
```

3. **CommentModel:**

```
class Comment(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    blog = models.ForeignKey(Blog, on_delete=models.CASCADE)
    content = models.TextField()
    created_at = models.DateTimeField(default=datetime.now)

    def __str__(self):
        return self.user.username + " Comment: " + self.content
```

3.3.1 Views

- **Home view :**

  This view contains all the blog approved by admin

- **Register view**

  In this view, new users register themself

- **login view**

  In this view users login

  User Login

  - **my blogs :** In this view, users can view their own blogs
  - **post blog :** User posts new blogs
  - **blog details :** This view shows complete blog content of a particular blog
  - **post Comments :** This feature allows users to comment a particular blog

  Admin panel

  - **approve blog :** In this view, admin can approve new blogs.
  - **delete blog :** In this view, Admin can delete any blog
  - **Django Admin :** This view gives admin access to the django in-built admin page.

3.3.2 Templates

- addblog.html
- blogdetail.html
- edit_profile.html
- login.html
- register.html
- userblog.html
- admin.html
- deletepage.html
- home.html
- nav.html
- review.html

3.2.3 Database

Now we will define the data models for our blog. A model is a Python class that subclasses django.db.models.Model, in which each attribute represents a database field. Using this subclass functionality, we automatically have access to everything within django.db.models.Models and can add additional fields and methods as desired. We will have a Post model in our database to store posts.

Chapter 4: Results

How Blogs are posted:

4.1 Home Page

First Page of the Django Blog Application project with all the posted blogs.

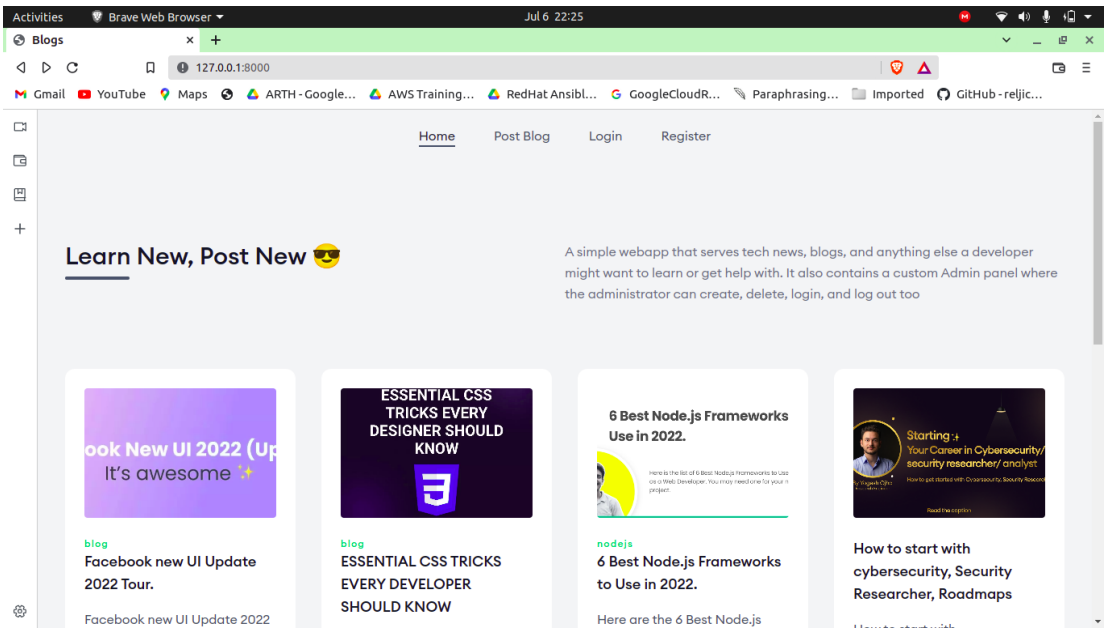

Fig 4-1 Home Page

4.2 User registration page

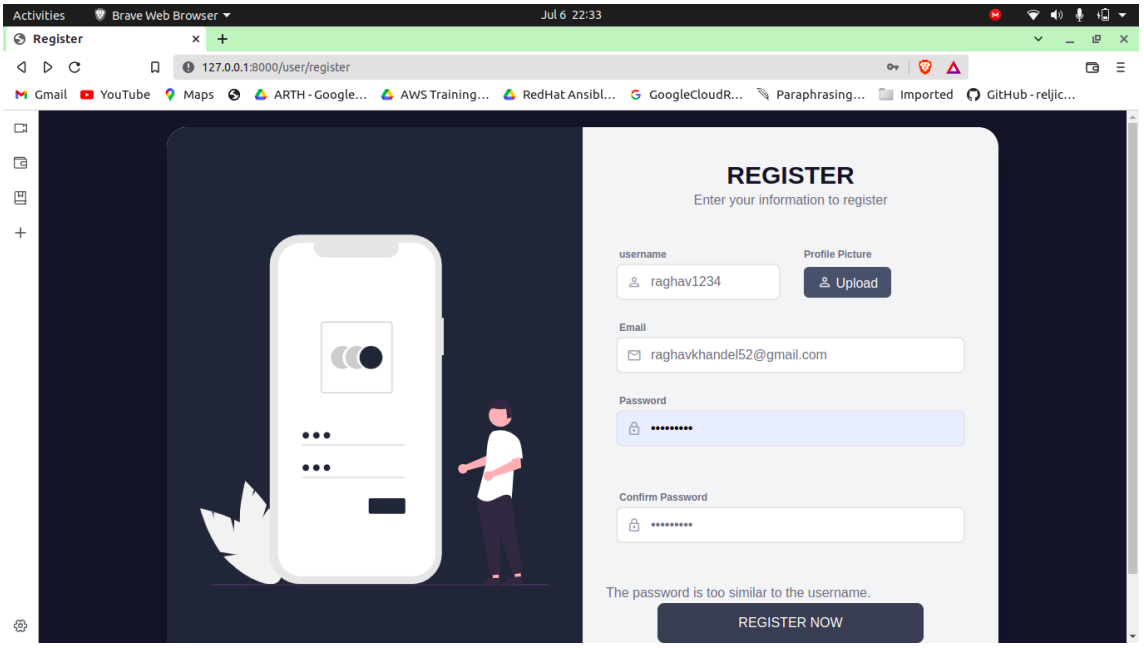

Fig 4-2: Registration Page

4.3 Login Page

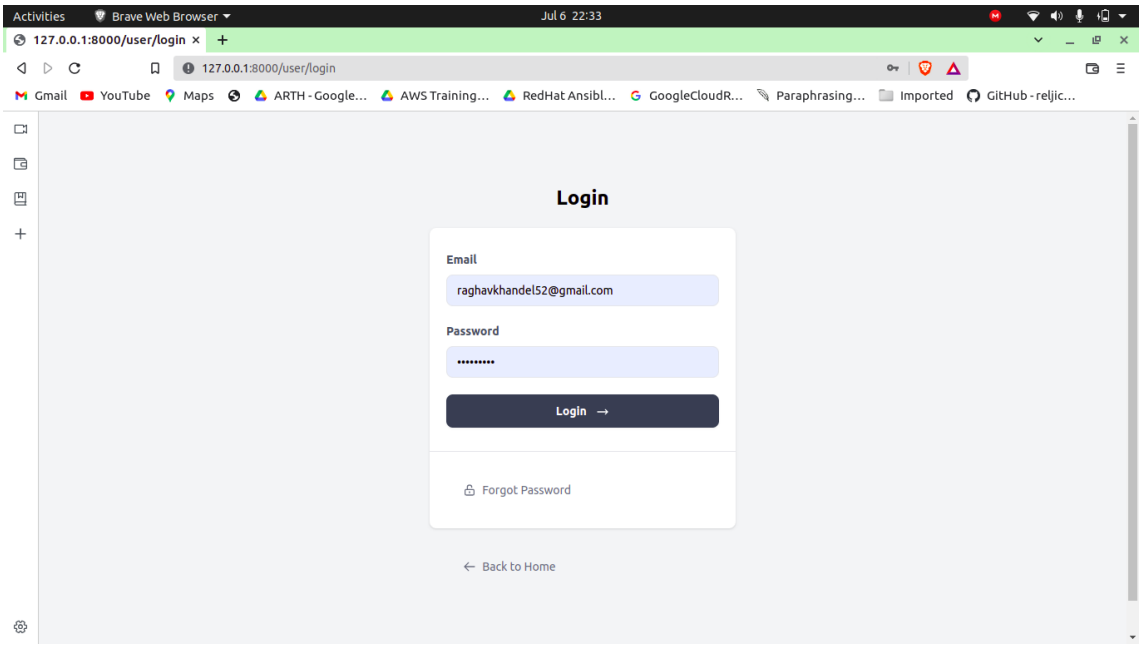

Fig 4-3: Login Page

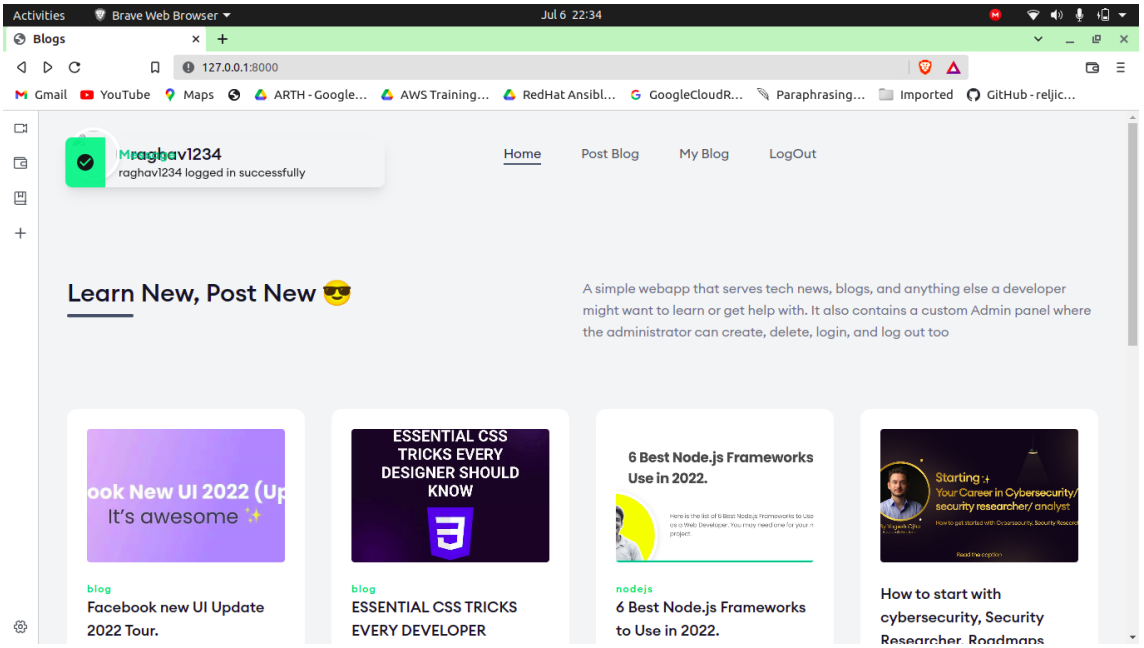

Fig 4-4: User Logged In

4.4 Post Blog

Logged in users can add a new blog

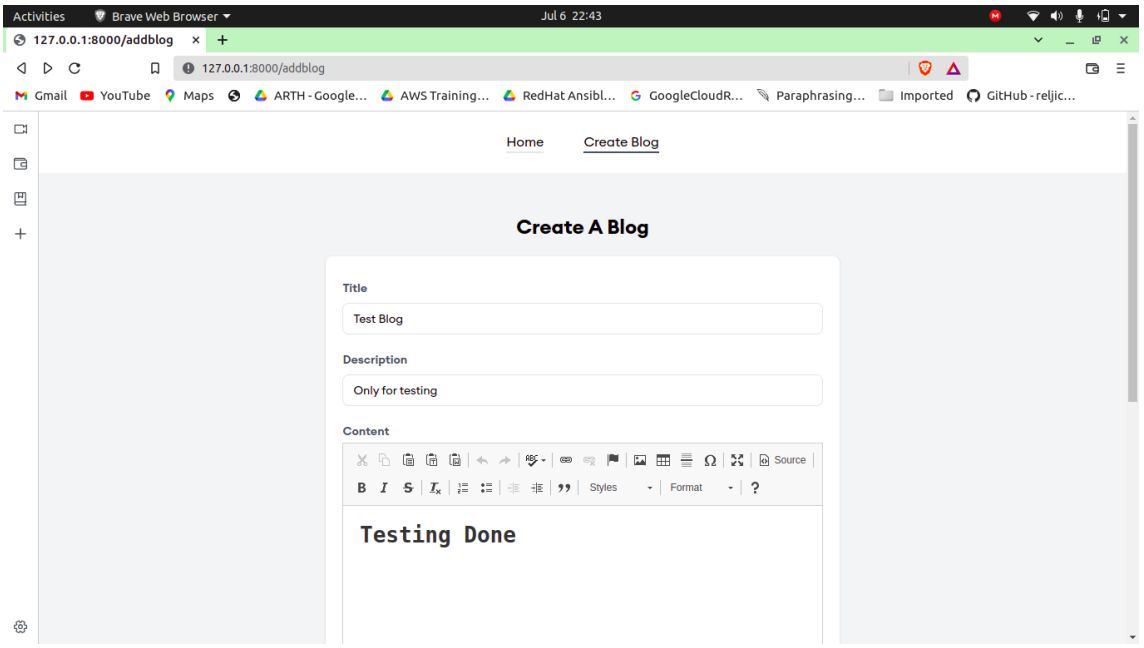

Fig 4-5: Create a new Blog

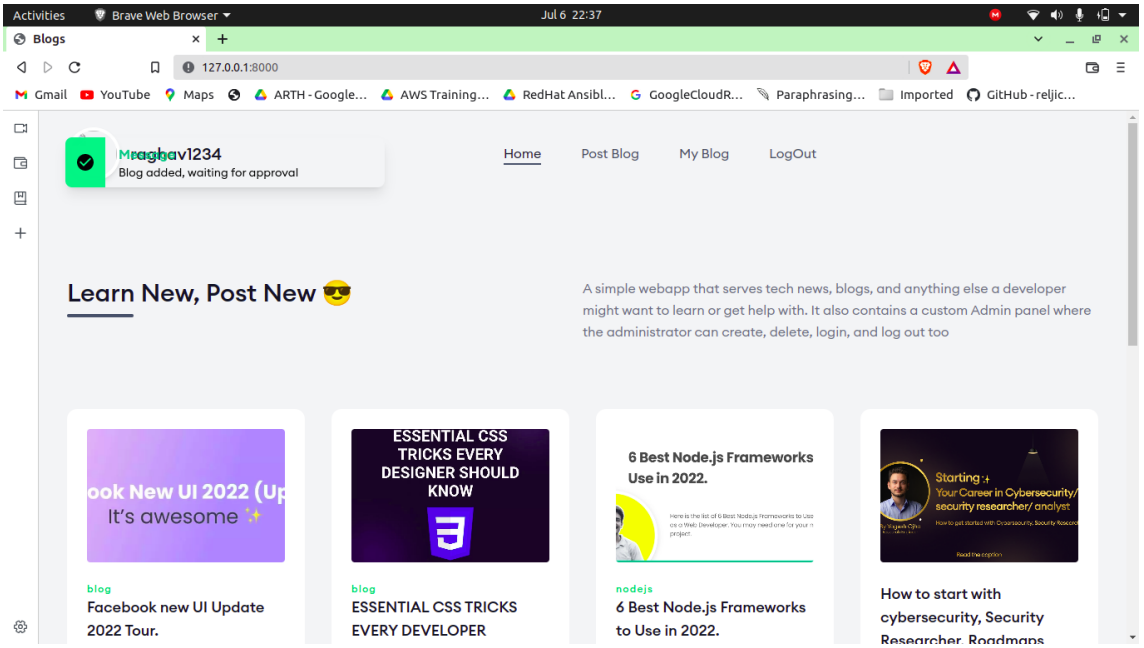

Fig 4-6: Blog added, waiting for approval

4.5 Comment

Users can also comment on any blog posts

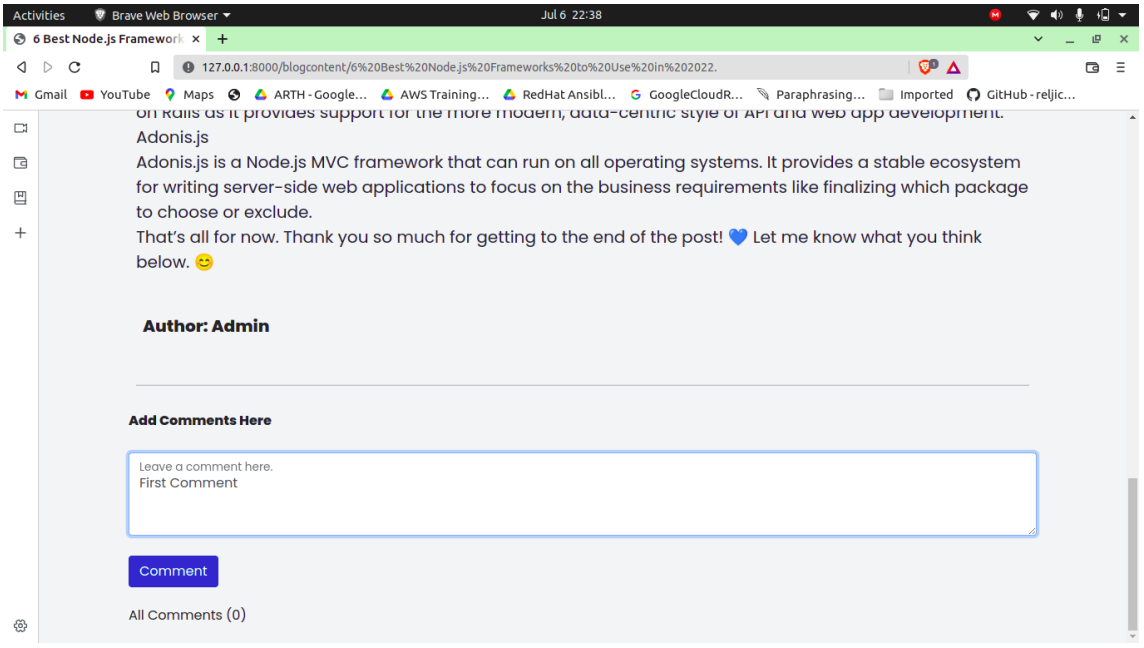

Fig. 4-7: Posting a Comment

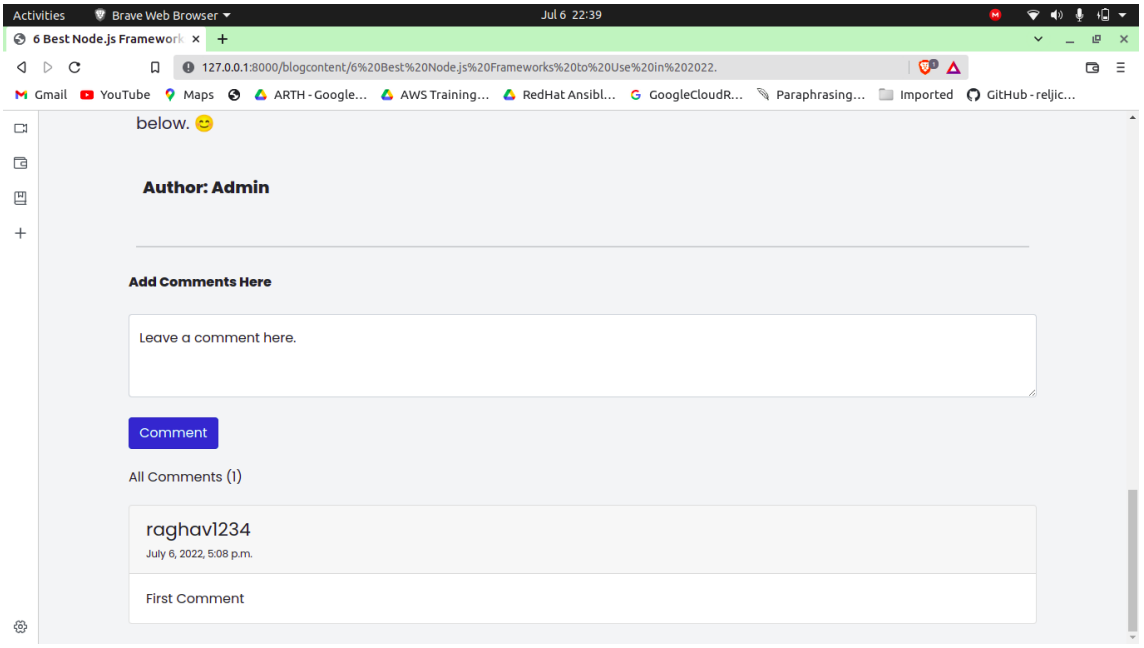

Fig. 4-8: Comment Posted

Chapter 5: Conclusion & Future Work

5.1 Conclusion

Since the advent of blogs, the number of online journals has significantly increased worldwide. Private bloggers write down their own thoughts and share their ideas and experiences with their readers.

Amateur and professional copywriters develop blogs on particular topics and make a name for themselves. Companies choose to develop their own company blogs to increase their brand presence, position themselves as experts and retain customer loyalty.

Successful blogs are authentic, unique, and focus on the target audience.

While developing a system a conscious effort has been made to create and develop a software package, making use of available tools, techniques, and resources – that will generate the proper system cases. While making the system an eye has been kept on making it as user-friendly as such one may hope that the system will be acceptable to any user and will adequately meet his/her needs. As in the case of any system development process where there are a number of shortcomings, there have been some shortcomings in the development of this system also.

With this project in Django, we have successfully developed a Blog Application with proper front-end and back-end. Now, we know many concepts in Django.

5.2 Future Enhancements

- Like a blog
- author followers
- Notification for following authors New Blog

References

[1] *Data-flair.training*. [Online]. Available: https://data-flair.training/. [Accessed: 01-Jul-2022].

[2] E. Ashbee, “The Lott resignation, ‘blogging’ and American conservatism,” *Polit. Q.*, vol. 74, no. 3, pp. 361–370, 2003.

[3] S. Albrecht, M. Lübcke, and R. Hartig-Perschke, “Weblog campaigning in the German Bundestag election 2005,” *Soc. Sci. Comput. Rev.*, vol. 25, no. 4, pp. 504–520, 2007.

[4] “Django tutorial,” *W3schools.com*. [Online]. Available: https://www.w3schools.com/django/. [Accessed: 01-Jul-2022].

[5] M. Otto and J. Thornton, “Get started with bootstrap,” *Getbootstrap.com*. [Online]. Available: https://getbootstrap.com/docs/5.2/getting-started/introduction/. [Accessed: 01-Jul-2022].

[6] “Getting Started,” in *Having Success with NSF*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2012, pp. 1–16.

[7] “Building A blog application with Django,” *Djangocentral.com*. [Online]. Available: https://djangocentral.com/building-a-blog-application-with-django/. [Accessed: 01-Jul-2022].

[8] Techvidvan.com. [Online]. Available: https://techvidvan.com/tutorials/wp-content/uploads/sites/2/2021/06/Django-file-structure.jpg. [Accessed: 09-Jul-2022].

[9] Geeksforgeeks.org. [Online]. Available: https://media.geeksforgeeks.org/wp-content/uploads/20210606092225/image.png. [Accessed: 09-Jul-2022].

[10] "Functional vs non functional requirements," *GeeksforGeeks*, 28-Apr-2020. [Online]. Available: https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/. [Accessed: 10-Jul-2022].

[11] "What is a Data Flow Diagram," *Lucidchart*. [Online]. Available: https://www.lucidchart.com/pages/data-flow-diagram. [Accessed: 10-Jul-2022].

[12] "IBM docs," *Ibm.com*, 02-Mar-2021. [Online]. Available: https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=diagrams-use-cases. [Accessed: 10-Jul-2022].