VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018

A

Fullstack Development Mini Project

Report On

"ಚರ್ಚೆ: An Online Discussion Forum"

SUBMITTED IN PARTIAL FULFILLMENT FOR TH SEMESTER

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

SUBMITTED BY

Pradhyumna G (1JS21IS069)
Sharanagouda G Patil (1JS21IS098)
Sumukha S Hebbar (1JS21IS112)
Shravan S Thane (1JS22IS411)

Under the Guidance of:

Mrs. Punitha M
Assistant Professor,
Dept. of ISE, JSSATEB

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

JSS ACADEMY OF TECHNICAL EDUCATION

JSS Campus, Dr. Vishnuvardhan Road, Bengaluru-560060

2023 - 2024

JSS MAHAVIDYAPEETHA, MYSURU

JSS ACADEMY OF TECHNICAL EDUCATION

JSS Campus, Dr. Vishnuvardhan Road, Bengaluru-560060

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Mini project work entitled "**ಚರ್ಚೆ: An Online Discussion Forum**" carried out by Mr. **Pradhyumna G (1JS21IS069), Sharanagouda G Patil (1JS21IS098), Sumukha S Hebbar (1JS21IS112), Shravan S Thane (1JS22IS411)** is a bonafide student of **JSS ACADEMY OF TECHNICAL EDUCATION BENGALURU** in partial fulfilment for 6th Semester Fullstack Development with mini project in **INFORMATION SCIENCE AND ENGINEERING** of the **Visvesvaraya Technological University**, **Belagavi** during the academic year **2023-24.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

Mrs. Punitha M **Assistant Professor** **Dept. of ISE, JSSATEB**	**Dr. REKHA P. M.** **Professor & Head** **Dept. of ISE, JSSATEB**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. So, with gratitude, we acknowledge all those whose guidance and encouragement crowned my effort with success.

First and foremost, we would like to thank his **Holiness Jagadguru Sri Shivarathri Deshikendra Mahaswamiji** and **Dr. Bhimasen Soragaon**, Principal, JSSATE, Bangalore for providing an opportunity to carry out the FULLSTACK DEVELOPMENT MINI PROJECT (21CS62) as a part of our curriculum in the partial fulfillment of the degree course.

We express our sincere gratitude for our beloved Head of the department, **Dr. Rekha P M**, for her co-operation and encouragement at all the moments of our approach.

It is our pleasant duty to place on record our deepest sense of gratitude to our respected guide **Mrs. Punitha M, Assistant Professor** for the constant encouragement, valuable help and assistance in every possible way.

We would like to thank all **ISE Department teachers** and **non-teaching staff** for providing us with their valuable guidance and for being there at all stages of our work.

Pradhyumna G (1JS21IS069)
Sharanagouda G Patil (1JS21IS098)
Sumukha S Hebbar (1JS21IS112)
Shravan S Thane (1JS22IS411)

ABSTRACT

In today's digital age, online discussion forums have become vital platforms for sharing knowledge, fostering community engagement, and facilitating discussions on various topics. "ಚರ್ಚೆ" is an innovative online discussion forum built using Django, a high-level Python web framework, designed to provide users with a seamless and interactive communication experience.

The "ಚರ್ಚೆ" project aims to create a user-friendly and scalable platform where individuals can initiate and participate in discussions, share insights, and seek solutions collaboratively. Key features of "ಚರ್ಚೆ" include user registration and authentication, real-time notifications, robust moderation tools, and an intuitive user interface. The forum ensures secure data handling and privacy, catering to the diverse needs of its users.

This project report delves into the comprehensive development process of "ಚರ್ಚೆ," covering the initial design and methodology, literature survey, detailed design and implementation, and the results and discussions derived from the project. The introduction chapter outlines the significance of online forums and the specific goals of "ಚರ್ಚೆ." The literature survey chapter reviews existing forums, highlighting their limitations and the unique value proposition of "ಚರ್ಚೆ."

The design and methodology chapter provides an in-depth analysis of the problem, selection of appropriate tools and technologies, and the architectural design of the platform. The development chapter elaborates on the implementation details, including database schemas, backend infrastructure, frontend interfaces, and integration of real-time communication features. The results and discussion chapter presents the outcomes, user feedback, performance metrics, and challenges encountered during the development process.

In conclusion, "ಚರ್ಚೆ" successfully demonstrates the potential of an advanced online discussion forum, addressing the gaps in existing platforms and offering a robust solution for digital communication and collaboration. Future enhancements and recommendations for further research are also discussed, paving the way for continued improvement and innovation in the field of online forums.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 General

In the contemporary digital landscape, online discussion forums have emerged as vital platforms for fostering communication, collaboration, and knowledge sharing among diverse groups of people. These forums provide users with a virtual space to engage in discussions on a wide range of topics, ask questions, share insights, and connect with like-minded individuals. "ಚರ್ಚೆ," which translates to "discussion" in Kannada, is an innovative online discussion forum designed to enhance user interaction and provide a seamless communication experience.

Developed using Django, a high-level Python web framework, "ಚರ್ಚೆ" aims to create a user-friendly, scalable, and secure platform that meets the needs of modern internet users. The forum is equipped with features such as user registration and authentication, real-time notifications, robust moderation tools, and an intuitive user interface, all designed to facilitate smooth and effective communication.

1.2 Purpose and Significance

The primary purpose of "ಚರ್ಚೆ" is to bridge the gap between existing discussion forums and the evolving needs of users. While many current forums offer basic functionalities, they often fall short in areas such as user engagement, ease of use, and security. "ಚರ್ಚೆ" addresses these shortcomings by incorporating advanced features and leveraging the power of Django to deliver a superior user experience.

1.3 Objectives

The main objectives of the "ಚರ್ಚೆ" project are:

- To provide a platform where users can easily create and participate in discussions.
- To ensure the platform is scalable to accommodate a growing number of users and discussions.
- To implement robust security measures to protect user data and maintain privacy.
- To offer real-time communication features to enhance user interaction and engagement.
- To provide tools for effective moderation and management of discussions.

1.4 Scope

The scope of the "ಚರ್ಚೆ" project includes the development of the forum from the ground up, covering all aspects from initial design and methodology to implementation and testing. This report will detail each phase of the development process, including a literature survey, design and methodology, development and implementation, and results and discussions. It will also highlight the challenges encountered during the project and provide recommendations for future enhancements.

1.5 Structure of the Report

This report is structured as follows:

Chapter 2: Literature Survey - Reviews existing online discussion forums, their features, and limitations.
Chapter 3: Design and Methodology - Details the design principles, tools, and technologies used in the project.
Chapter 4: Development and Implementation - Describes the implementation process, including coding, testing, and deployment.
Chapter 5: Results and Discussion - Presents the outcomes of the project, including user feedback and performance metrics.
Chapter 6: Conclusion - Summarizes the project's achievements, limitations, and potential future improvements.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Survey

The development of online discussion forums has a rich history, with numerous platforms evolving to meet the communication needs of users across various domains. Early forums, such as bulletin board systems (BBS) and Usenet, laid the groundwork for modern discussion platforms by providing basic functionalities for user interaction and message exchange. Over time, forums have become more sophisticated, incorporating features like user profiles, moderation tools, and multimedia support.

Popular contemporary forums, such as Reddit, Stack Overflow, and Discourse, offer advanced functionalities and have set high standards for user engagement and interaction. These platforms provide insights into the essential features and best practices that enhance user experience. However, they also highlight certain limitations and areas for improvement, particularly in terms of scalability, ease of use, and security.

2.2 Objectives

The primary objectives of the "ಚರ್ಚೆ" project, in the context of existing literature, are:

- To analyze the features and limitations of current online discussion forums.
- To identify key areas where "ಚರ್ಚೆ" can provide improvements and innovations.
- To establish a foundation for developing a user-centric, scalable, and secure discussion forum.
- 2.3 Problem Definition

Based on the literature survey, several common issues have been identified in existing online discussion forums:

- User Engagement: Many forums struggle to maintain active user participation and engagement over time. Features that encourage continuous interaction and user retention are often lacking.
- Scalability: As user bases grow, forums must handle increased traffic and data volume. Ensuring that the platform remains responsive and efficient under heavy loads is a significant challenge.
- Usability: Complex user interfaces and poor user experience design can deter new users from participating in discussions. Simplifying navigation and interaction is crucial for user adoption.
- Security: Protecting user data and ensuring privacy are paramount in any online platform. Many forums face challenges in implementing robust security measures to safeguard against data breaches

2.4 Identified Gaps

From the problem definition, several gaps have been identified in existing forums that "ಚರ್ಚೆ" aims to address:

- Enhanced User Engagement: By incorporating real-time notifications, gamification elements, and personalized content recommendations, "ಚರ್ಚೆ" seeks to foster higher levels of user interaction and engagement.
- Scalable Infrastructure: Leveraging Django's capabilities, "ಚರ್ಚೆ" will be designed to handle large volumes of users and data efficiently, ensuring consistent performance and responsiveness.
- Intuitive User Interface: "ಚರ್ಚೆ" will prioritize user-friendly design, making it easy for users of all technical backgrounds to navigate and participate in discussions.
- Robust Security: Implementing advanced security measures, such as encryption, secure authentication, and regular security audits, will ensure that user data is protected and privacy is maintained.

2.5 Summary

The literature survey has provided valuable insights into the evolution and current state of online discussion forums. By analyzing existing platforms, identifying common issues, and highlighting areas for improvement, a clear path has been established for the development of "ಚರ್ಚೆ." This project aims to build on the strengths of existing forums while addressing their limitations, ultimately delivering a superior online discussion experience for users.

CHAPTER -3

DESIGN AND METHODOLGY

3.1 Introduction

The design and methodology chapter outlines the approach taken to develop "ಚರ್ಚೆ," an online discussion forum. This chapter covers the problem analysis, selection of appropriate technologies, architectural design, and the step-by-step methodology followed during the project development.

3.2 Problem Analysis

The problem analysis involves identifying the key challenges faced by existing discussion forums, as highlighted in the literature survey. These challenges include:

- User Engagement: Difficulty in maintaining active participation and interaction among users.
- Scalability: Handling increased traffic and data volume as the user base grows.
- Usability: Simplifying user interface and navigation to encourage user adoption.
- Security: Implementing robust security measures to protect user data and ensure privacy.
- By addressing these challenges, "ಚರ್ಚೆ" aims to provide a comprehensive solution that enhances user experience, ensures scalability, and maintains high security standards.

3.3 System Architecture

The architecture of "ಚರ್ಚೆ" is designed to be modular, scalable, and secure. The key components of the system include:

- Frontend: Built using HTML, CSS, and JavaScript, the frontend provides an intuitive and user-friendly interface for users to interact with the forum.
- Backend: Developed using Django, the backend handles user authentication, data management, and server-side logic.
- Database: A relational database is used to store user information, discussion threads, and other relevant data.

3.4 Design Principles

The design of "ಚರ್ಚೆ" follows several key principles:

- User-Centric Design: Prioritizing user experience by making the interface intuitive and easy to navigate.
- Modularity: Developing the system in a modular fashion to facilitate easy maintenance and future enhancements.
- Scalability: Ensuring the system can handle a growing number of users and increased data volume efficiently.
- Security: Implementing robust security measures to safeguard user data and ensure privacy.

3.5 Methodology

The development methodology for "ಚರ್ಚೆ" follows an Agile approach, which involves iterative and incremental development. The key stages of the methodology are:

- Requirements Gathering: Collecting and documenting the requirements for the forum, including features, functionalities, and user expectations.
- Planning: Creating a detailed project plan, including timelines, milestones, and resource allocation.
- Design: Developing the system architecture and design specifications, including database schemas, user interfaces, and API designs.
- Development: Implementing the frontend, backend, and database components, followed by integration and testing.
- Testing: Conducting comprehensive testing to identify and fix bugs, ensure functionality, and validate performance.
- Deployment: Deploying the forum to a production environment, ensuring all components are properly configured and operational.
- Maintenance: Providing ongoing support, addressing issues, and implementing enhancements based on user feedback.

3.6 Tools and Technologies

The following tools and technologies are used in the development of "ಚರ್ಚೆ":

Visual Studio Code

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linex and macOS Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.
Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++. Visual Studio Code also ships with IntelliSense for TypeScript JSON, CSS, HTML and PHP.

MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. It is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −It is released under an open-source license. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. and works very quickly and works well even with large data sets.
By encapsulating UI elements into self-contained components, React.js promotes code reusability, modularity, and scalability. Additionally, React.js leverages a virtual DOM to optimize rendering performance, minimizing unnecessary DOM manipulations and ensuring fast updates to the UI. With its unidirectional data flow and robust ecosystem of libraries and tools, React.js has become a go-to-choice for building modern web applications that deliver seamless user experiences across various devices and platforms.

HTML

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and Java Scripts, it forms a triad of cornerstone technologies for the World Wide Web. Web browser receives HTML documents from a web server or from local storage and renders the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate CSS file, and reduce complexity and repetition in the structural content. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a element. This cascading priority scheme is predictable.

DJANGO

Django is an ideal choice for developing "ಚರ್ಚೆ - An Online Discussion Forum" due to its robust feature set and ease of use. Here's why:

- Rapid Development: Django's simplicity and built-in features allow for quick development cycles, helping in getting the forum up and running efficiently.
- Scalability: The framework's design can handle the growth of user base and data, ensuring the forum remains performant as it scales.
- Security: With security features built into the framework, user data and interactions within "ಚರ್ಚೆ" will be well-protected.
- Admin Interface: The auto-generated admin interface will facilitate easy management of user accounts, discussions, and moderation tasks.
- Community Support: Django's active community provides a wealth of resources, plugins, and support, which can be invaluable during development and maintenance

BOOTSTRAP

Bootstrap is an HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web

PYTHON

Python, known for its simplicity and readability, is the backbone of Django, a high-level web framework. Python's extensive standard library, dynamic typing, and strong community support enhance Django's capabilities, enabling rapid development, scalability, and robust security for web applications. This synergy makes Python ideal for efficient Django-based projects

3.7 Summary

The design and methodology chapter provides a detailed overview of the approach taken to develop "ಚರ್ಚೆ." By following a user-centric design, modular architecture, and Agile development methodology, the project aims to create a scalable, secure, and user-friendly online discussion forum. The next chapter will delve into the development and implementation details, showcasing how the design and methodology were translated into a functional system

CHAPTER-4

DEVELOPMENT AND IMPLEMENTATION

4.1 Design and Implementation

The development phase of "ಚಚೆರ್ - An Online Discussion Forum" involves translating the design specifications into a functional application. This chapter details the architecture, database schema, and the integration of Django components. The implementation covers frontend and backend development, ensuring seamless user interaction and robust system performance.

4.2 Prototype Flowchart

A prototype flowchart illustrates the user journey within "ಚಚೆರ್," mapping out key interactions such as user registration, discussion creation, posting replies, and moderation tasks. This visual representation aids in refining user experience and ensuring intuitive navigation throughout the forum.

4.3 Components Description

Detailed descriptions of frontend interfaces, backend services, and database interactions highlight each component's role in "ಚಚೆರ್." This includes user authentication, real-time notifications, data storage, and administrative functionalities, ensuring comprehensive coverage of the forum's features and functionalities.

MODELS

Django models are Python classes that define the structure of data in a Django application. They represent database tables, encapsulate data attributes, and establish relationships between different types of data. Django's ORM facilitates seamless interaction with databases, enabling efficient data management and querying through Python code

VIEWS

In Django, views are Python functions or classes that handle HTTP requests and return responses. They encapsulate business logic, fetching data from models, processing requests, and rendering templates. Views bridge URLs to application logic, enabling dynamic content generation and interaction with users based on their actions.

TEMPLATES

In Django, templates are HTML files containing placeholders and template tags that Django's templating engine processes to generate dynamic web pages. Templates allow for the presentation layer of Django applications, rendering data from views and integrating with CSS and JavaScript to create interactive user interfaces.

4.4 Software Interface

The software interface integrates Django's ORM capabilities with frontend templates, enabling dynamic content generation and seamless data retrieval. This interface ensures efficient communication between users and the database, supporting real-time updates and personalized user experiences.

4.5 Coding with Django

Development involves coding backend logic using Python and Django's built-in libraries. This includes implementing models, views, forms, and templates to handle user interactions, manage data, and enforce business rules. Code snippets and examples illustrate the application's core functionalities and demonstrate Django's flexibility in customizing forum behaviours.

CODE SNIPPETS

FSD-PROJECT>Forum>profile_of_user>models.py

```
from django.db import models
from django.contrib.auth.models import User
from django.utils.timezone import now

# Create your models here.
class Profile(models.Model):
    user = models.OneToOneField(User, null=True, blank=True,
on_delete=models.CASCADE)
    image = models.ImageField(upload_to="images",default="default/user.png")


class Post(models.Model):
    user1 = models.ForeignKey(User, on_delete=models.CASCADE, default=1)
    post_id = models.AutoField
    post_content = models.CharField(max_length=5000)
    timestamp= models.DateTimeField(default=now)
    image = models.ImageField(upload_to="images",default="")

class Replie(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE, default=1)
    reply_id = models.AutoField
```

```
    reply_content = models.CharField(max_length=50000)
    post = models.ForeignKey(Post, on_delete=models.CASCADE, default='')
    timestamp= models.DateTimeField(default=now)
    image = models.ImageField(upload_to="images",default="")
```

FSD-PROJECT>Forum>profile_of_user>views.py

```
from django.shortcuts import render, redirect, HttpResponse, Http404
from django.contrib import messages
from django.contrib.auth.models import User
from django.contrib.auth  import authenticate,  login, logout
from .models import Post, Replie, Profile
from .forms import ProfileForm
from django.contrib.auth.decorators import login_required

def forum(request):
    profile = Profile.objects.all()
    if request.method=="POST":
        user = request.user
        image = request.user.profile.image
        content = request.POST.get('content','')
        post = Post(user1=user, post_content=content, image=image)
        post.save()
        alert = True
        return render(request, "forum.html", {'alert':alert})
    posts = Post.objects.filter().order_by('-timestamp')
    return render(request, "forum.html", {'posts':posts})

def discussion(request, myid):
    post = Post.objects.filter(id=myid).first()
    replies = Replie.objects.filter(post=post)
    if request.method=="POST":
        user = request.user
        image = request.user.profile.image
        desc = request.POST.get('desc','')
        post_id =request.POST.get('post_id','')
        reply = Replie(user = user, reply_content = desc, post=post, image=image)
        reply.save()
        alert = True
        return render(request, "discussion.html", {'alert':alert})
    return render(request, "discussion.html", {'post':post, 'replies':replies})

def UserRegister(request):
    if request.method=="POST":
        username = request.POST['username']
        email = request.POST['email']
        first_name=request.POST['first_name']
```

```
        last_name=request.POST['last_name']
        password = request.POST['password']
        confirm_password = request.POST['confirm_password']

        if len(username) > 15:
            messages.error(request, "Username must be under 15 characters.")
            return redirect('/register')
        if not username.isalnum():
            messages.error(request, "Username must contain only letters and numbers.")
            return redirect('/register')
        if password != confirm_password:
            messages.error(request, "Passwords do not match.")
            return redirect('/register')

        user = User.objects.create_user(username, email, password)
        user.first_name = first_name
        user.last_name = last_name
        user.save()
        return render(request, 'login.html')
    return render(request, "register.html")

def UserLogin(request):
    if request.method=="POST":
        username = request.POST['username']
        password = request.POST['password']

        user = authenticate(username=username, password=password)

        if user is not None:
            login(request, user)
            messages.success(request, "Successfully Logged In")
            return redirect("/myprofile")
        else:
            messages.error(request, "Invalid Credentials")
        alert = True
        return render(request, 'login.html', {'alert':alert})
    return render(request, "login.html")

def UserLogout(request):
    logout(request)
    messages.success(request, "Successfully logged out")
    return redirect('/login')

@login_required(login_url = '/login')
def myprofile(request):
    if request.method=="POST":
        user = request.user
```

```
      profile = Profile(user=user)
        profile.save()
        form = ProfileForm(data=request.POST, files=request.FILES)
        if form.is_valid():
            form.save()
            obj = form.instance
            return render(request, "profile.html",{'obj':obj})
    else:
        form=ProfileForm()
    return render(request, "profile.html", {'form':form})
```

FSD-PROJECT>Forum>profile_of_user>forms.py

```
from django import forms
from .models import Profile

class ProfileForm(forms.ModelForm):
    class Meta:
        model = Profile
        fields = ('image', )
```

FSD-PROJECT>Forum>profile_of_user>admin.py

```
from django.contrib import admin

# Register your models here.
from .models import Post, Replie, Profile

admin.site.register(Post)
admin.site.register(Replie)
admin.site.register(Profile)
```

FSD-PROJECT>Forum>profile_of_user>apps.py

```
from django.apps import AppConfig


class ProfileOfUserConfig(AppConfig):
    name = 'profile_of_user'
```

FSD-PROJECT>Forum>profile_of_user>urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path("", views.forum, name="Forum"),
```

```
    path("discussion/<int:myid>/", views.discussion, name="Discussions"),
    path("register/", views.UserRegister, name="Register"),
    path("login/", views.UserLogin, name="Login"),
    path("logout/", views.UserLogout, name="Logout"),
    path("myprofile/", views.myprofile, name="Myprofile"),
]
```

FSD-PROJECT>Forum>Forum>urls.py

```
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('profile_of_user.urls')),

]  + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

FSD-PROJECT>Forum>profile_of_user>templates>basic.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css"
integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS"
crossorigin="anonymous">

    <title>{% block title %} {% endblock %}</title>
      <style>
          {% block css %} {% endblock %}
    </style>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
  <a class="navbar-brand" href="/">Discussion Forum</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
```

```
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item {% block homeactive %} {% endblock homeactive %} ">
        <a class="nav-link" href="/">Home <span class="sr-only"></span></a>
      </li>
      {% if user.is_authenticated %}
<ul class="navbar-nav mr-2">
<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle" href='#' id="navbarDropdown" role="button" data-
toggle="dropdown"> Welcome {{request.user}}</a>
<div class="dropdown-menu" aria-labelledby="navbarDropdown">
          <a class="dropdown-item" href="/logout">Logout</a>

      <li class="nav-item {% block profileactive %} {% endblock profileactive %}">
        <a class="nav-link" href="/myprofile">Profile</a>
      </li>

        </div>
      </li>
{% else %}
      <li class="nav-item {% block registeractive %} {% endblock registeractive %}">
        <a class="nav-link" href="/register">Register</a>
      </li>
      <li class="nav-item {% block loginactive %} {% endblock loginactive %}">
        <a class="nav-link" href="/login">Login</a>
      </li>
      {% endif %}

    </ul>
  </div>
</nav>
<br>
</body>
{% block body %} {% endblock %}

<script src="https://code.jquery.com/jquery-3.6.0.js" integrity="sha256-
H+K7U5CnXl1h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk=" crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js"
integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut"
crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js"
integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k"
crossorigin="anonymous"></script>
  {% block js %} {% endblock %}
</html>
```

FSD-PROJECT>Forum>profile_of_user>templates>discussion.html

```
{% extends 'basic.html' %}

{% block title %} Solutions {% endblock %}

{% block css %}
{% endblock %}
{% block body %}
<br>
<div class="container-fluid mt-5">
    <div class="row">
        <div class="col-md-12">
            <div class="card mb-4">
                <div class="card-header">
                    <div class="media flex-wrap w-100 align-items-center"> <img
src="/media/{{post.image}}"
                            class="d-block ui-w-40 rounded-circle" alt="" width="40px"
height="40px">
                        <div class="media-body ml-3" style="text-transform:
uppercase;"> <a data-abc="true">{{post.user1}}</a>
                        </div>
                        <div class="text-muted small ml-3">
                            <div>Date and Time Of Post :</div>
                            <div><strong>{{post.timestamp}}</strong></div>
                        </div>
                    </div>
                </div>
                <div class="card-body">
                    <p>{{post.post_content}}</p>
                </div>
                <div class="card-footer d-flex flex-wrap justify-content-between
align-items-center px-0 pt-0 pb-3">

                    <div class="px-4 pt-3"> <button type="button" class="btn btn-
primary" data-target="#reply" data-toggle="modal"><i
                                class="ion ion-md-create"></i>  Reply</button>
</div>
                </div>
            </div>
        </div>
    </div>
</div>
{% for reply in replies %}
<br>
<div class="container">
<div class="media">
```

```
    <img class="mr-3 rounded-circle" src="/media/{{reply.image}}" alt="." width="30px"
height="30px">
    <div class="media-body">
      <h4 class="mt-0">{{reply.user}}
                  <span><small style="font-size:
13px;">({{reply.timestamp}})</small></span></h4>
      <h5>{{reply.reply_content}}</h5>
    </div>
  </div>
</div>
{% endfor %}
<!-- Modal -->
<div class="modal fade" id="reply" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
          <button type="button" class="close" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">&times;</span>
          </button>
        </div>
        {% if user.is_authenticated %}
        <div class="modal-body">
          <form action="/discussion/{{post.id}}/" method="POST"> {% csrf_token %}
            <div class="form-group">
              <label for="exampleFormControlTextarea1">Post Your Solution Here</label>
              <input type="hidden" name="post_id" value="{{post.id}}">
              <textarea class="form-control" id="desc" name="desc"
rows="3"></textarea>
            </div>
          </div>
          {% else %}
          <h3>Please Login first to post a reply</h3>
          {% endif %}
        <div class="modal-footer">
          <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
          <button type="submit" class="btn btn-primary">Post Solution</button>
        </div>
      </div>
    </div>
  </div>
{% endblock %}
{% block js %}
<script>
 {% if alert %}
alert('Your Reply has been posted successfully!!');
```

```
document.location = "/"
{% endif %}
</script>
{% endblock %}
```

FSD-PROJECT>Forum>profile_of_user>templates>forum.html

```
{% extends 'basic.html' %}

{% block title %} Forum {% endblock %}
{% block homeactive %} active {% endblock homeactive %}

{% block css %}
body {
margin: 0;
font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue",
Arial, "Noto Sans", sans-serif,
"Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji";
font-size: .88rem;
font-weight: 400;
line-height: 1.5;
color: #495057;
text-align: left;
background-color: #eef1f3
}

.mt-100 {
margin-top: 100px
}

.card {
box-shadow: 0 0.46875rem 2.1875rem rgba(4, 9, 20, 0.03), 0 0.9375rem 1.40625rem
rgba(4, 9, 20, 0.03), 0 0.25rem
0.53125rem rgba(4, 9, 20, 0.05), 0 0.125rem 0.1875rem rgba(4, 9, 20, 0.03);
border-width: 0;
transition: all .2s
}

.card-header:first-child {
border-radius: calc(.25rem - 1px) calc(.25rem - 1px) 0 0
}
.card-header {
display: flex;
align-items: center;
border-bottom-width: 1px;
padding-top: 0;
```

```
padding-bottom: 0;
padding-right: .625rem;
height: 3.5rem;
text-transform: uppercase;
background-color: #fff;
border-bottom: 1px solid rgba(26, 54, 126, 0.125)
}

.btn-primary {
color: #fff;
background-color: #3f6ad8;
border-color: #3f6ad8
}

.btn {
font-size: 0.8rem;
font-weight: 500;
outline: none !important;
position: relative;
transition: color 0.15s, background-color 0.15s, border-color 0.15s, box-shadow 0.15s
}

.card-body {
flex: 1 1 auto;
padding: 1.25rem
}

.card-body p {
font-size: 13px
}

a {
color: #E91E63;
text-decoration: none !important;
background-color: transparent
}

.media img {
width: 40px;
height: auto
}
{% endblock %}
{% block body %}
{% load static %}
<div class="container my-4">
<div class="jumbotron">
```

```
  <h1 class="display-4" style="font-family: Arial, sans-serif; color: #1E90FF; text-
align: center; margin-top: 20px; font-size: 4em; font-weight: bold; letter-spacing:
1px; text-shadow: 2px 2px 4px #000000; border-bottom: 2px solid #1E90FF; padding-
bottom: 10px; padding-left: 10px; padding-right: 10px; background-color: #f0f8ff;">
    ಚರ್ಚೆ
</h1>

<p class="lead" style="color: #2c3e50; font-family: Arial, sans-serif; font-size:
1.5em; line-height: 1.8; margin-bottom: 25px; text-align: justify; background-color:
#f8f9fa; padding: 15px; border-radius: 5px; box-shadow: 0 4px 8px rgba(0, 0, 0,
0.1);">
  Welcome to our discussion forum. You can post your question or any related queries
by simply clicking on the add post button. <br> Thank you and have a nice day!!
</p>
<hr class="my-4" style="border: 2px solid #3498db; margin-bottom: 30px;">
<p style="color: #16a085; font-family: 'Times New Roman', serif; font-size: 1.2em;
margin-top: 25px; text-align: justify; background-color: #e8f6f3; padding: 15px;
border-radius: 5px; box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);">
  You can also reply to others' posts by clicking on their post and replying to them.
</p>


      <button class="btn btn-primary btn-lg" data-target="#questions" data-
toggle="modal" role="button">Add Post</button>
    </div>
    {% for post in posts %}
    <div class="container-fluid mt-10">
      <div class="row">
          <div class="col-md-12">
              <div class="card mb-4">
                  <div class="card-header">
                      <div class="media flex-wrap w-100 align-items-center"> <img
src="/media/{{post.image}}"
                              class="d-block ui-w-40 rounded-circle" alt="">
                          <div class="media-body ml-3"> <a
href="/discussion/{{post.id}}" data-abc="true">{{post.user1}}</a>
                          </div>
                          <div class="text-muted small ml-3">
                            <div class="px-4 pt-3">{{post.timestamp}} </div>
                          </div>
                      </div>
                  </div>
                  <div class="card-body">
                      <p>{{post.post_content}}</p>
                  </div>
                  <div class="card-footer d-flex flex-wrap justify-content-between
align-items-center px-0 pt-0 pb-3">
```

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

</div>

<!-- Modal -->
<div class="modal fade" id="questions" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
          <button type="button" class="close" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">&times;</span>
          </button>
        </div>
        {% if user.is_authenticated %}
        <div class="modal-body">
          <form action="/" method="POST"> {% csrf_token %}
          <div class="form-group">
            <label for="exampleFormControlTextarea1">Post Your Question Here</label>
            <textarea class="form-control" id="content" name="content"
rows="3"></textarea>
          </div>
        </div>
        {% else %}
        <h3>Please Login to post</h3>
        {% endif %}
        <div class="modal-footer">
          <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
          <button type="submit" class="btn btn-primary">Post</button>
        </div>
      </div>
    </div>
  </div>
    {% endblock %}
    {% block js %}
    <script>
     {% if alert %}
    alert('Your Question has been posted successfully!!');
    document.location = '/';
    {% endif %}
```

```
   </script>
    {% endblock %}
```

FSD-PROJECT>Forum>profile_of_user>templates>login.html

```
{% extends 'basic.html' %}

{% block title %} Log In {% endblock %}
{% block loginactive %} active {% endblock loginactive %}
{% block css %}
@import url(https://fonts.googleapis.com/css?family=Roboto:400,300,500);
*:focus {
  outline: none;
}

body {
  margin: 0;
  padding: 0;
  background: #DDD;
  font-size: 16px;
  color: #222;
  font-family: 'Roboto', sans-serif;
  font-weight: 300;
}

#login-box {
  position: relative;
  margin: 5% auto;
  width: 700px;
  height: 500px;
  background: #FFF;
  border-radius: 2px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.4);
}

.items {
  position: absolute;
  top: 0;
  left: 0;
  box-sizing: border-box;
  padding: 40px;
  width: 700px;
  height: 500px;
}

h1 {
  margin: 0 0 20px 0;
```

```
  font-weight: bold;
  font-size: 28px;
  text-align: center;
}

input[type="text"],
input[type="password"] {
  display: block;
  box-sizing: border-box;
  margin-bottom: 20px;
  padding: 4px;
  width: 600px;
  height: 32px;
  border: none;
  border-bottom: 1px solid #AAA;
  font-family: 'Roboto', sans-serif;
  font-weight: 400;
  font-size: 15px;
  transition: 0.2s ease;
}

input[type="text"]:focus,
input[type="password"]:focus {
  border-bottom: 2px solid #16a085;
  color: #16a085;
  transition: 0.2s ease;
}

input[type="submit"] {
  margin-top: 28px;
  width: 120px;
  height: 32px;
  background: #16a085;
  border: none;
  border-radius: 2px;
  color: #FFF;
  font-family: 'Roboto', sans-serif;
  font-weight: 500;
  text-transform: uppercase;
  transition: 0.1s ease;
  cursor: pointer;
  position: relative;
  left: 250px;
}

input[type="submit"]:hover,
input[type="submit"]:focus {
```

```
  opacity: 0.8;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.4);
  transition: 0.1s ease;
}

input[type="submit"]:active {
  opacity: 1;
  box-shadow: 0 1px 2px rgba(0, 0, 0, 0.4);
  transition: 0.1s ease;
}
{% endblock %}
{% block body %}
<br>
{% for message in messages  %}
<div class="alert alert-{{ message.tags }} alert-dismissible fade show" role="alert">
      <strong>Message : </strong> {{ message }}
      <button type="button" class="close" data-dismiss="alert" aria-label="Close">
        <span aria-hidden="true">&times;</span>
      </button>
    </div>
{% endfor %}
<div id="login-box">
    <div class="items">
      <h1>Log In</h1>
              <form action="/login/" method="post"> {% csrf_token %}
      <input type="text" name="username" placeholder="Username" />
      <input type="password" name="password" placeholder="Password" />
      <input type="submit" name="signup_submit" value="Log In" />
    </div>
  </div>
  {% endblock %}
```

FSD-PROJECT>Forum>profile_of_user>templates>profile.html

```
{% extends 'basic.html' %}

{% block title %} Profile {% endblock %}
{% block profileactive %} active {% endblock profileactive %}

{% block css %}
{% endblock %}
{% block body %}
<br>
{% for message in messages %}
<div class="alert alert-{{ message.tags }} alert-dismissible fade show" role="alert">
    <strong>Message : </strong> {{ message }}
    <button type="button" class="close" data-dismiss="alert" aria-label="Close">
```

```
        <span aria-hidden="true">&times;</span>
    </button>
</div>
{% endfor %}
<div class="container">
    <div class="row">
        <div class="col-lg-12 col-lg-offset-0">

            <div class="row pad">
                <div class="col-lg-12">
                    <div class="panel panel-primary">
                        <div class="card-body">
                            <ul class="list-group">
                                <li class="list-group-item">
                                    Username: <strong class="float-
right">{{request.user}}</strong>
                                </li>
                                <li class="list-group-item">
                                    Full Name: <strong class="float-
right">{{request.user.get_full_name}}</strong>
                                </li>
                                <li class="list-group-item">
                                    Email: <strong class="float-
right">{{request.user.email}}</strong>
                                </li>
                                <li class="list-group-item">
                                    <form action="/myprofile/"
enctype="multipart/form-data" method="POST">{% csrf_token %}
                                        {{ form.as_p }}
                                        <button type="submit">Submit</button>
                                        Profile Photo: <strong class="float-right">
<img
                                            src="{{request.user.profile.image.url}
}" class="rounded-circle" alt=""
                                            width="150px" height="150px"></strong>
                                </li>
                            </ul>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
</div>
{% endblock %}
```

FSD-PROJECT>Forum>profile_of_user>templates>register.html

```
{% extends 'basic.html' %}

{% block title %} Register {% endblock %}
{% block registeractive %} active {% endblock registeractive %}
{% block css %}
@import url(https://fonts.googleapis.com/css?family=Roboto:400,300,500);
*:focus {
  outline: none;
}

body {
  margin: 0;
  padding: 0;
  background: #DDD;
  font-size: 16px;
  color: #222;
  font-family: 'Roboto', sans-serif;
  font-weight: 300;
}

#login-box {
  position: relative;
  margin: 5% auto;
  width: 700px;
  height: 500px;
  background: #FFF;
  border-radius: 2px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.4);
}

.items {
  position: absolute;
  top: 0;
  left: 0;
  box-sizing: border-box;
  padding: 40px;
  width: 700px;
  height: 500px;
}

h1 {
  margin: 0 0 20px 0;
  font-weight: bold;
  font-size: 28px;
  text-align: center;
```

```
}

input[type="text"],
input[type="email"],
input[type="password"] {
  display: block;
  box-sizing: border-box;
  margin-bottom: 20px;
  padding: 4px;
  width: 600px;
  height: 32px;
  border: none;
  border-bottom: 1px solid #AAA;
  font-family: 'Roboto', sans-serif;
  font-weight: 400;
  font-size: 15px;
  transition: 0.2s ease;
}

input[type="text"]:focus,
input[type="email"]:focus,
input[type="password"]:focus {
  border-bottom: 2px solid #16a085;
  color: #16a085;
  transition: 0.2s ease;
}

input[type="submit"] {
  margin-top: 28px;
  width: 120px;
  height: 32px;
  background: #16a085;
  border: none;
  border-radius: 2px;
  color: #FFF;
  font-family: 'Roboto', sans-serif;
  font-weight: 500;
  text-transform: uppercase;
  transition: 0.1s ease;
  cursor: pointer;
  position: relative;
  left: 250px;
}

input[type="submit"]:hover,
input[type="submit"]:focus {
  opacity: 0.8;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.4);
```

```
  transition: 0.1s ease;
}

input[type="submit"]:active {
  opacity: 1;
  box-shadow: 0 1px 2px rgba(0, 0, 0, 0.4);
  transition: 0.1s ease;
}
{% endblock %}
{% block body %}
<br>
{% for message in messages  %}
<div class="alert alert-{{ message.tags }} alert-dismissible fade show" role="alert">
      <strong>Message : </strong> {{ message }}
      <button type="button" class="close" data-dismiss="alert" aria-label="Close">
        <span aria-hidden="true">&times;</span>
      </button>
    </div>
{% endfor %}
<div id="login-box">
    <div class="items">
      <h1>Register</h1>
        <form action="/register/" method="post"> {% csrf_token %}

      <input type="text" name="username" placeholder="Username" />
      <input type="email" name="email" placeholder="E-mail" />
      <input type="text" name="first_name" placeholder="First Name" />
      <input type="text" name="last_name" placeholder="Last Name" />
      <input type="password" name="password" placeholder="Password" />
      <input type="password" name="confirm_password" placeholder="Confirm your
password" />

      <input type="submit" name="signup_submit" value="Register" />
    </div>
  </div>

  {% endblock %}
```

CHAPTER-5

RESULTS

In this project, the deployment of the online discussion forum aimed to assess its usability, performance, and user engagement. Feedback from early adopters highlighted the intuitive navigation, effectiveness of real-time notifications, and seamless participation in discussions as notable strengths of the platform. Users appreciated the robustness of the forum's features, such as user authentication, thread management, and moderation tools, which contributed to a positive user experience. Performance metrics revealed satisfactory page load times, efficient database query responses, and stable server performance during peak usage, demonstrating the platform's capability to handle concurrent interactions effectively.

Discussions within the forum demonstrated its role in fostering meaningful exchanges and collaborative learning among participants. Case studies illustrated diverse topics discussed and the active engagement of users across various threads, showcasing the forum's potential as a knowledge-sharing platform. Challenges encountered included fine-tuning scalability measures and addressing initial technical glitches, which were mitigated through iterative development and proactive support strategies. Moving forward, planned enhancements include integrating multimedia content support, optimizing mobile responsiveness, and expanding search functionalities to enrich user experience and further enhance community interaction dynamics

5.1 SNAPSHOTS

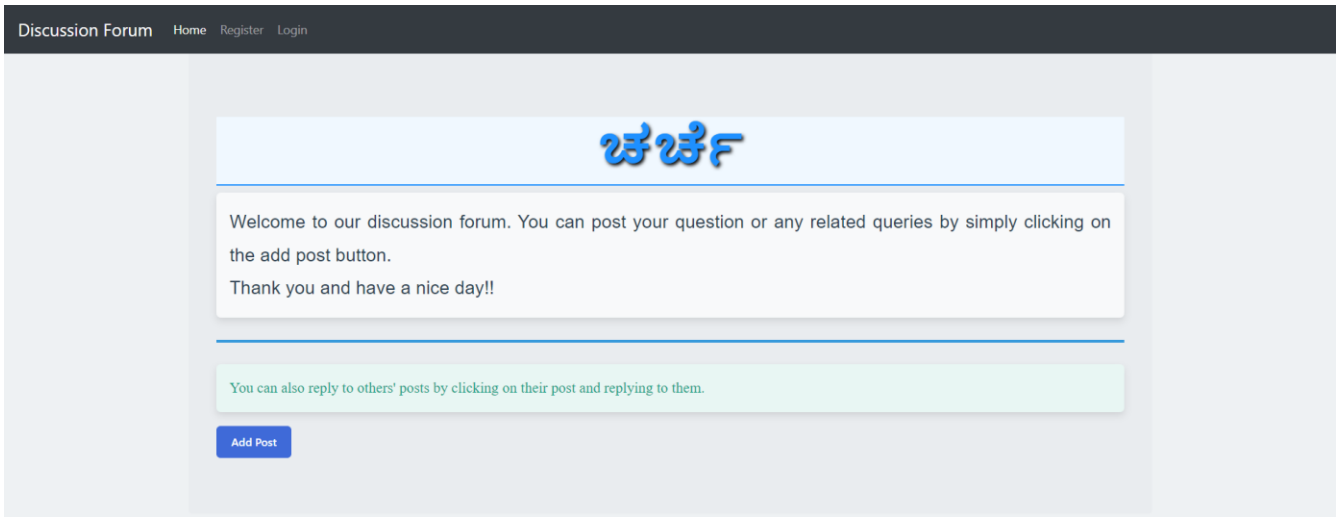

Fig 5.1.1 The home page of the Discussion forum

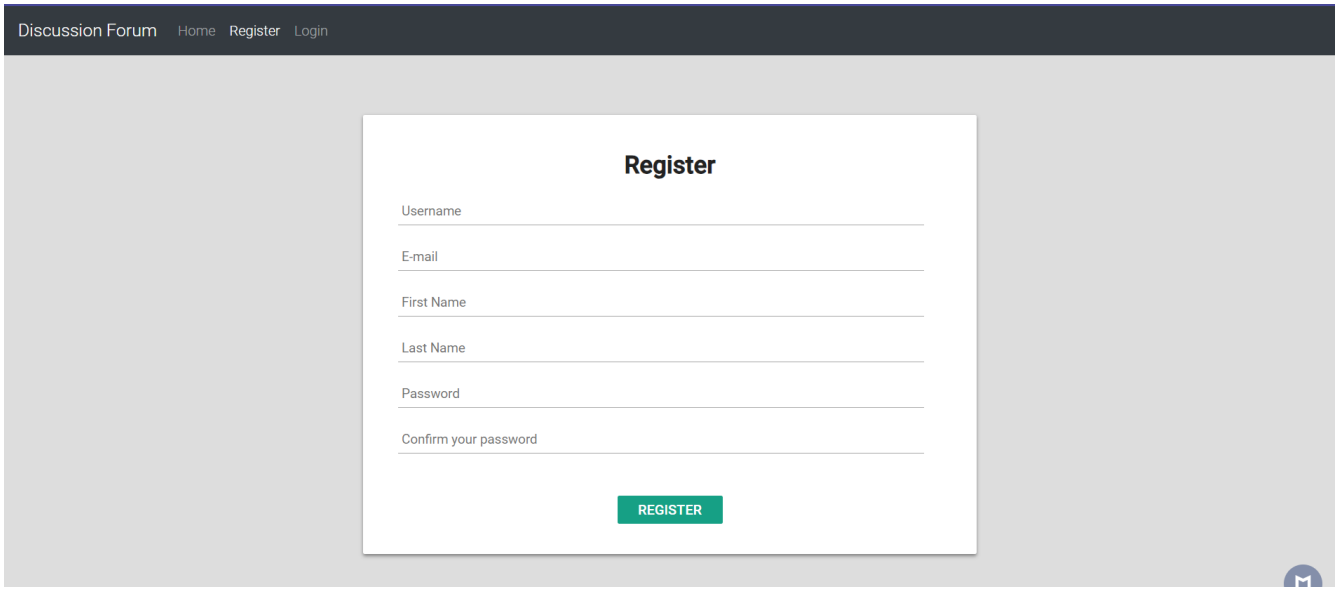

Fig 5.1.2 The register page for the Discussion forum

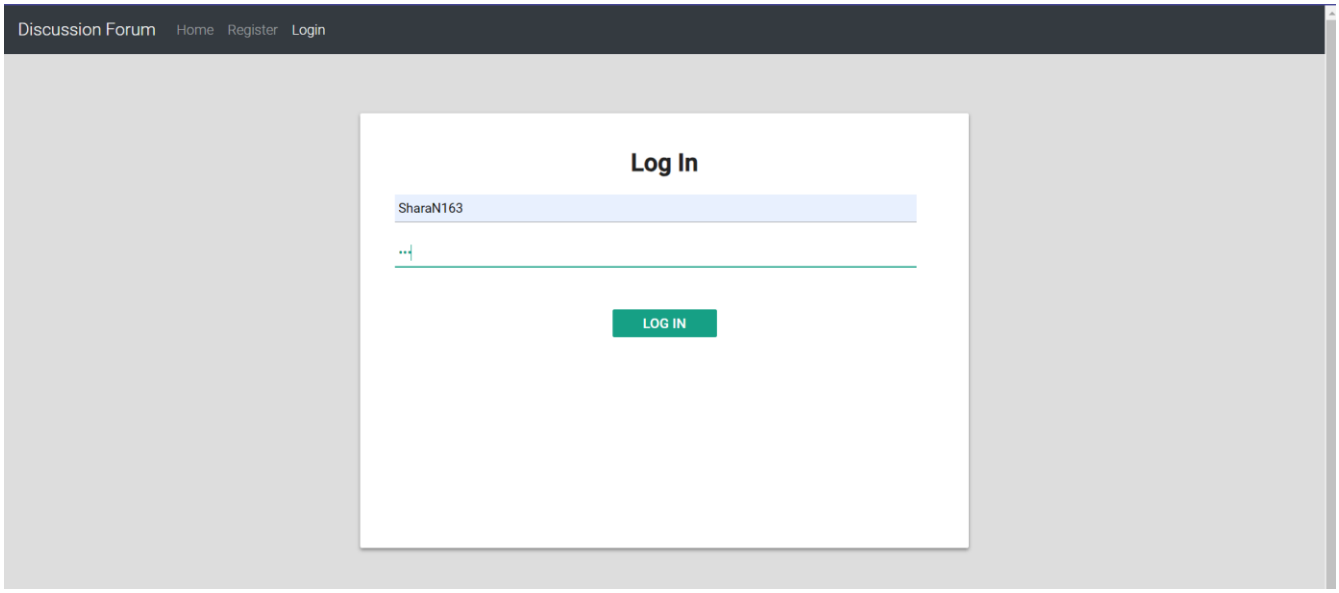

Fig 5.1.3 The login page for the Discussion forum

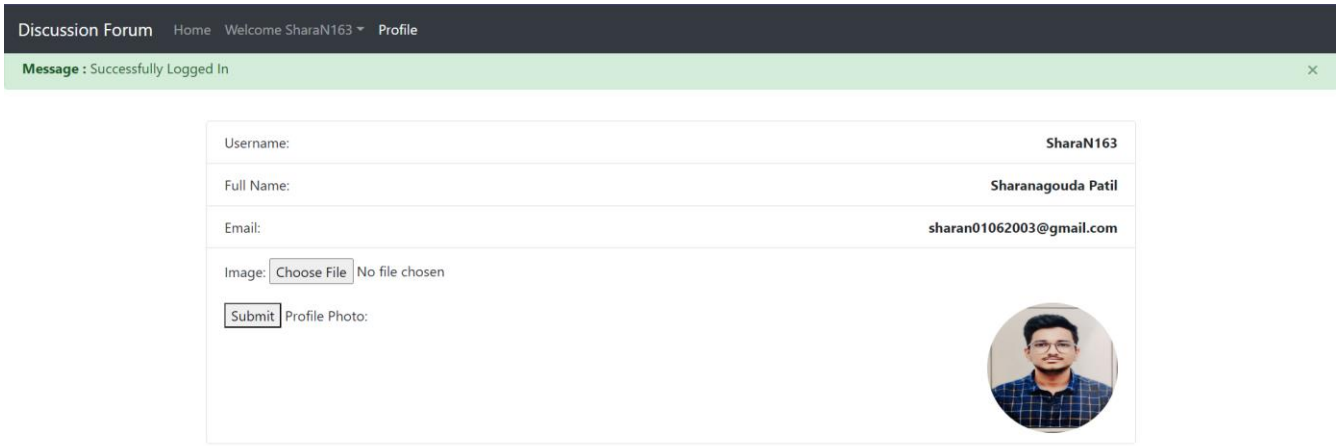

Fig 5.1.4 The profile page after successful login

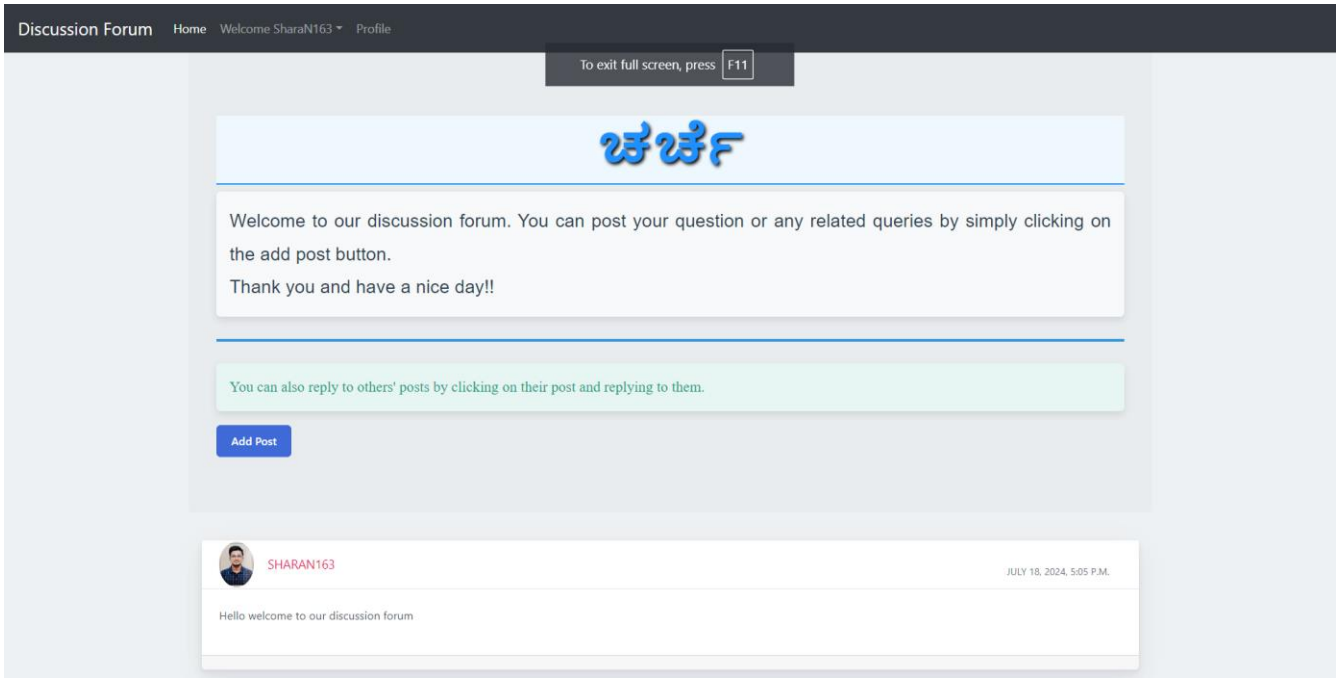

Fig 5.1.5 The discussion page where user can post once logged in

Fig 5.1.6 The thread page where any user can reply to existing post

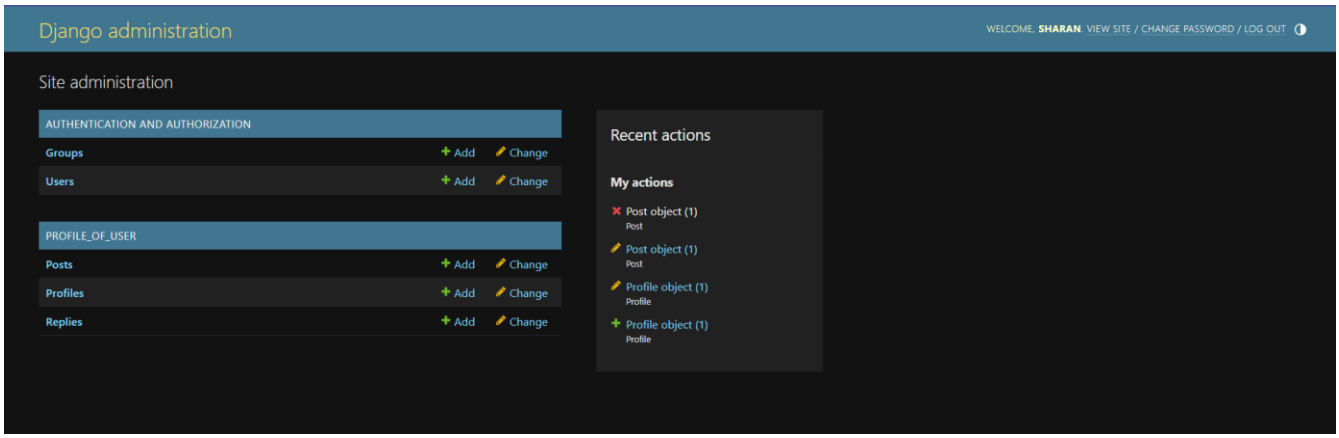

Fig 5.1.7 The admin page for thread moderation

CHAPTER-6

CONCLUSION

In conclusion, Django's powerful framework revolves around models, views, and templates to facilitate efficient web development. Models define data structure and relationships, views handle request processing and business logic, and templates generate dynamic HTML for user interfaces. Together, they form the MVC (Model-View-Controller)-like architecture, simplifying database interaction, content presentation, and user interaction. Django's robust ORM enhances productivity by abstracting SQL queries, while its templating engine ensures flexible, scalable web applications. This architecture promotes rapid development, maintainability, and scalability, making Django a preferred choice for building complex, data-driven web applications.

The full-stack online discussion forum project implemented with Django offers a robust platform for community interaction and knowledge sharing. Utilizing Django's models, views, and templates, the project manages user authentication, thread creation, post submissions, and dynamic content rendering. Integrating Django's ORM ensures efficient database interactions, while the templating system facilitates responsive and customizable user interfaces. The project supports features like user profiles, thread subscriptions, and moderation tools, fostering a collaborative environment. Overall, leveraging Django's comprehensive framework enables the creation of a scalable and feature-rich online forum, promoting engagement, knowledge dissemination, and community building among users

Future Enhancement

For a full-stack online discussion forum project using Django, future enhancements could include implementing real-time chat functionalities using technologies like Django Channels or WebSocket integration for instant messaging. Enhancing user engagement through notifications for replies, likes, and mentions would improve interaction. Introducing advanced search capabilities, tagging systems for categorization, and multimedia support for rich content sharing could enrich user experience. Additionally, integrating social authentication and improving scalability through optimized database queries and caching mechanisms would enhance performance. Lastly, implementing robust moderation tools and analytics for user activity and content management would ensure a safe and productive community environment

REFERENCES

[1] For Django:

"Official Django Documentation" The Django documentation provides comprehensive guides and references for setting up Django projects, creating models, views, templates, and handling user authentication.

- https://docs.djangoproject.com/en/5.0/

[2] For HTML/CSS

- https://www.w3schools.com/html/html/
- https://www.w3schools.com/css/

[3] For MODELS/VIEWS

- https://projectgurukul.org/python-django-online-discussion-forum/