

Internship Project Report

STUDENT DISCUSSION FORUM

TEAM -9

ABSTRACT

In the times of Romans it meant a public place at the center of market or town where open discussions on judicial, political, and other issues were held. As the times changed and technologies have developed the discussions in the public places have become less and people started using technology to communicate and express their views. With the changing times we have started using technology a lot and expect to have technologies which make our lives easier and faster.

So this is one of those platforms where the students of a college can communicate with others and express their views and implement those ideas accordingly. Basically this student forum is divided into two modules where one is Admin and other one is User. The admin handles all the users and their posts. And the Users can add posts and view others posts and communicate with others.

Table of Contents

	Title Page	i
	Abstract	ii
1.	INTRODUCTION	1
	1.1 Problem Definition	1
	1.2 Methodologies	2
	1.3 Outline of the results	
2.	LITERATURE SURVEY	3
	2.1 Introduction to the problem domain terminology	3
	2.2 Existing solutions	4
	2.3 Tools/Technologies used	4
3.	DESIGN OF THE PROPOSED SYSTEM/METHOD/ALGORITHM	5
	3.1 Module Description	5
4	IMPLEMENTATION OF THE PROPOSED SYSTEM	6
	4.1 Flowcharts / DFDs / ER Diagrams	6
	4.2 Algorithms / Pseudo Code	9
	4.3 Testing Process	11
5.	RESULTS / OUTPUTS AND DISCUSSIONS	13
6.	CONCLUSIONS / RECOMMENDATIONS	18
	6.1 Conclusions	
	6.2 Recommendations / Future Work /Future Scope	
	REFERENCES	19

1. INTRODUCTION

1.1 Problem Definition

Student Discussion Forum is a platform which is specially designed for students and faculty. As we see there are a lot of ways for a student communicate with other student and faculty. But there is no platform where a student's ideas or thoughts can reach all the students of the college and also the faculty.

This is a platform which helps every student to communicate with every other student irrespective of the year of their study. They can exchange ideas, suggest improvements, and also gain knowledge on the current things happening in the college as well as the industry. The final year students can suggest the juniors on how to work for their placements and projects. They can give advices. Also students can clarify their doubts be it programming related or anything else. The promotions of events for the cultural and technical fests can also be done. So this is a platform which has multiple purposes. And added advantage is even the faculty can interact with the students and suggest them improvements and help them.

So the project is divided into 2 modules:

1. Admin
2. Users

Coming to the Admin, he has the following functionalities:

1. View and manage users and their posts.
2. A section for reported users.
3. Block a user.
4. Delete a post.
5. Statistics.

The Users have the following functionalities:

1. Login page with roll number.
2. Home screen with all posts. An option to reply to a post.
3. An option to filter/search posts.
4. Add new post.
5. Profile page to view and manage user's posts.
6. An option to report users/posts.

1.2 Methodology

A lot of colleges do not have a platform like this. In this application the admin module is done in Web and the user module in Android. We have used Django Rest framework for writing REST APIs. Django rest framework allows us to write API calls by retrieving the data making post calls. And php scripts in android for data retrieval and storing data into the database. We have used Bootstrap4 for the web templates.

1.3 Outline of Results

The Admin functionalities were mentioned above so when those functionalities are done then the corresponding output of that function is stored in the database. This is similar in the User as well. For example the Admin edits a user's post then the corresponding user's post changes are reflected in the database and returns to the blog main page.

2. LITERATURE SURVEY

Student Discussion Forum is a application where students get to interact. We have used Django REST APIs, Django, Flask, Bootstrap, Android Studio for building this project.

2.1 Introduction to Problem Domain

2.1.1 Django Rest Framework

Django REST framework is a powerful and flexible toolkit for building Web APIs. The Web browsable API is a huge usability win for developers. Authentication policy includes packages for OAuth1a and OAuth2. Serialization supports both ORM and non-ORM data sources.

Serializers allow complex data such as querysets and model instances to be converted to native Python datatypes that can then be easily rendered into JSON , XML or other content types. Serialization is the process of converting an object into a stream of bytes to store the object or transmit it to memory, a database, or a file. Its main purpose is to save the state of an object in order to be able to recreate it when needed. The reverse process is called deserialization.

Django REST framework allows you to combine the logic for a set of related views in a single class, called a ViewSet . ViewSet class is simply a type of class-based View, that does not provide any method handlers such as . get() or . post() , and instead provides actions such as . list() and .

Since version 1.7, Django counts with the built-in JsonResponse class, which is a subclass of HttpResponse . Its default Content-Type header is set to application/json, which is really convenient. It also comes with a JSON encoder, so you don't need to serialize the data before returning the response object.

2.1.2 Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. 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.

- Django was designed to help developers take applications from concept to completion as quickly as possible.

- Django takes security seriously and helps developers avoid many common security mistakes.
- Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale.

Some of the real-time Django Applications are:

- Disqus. This is one of the largest Django projects so far.
- Instagram.
- Spotify.
- YouTube.
- The Washington Post.
- Bitbucket.
- Dropbox.
- Eventbrite.

2.2 Existing solutions

We have applications where the faculty posts the daily tasks, the lecture videos or related files. The student can just view those and submit their activities. But as of now we do not have an application where all the students can interact and share their views. The added advantage is the lecturers can also express their opinions and suggest on any topic or projects or anything.

2.3 Tools/Technologies used

2.3.1 Software Requirements:

- Windows 10
- Python
- Django
- Bootstrap 4
- Android Studio

2.3.2 Hardware Requirements:

- Any P.C with Windows, 256 MB RAM (Client).
- Internet connection with 33.6 KBPS Modem.

3. DESIGN OF THE PROPOSED SYSTEM

3.1 Module Description

3.1.1 Database Creation

We firstly need a database to store the details of admin, users, posts and their comments. So in admin table we have the admin roll number or any unique ID for logging in, their name, login time. Coming to users table we have student roll number as primary key, student name, login time, number of persons reported that person. The posts table has sno which is primary key for each post, roll number, post title, post content, number of persons reported for a user. And coming to comments we have unique id for each reply and to which post they are replying that post id, post content and date.

3.1.2 Processing User functionalities

The user can login or register; he/she can add a new post, report a user and manage his/her profile. So this was done in android. When the user performs an activity the backend responds to their request and responds. The response is given to the user in the UI.

3.1.3 Processing Admin functionalities

The admin can login or register; he/she can view all posts, report a user and manage his/her posts, report a post, block a user. This was done in Django. We have used Django rest framework for writing API calls. When the user performs an activity the backend responds to their request and responds. The response is given to the admin in the web.

4. IMPLEMENTATION OF THE PROPOSED SYSTEM

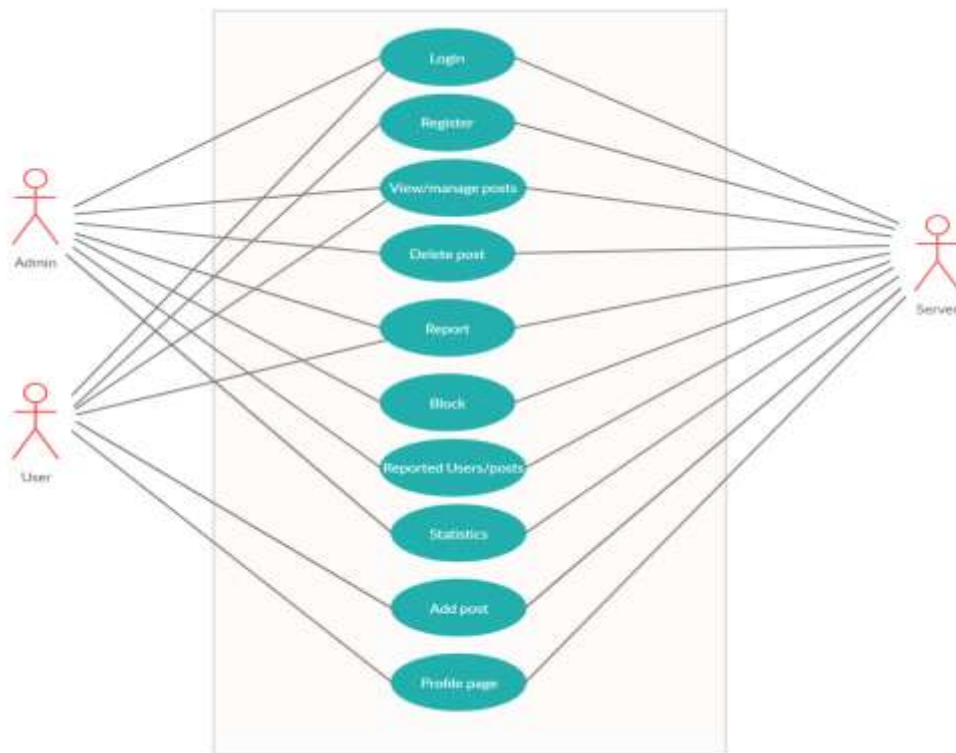
4.1 Use Case Diagram

In the Unified Modeling Language (**UML**), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors.

A use case diagram doesn't go into a lot of detail. It is a proper use case diagram depicts a high-level overview of the relationship between use cases, actors, and systems. Experts recommend that use case diagrams be used to supplement a more descriptive textual use case.

Its components include:

- **Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.
- **System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.
- **Goals:** The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal



4.2 Class Diagram

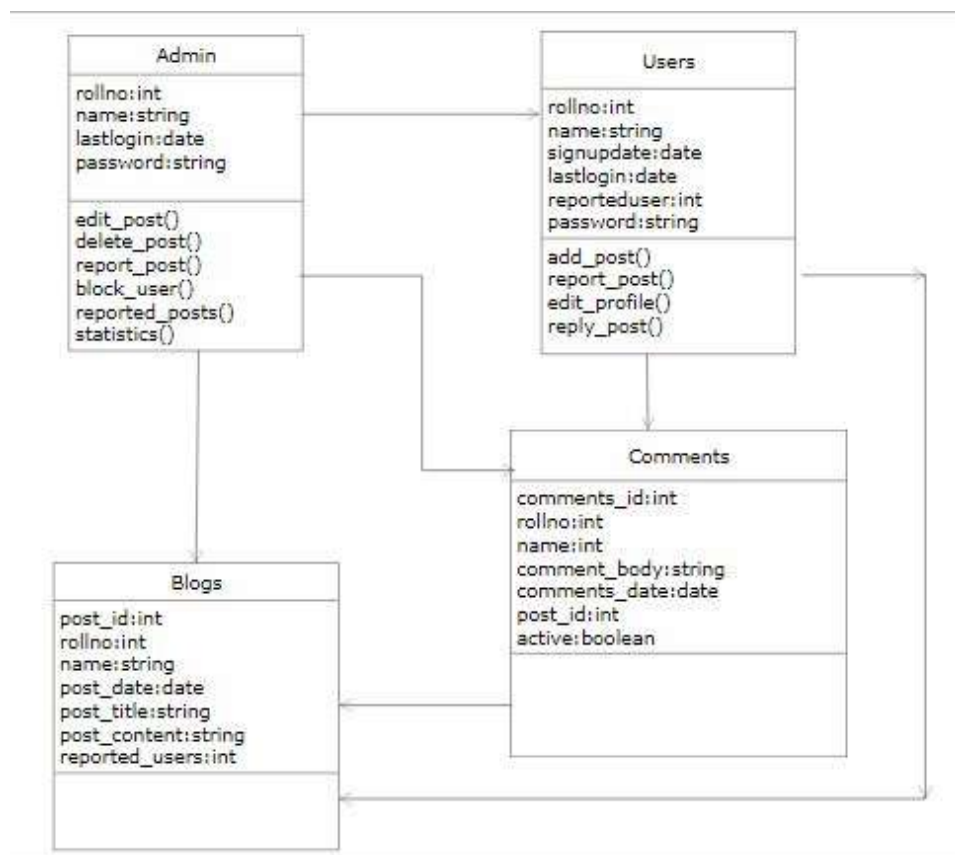
Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of the class diagram is –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.



4.3 Database Structure

Admin:

	rollno	name	lastlogin	password
▶	1	Sagnik	2020-07-02 16:17:05	sagnik123
	2	Vishnu Priya	2020-06-30 20:19:16	priya123
*	NULL	NULL	NULL	NULL

User:

	rollno	name	signup_date	lastlogin_date	reported_users	password	block_status
▶	1	John Smith	2020-06-15 18:09:15	2020-06-16 18:19:15	NULL	abc123	1
	2	vishnupriyag	2020-06-29 06:19:28	2020-06-29 06:19:28	0	priya123	0
	3	Sagnik	2020-06-29 11:00:00	2020-06-29 11:00:00	0	sagnik	1
	4	rahul	2020-06-29 11:00:00	2020-06-29 11:00:00	0	rahul	1
	101	priya	2020-06-18 12:44:03	2020-06-18 13:44:03	NULL	priya	1
	144	vishnupriya	2020-06-18 12:44:03	2020-06-18 13:44:03	NULL	priya144	1
	200	ravi	2020-06-29 16:00:00	2020-06-29 16:00:00	0	ravi	1
	1234	Rajiv	2020-06-29 13:00:00	2020-06-29 13:00:00	0	rajiv123	1
	1332	david	2020-06-30 15:00:00	2020-06-30 15:00:00	0	david	1
	12345	hruday	2020-06-29 13:00:00	2020-06-29 13:00:00	0	hruday	1
	2147...	user1	2020-06-29 11:00:00	2020-06-29 11:00:00	0	123456	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Blogs:

	sno	rollno	name	post_date	post_title	post_content	reported_user
▶	4	101	priya	2020-06-19 10:10:20	Second Blog !!	Hello everyone, lets discss!! heloooo	1
	6	144	vishnupriya	2020-07-02 06:00:00	First Blog	Hello everyone!!	0
	7	144	vishnupriya	2020-06-19 07:10:20	Second Blog	Hello everyone, lets try something new!!	0
	8	1	John Smith	2020-06-21 08:10:15	Discussion	Lets start a discussion	0
	9	101	priya	2020-06-21 08:10:15	Heyy guys	Lets start a discussion	0
	15	144	vishnupriya	2020-06-29 12:00:00	My first post	Welcome to my first post	0
	17	144	vishnupriya	2020-06-29 12:00:00	Machine Learning Updated	Machine Learning is new trend	0
	18	144	vishnupriya	2020-06-29 12:00:00	Web Development	I love developing applications in WEB	0
	20	144	vishnupriya	2020-06-29 12:00:00	Data Science	Data Analysis	0
	21	12345	hruday	2020-06-29 13:00:00	Android Development	<p>I love App Development </p>	1
	22	3	sagnik	2020-06-30 16:00:00	Foosball	Foosball is amazing	0

Comments:

	comments_id	rollno	name	comment_body	comments_date	post_id	active
▶	6	101	priya	Heyy	2020-06-29 19:07:15	8	1
	18	144	vishnupriya	Amazing	2020-06-30 14:00:00	6	1
	20	144	vishnupriya	Really Great	2020-06-30 14:00:00	6	1
	21	144	vishnupriya	Really Great	2020-06-30 14:00:00	6	1
	22	144	vishnupriya	Really Great	2020-06-30 14:00:00	6	1
	28	144	vishnupriya	bhdnte	2020-06-30 14:00:00	6	1
	29	144	vishnupriya	Very Good	2020-06-30 14:00:00	4	1
	30	3	sagnik	One of my favourite articles	2020-06-30 15:00:00	17	1
	31	1332	david	Definitely	2020-06-30 15:00:00	9	1
	32	1332	david	Wow	2020-06-30 15:00:00	6	1

4.4 Pseudo Code

```
from flask import Flask, jsonify
```

```
from flask_restful import Api
```

```
from flask_jwt_extended import JWTManager
```

```
from resources.user import UserRegister, UserLogin, UserBlogs, InsertBlogs, SearchBlog, DeleteUserPost, FindBlogs, EditBlogs, UserReplies, AddReplies
```

```
from resources.admin import AdminRegister, AdminLogin, GetAllBlogs, BlockUser, DeletePost
```

```
from datetime import datetime, timedelta
```

```
app = Flask(__name__)
```

```
app.config['PROPAGATE_EXCEPTIONS'] = True
```

```
app.config['JWT_SECRET_KEY'] = 'coscskillup'
```

```
api = Api(app)
```

```
jwt = JWTManager(app)
```

```
@jwt.unauthorized_loader
```

```
def missing_token_callback(error):
```

```
    return jsonify({
```

```
        'error':'authorization required',  
        "description":"Request does not contain an access token"  
    }}, 401
```

```
@jwt.invalid_token_loader
```

```
def invalid_token_callback(error):
```

```
    return jsonify({  
        'error':'invalid token',  
        "description":"Signature verification failed"  
    }}, 401
```

```
api.add_resource(UserRegister, '/userregister')
```

```
api.add_resource(AdminRegister, '/adminregister')
```

```
api.add_resource(UserLogin, '/userlogin')
```

```
api.add_resource(AdminLogin, '/adminlogin')
```

```
api.add_resource(GetAllBlogs, '/getblogs')
```

```
api.add_resource(UserBlogs, '/userblogs')
```

```
api.add_resource(InsertBlogs, '/insertblog')
```

```
api.add_resource(SearchBlog, '/searchblog')
```

```
api.add_resource(BlockUser, '/blockuser')
```

```
api.add_resource>DeletePost, '/deletepost')
```

```
api.add_resource>DeleteUserPost, '/deleteuserpost')
```

```
api.add_resource(FindBlogs, '/findblog')
```

```
api.add_resource>EditBlogs, '/editblog')
```

```
api.add_resource>UserReplies, '/userreply')
```

```
api.add_resource(AddReplies, '/addreply')
```

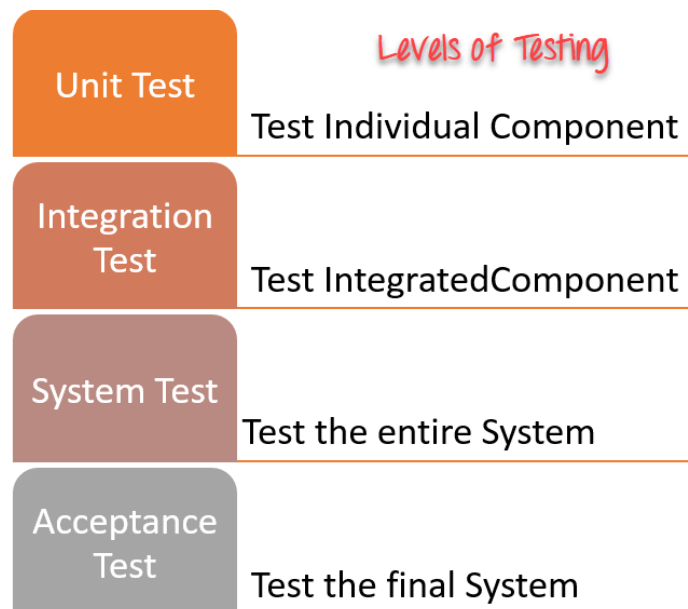
```
if __name__ == '__main__':
```

```
    app.run()
```

4.5 Testing Process:

Testing is the process of evaluating a system or its component's with the intent to find that whether it satisfies the specified requirements or not .This activity results in the actual, expected and difference between their results, i.e testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

In order to make sure that system does not have any errors, the different levels of testing strategies that are applied at different phases of software development are



4.5.1 Unit Testing:

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

4.5.2 Integration Testing:

The testing of combined parts of an application to determine if they function correctly together is Integration testing .This testing can be done by using two different methods

4.5.2.1 Top-down Integration Testing:

In Top-Down integration testing, the highest-level modules are tested first and then progressively lower-level modules are tested.

4.5.2.2 Bottom-up Integration Testing:

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time .When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

4.5.3 System Testing:

This is the next level in the testing and tests the system as a whole .Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards.

4.5.4 Acceptance Testing:

The main purpose of this Testing is to find whether application meets the intended specifications and satisfies the client's requirements .We will follow two different methods in this testing.

4.5.4.1 Alpha Testing:

This test is the first stage of testing and will be performed amongst the teams .Unit testing, integration testing and system testing when combined are known as alpha testing. During this phase, the following will be tested in the application:

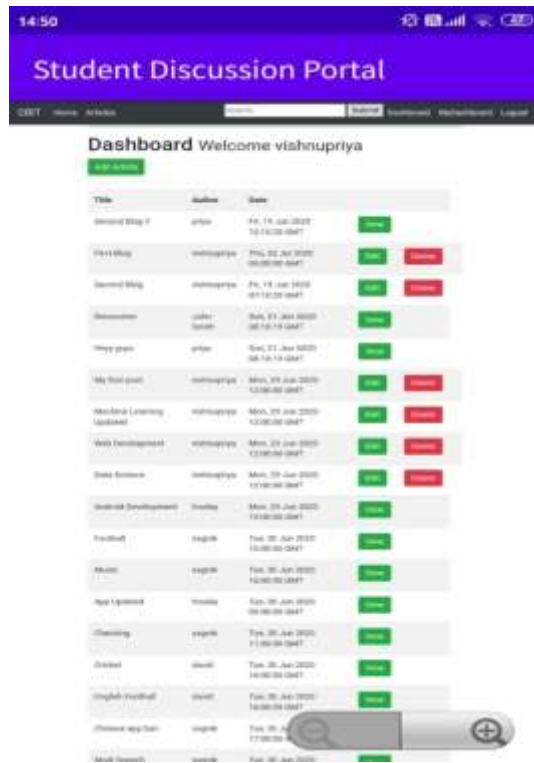
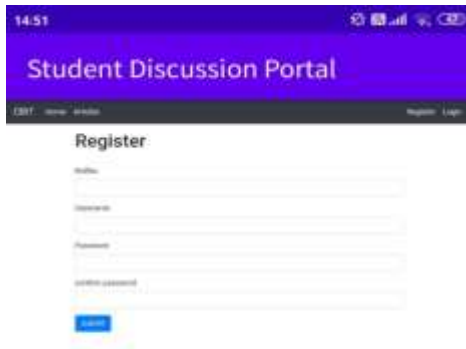
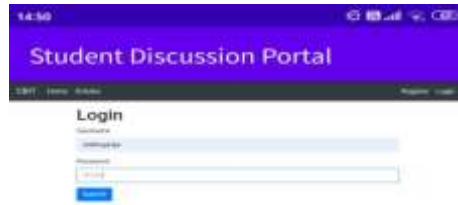
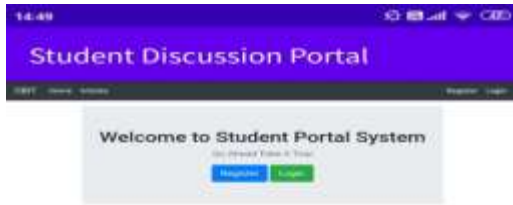
- Spelling Mistakes.
- Broken Links.
- The Application will be tested on machines with the lowest specification to test loading times and any latency problems.

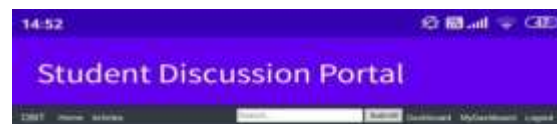
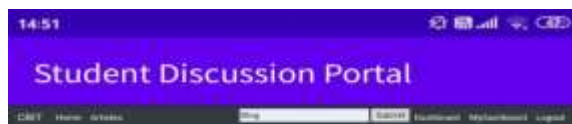
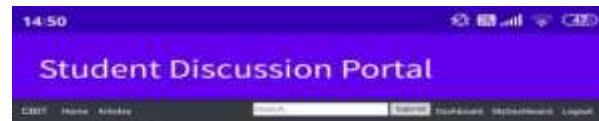
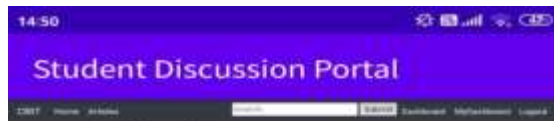
4.5.4.2 Beta Testing:

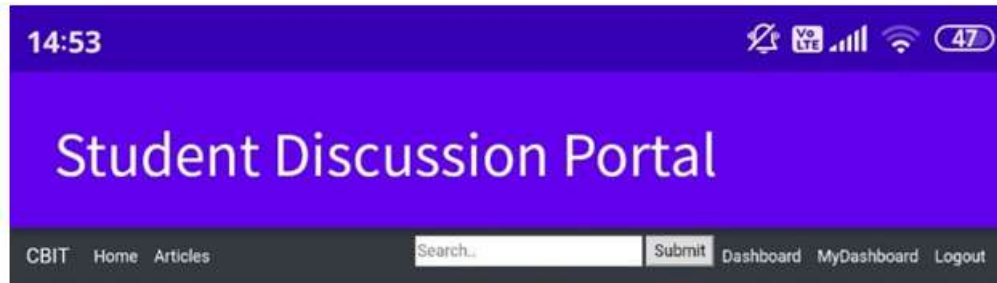
In beta testing, a sample of the intended audience tests the application and sends their feedback to the project team .Getting the feedback; the project team can fix the problems before releasing the software to the actual users.

5. Results / Output and Discussions

Users:







Edit Article

Title

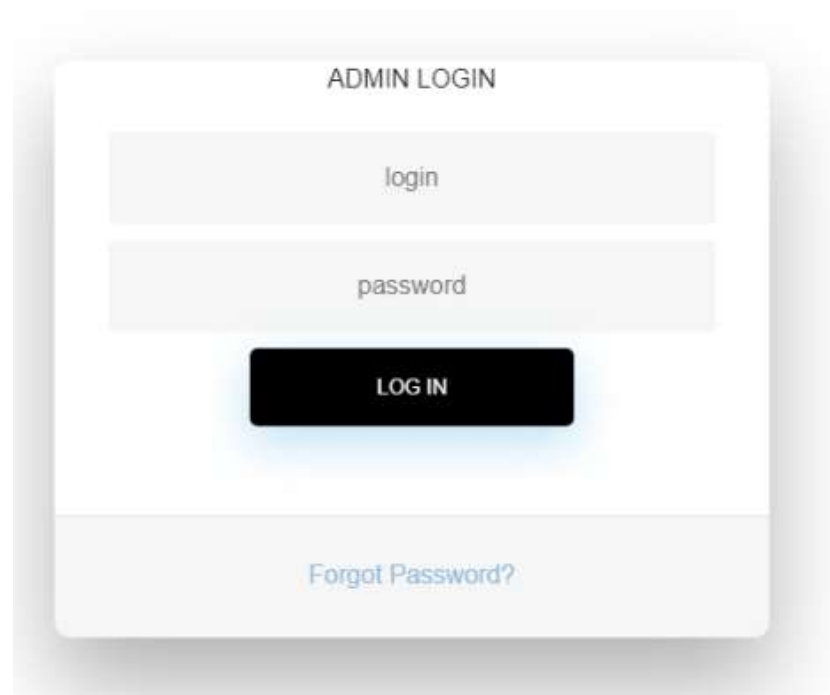
First Blog

Body

Hello everyone!!

Submit

Admin:



Welcome to the Student Discussion Forum

Share your ideas and interact with each other.!

Second Blog !!

priya | June 15, 2020, 10:10 a.m.

Hello everyone, lets discuss!! hellooo

[Read More -->](#)[Edit](#) [Delete](#) [Report](#) [Block User](#)

First Blog

vishnupriya | July 2, 2020, 6 a.m.

Hello everyone!!

[Read More -->](#)[Edit](#) [Delete](#) [Report](#) [Block User](#)

Second Blog

vishnupriya | June 19, 2020, 7:10 a.m.

Hello everyone, lets try something new!!

[Read More -->](#)[Edit](#) [Delete](#) [Report](#) [Block User](#)

Discussion

John Smith | June 21, 2020, 8:10 a.m.

Lets start a discussion

[Read More -->](#)

Title

Content

Welcome to the Student Discussion Forum

Share your ideas and interact with each other..!

Date	Number of posts
2020-Jun-19	2
2020-Jul-02	1
2020-Jun-21	2
2020-Jun-29	5
2020-Jun-30	8

@Team9

Welcome to the Student Discussion Forum

Share your ideas and interact with each other..!

User	Post Title	Number of people reported
priya	Second Blog !!	1
hruday	Android Development	1

@Team9

6. Conclusion/ Recommendations

6.1 Conclusions

We all need a platform where we can share our thoughts, discuss on any issue, clarify any doubts, and get suggestions and advices on career and future. There a lot of things like this where we want to hear things from people who are settled in a great position or people having great experiences from in various things. Not just about studies it can be anything. So we all needed a platform like this. So here we are with a great platform where we can have all these things.

We wanted to make things easier for college students so came up with this. We have successfully built an application with various functionalities.

6.2 Future Scope

Initially we would like to release it and see how students and faculty would respond. Based on their experience and suggestions we would like to improve it further.

References

[1] Stack Overflow

[2] <https://djangocentral.com/creating-comments-system-with-django/>

[3] <https://ultimatedjango.com/blog/how-to-consume-rest-apis-with-django-python-reques/>