Chapter -6 Curiosity Room

6.1 An overview of the project

The Project comprises of a web-app that deals with user-generated content such as-
- links
- text-based posts

The platform allows for discussions on the said posts in what is essentially a bulletin board system. Users can create threads, post on these threads, and can have discussions regarding various topics with other users.

'Bots' have been employed to manage the posted content. This has been done to ensure that the posted content and subsequent discussions are 'civil' so as to not hurt the sentiments of a particular user/group of users in any way, shape or form.

Filtering out explicit or abusive content is an essential feature towards the stated purpose.

Deep learning models have been applied to filter out offensive posts, and if found offensive/hurtful the said post will be flagged. The user whose post is flagged can make an appeal in which case the post will be manually checked by a moderator.

On the same lines, a user can 'report' a post as offensive/hurtful and if it is found to be offensive, the deep learning system adjusts accordingly learning from the new experience thereby getting better as more and more users join the platform and participate actively.

Users also have the ability to create various 'threads'-
which essentially act like 'rooms' prompting discussion on various topics of interest. Once created, users have the provision of contributing to said 'threads' by posting information in the forms mentioned above.

Distinguishing between popular content which is more likely to be 'genuine' and 'authentic'

is imperative for any user. The platform provides this functionality in the form of an 'upvote-downvote' system.

Providing users with the ability to 'upvote/downvote' does not only provide an efficient way to discern which posts are popular on the platform thereby indicating high user interest, but it also adds an element of interactivity providing a user with a sense of involvement in the community and a means to express his/her liking or disliking of the content posted.

several systems in use. Few components/features are listed:

6.1.1 Home:

The homepage of the web app has a global feed with all the different posts/rooms created from different users present. Irrespective of the fact whether a user is logged in tor not hey can access the global feed and view the post and comments related to it.
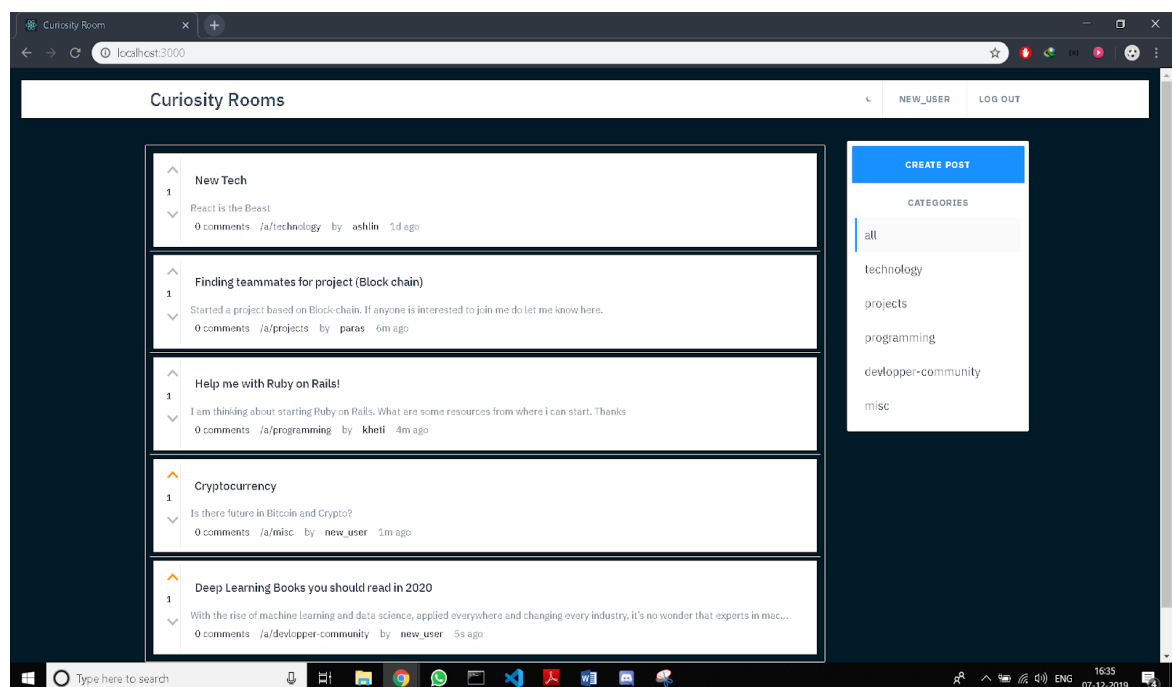

Fig.6.1 Homepage showing global feed

6.1.2　Posts and Comment:

Every post is made by a logged in user has its own page where other user can comment and raise queries or support the post with new similar thoughts. Apart from that number of views for that particular posts is also mentioned.
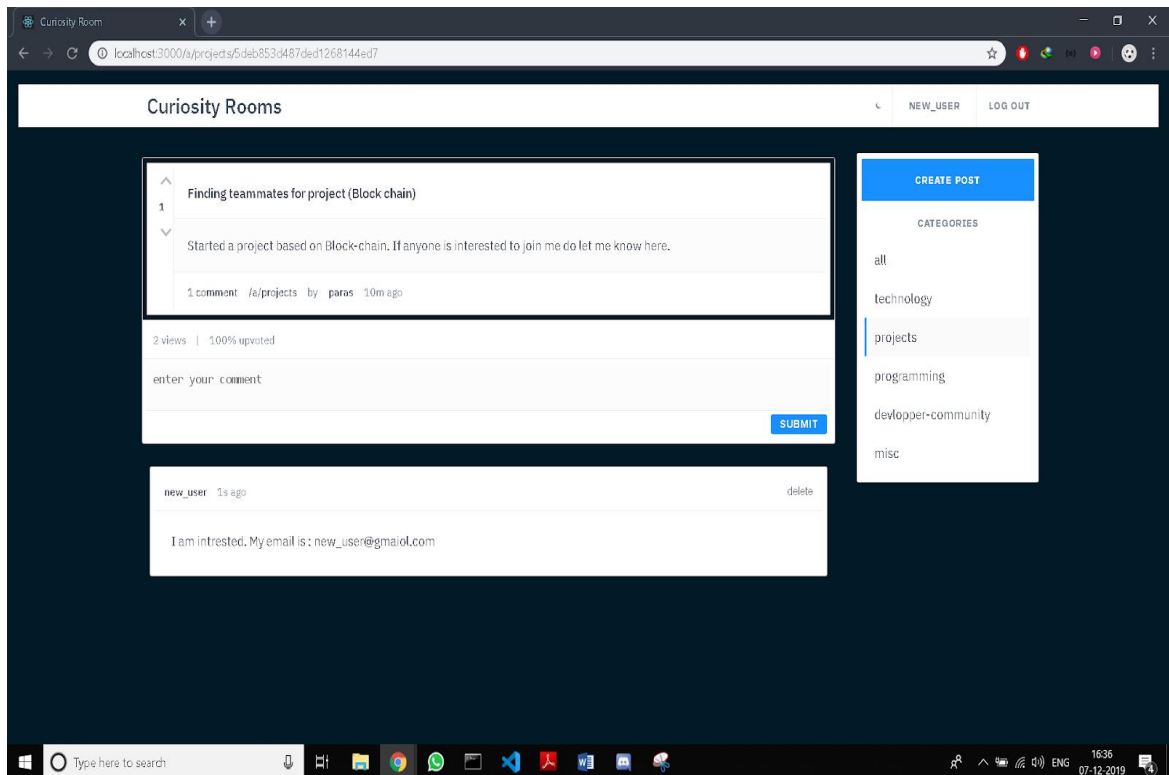

Fig.6.2　A Chat Room

6.1.3　Promote and Demote:

A user can either promote or demote a post by using the arrow keys present to the left of each post. A posts value can be determined by knowing the number of promotes and demotes it received.
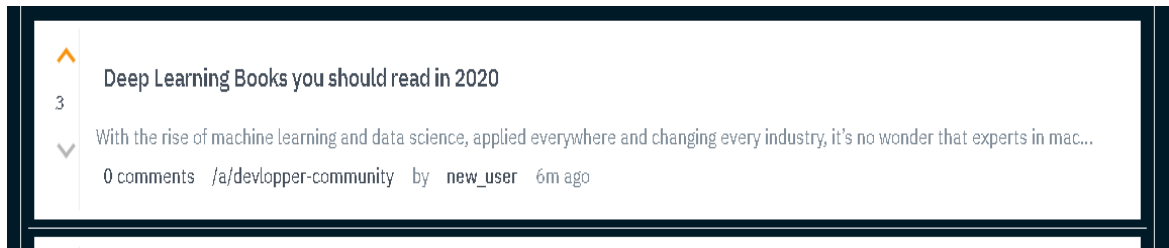
Thus it acts as motivation

Fig. 6.3 Promotes and Demotes on a post

6.1.4   Login and Signup:

A user can login using the login button present at the top right corner of the screen. User can also create a new id by using signup. For a user to create posts and add comments to others posts. Also, signup provides user to view all the posts made by them.
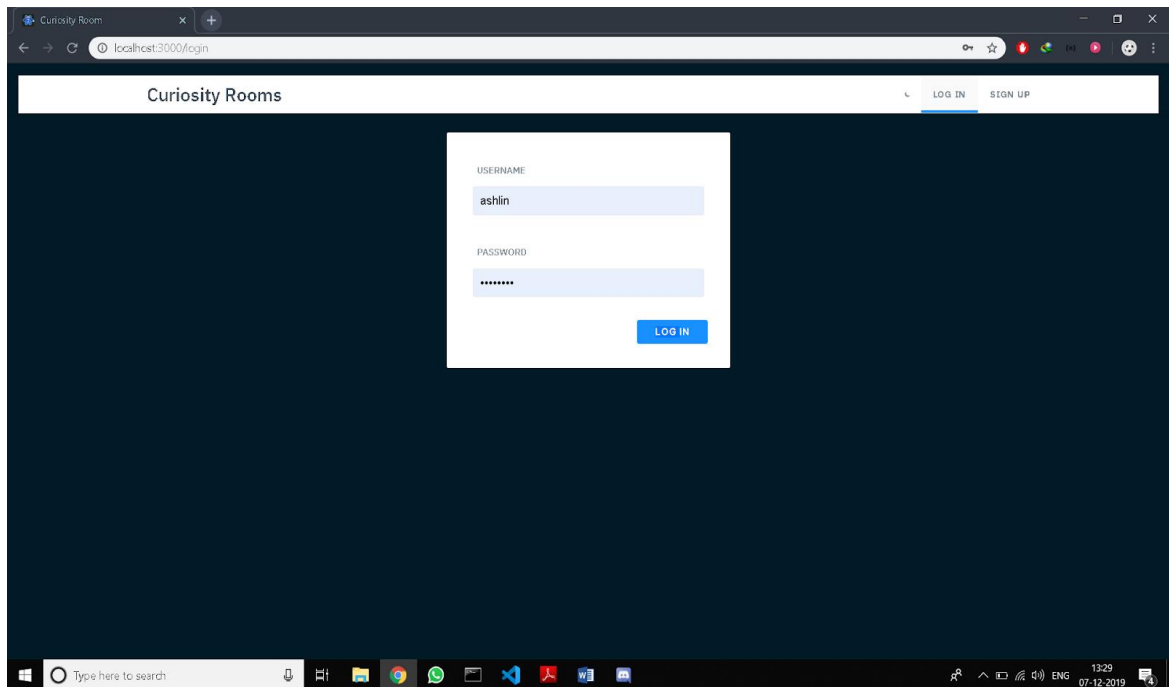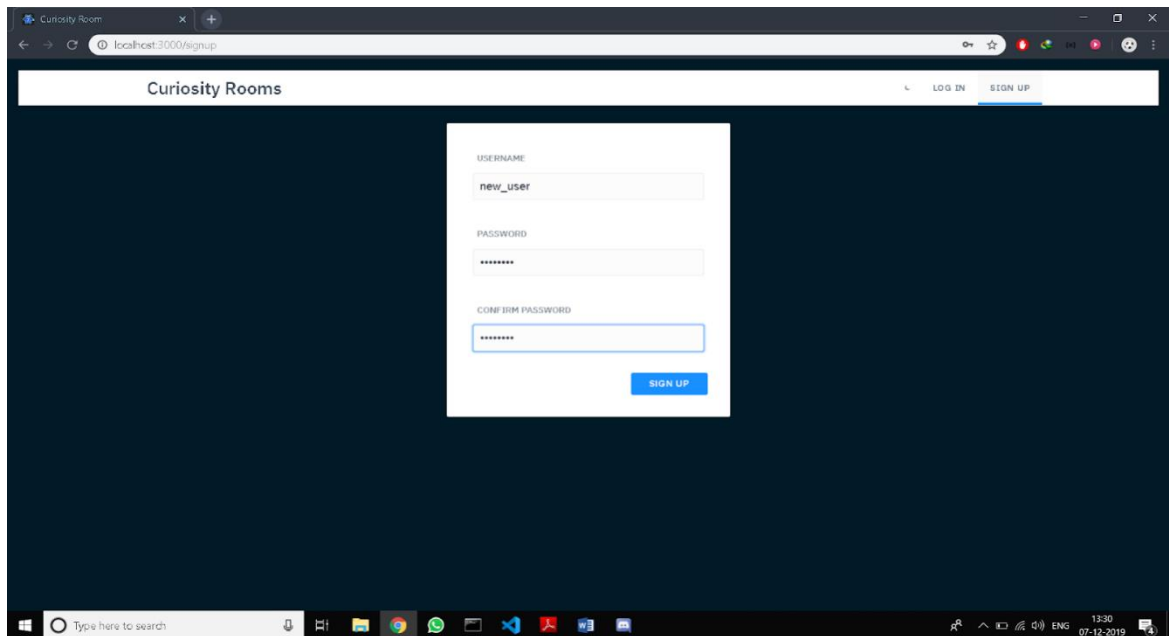

Fig.6.4 Log In Page

Fig.6.5 Sign Up Page

6.1.5 Category Pane:

Each post has a category associated to them, so we can filter all those posts related
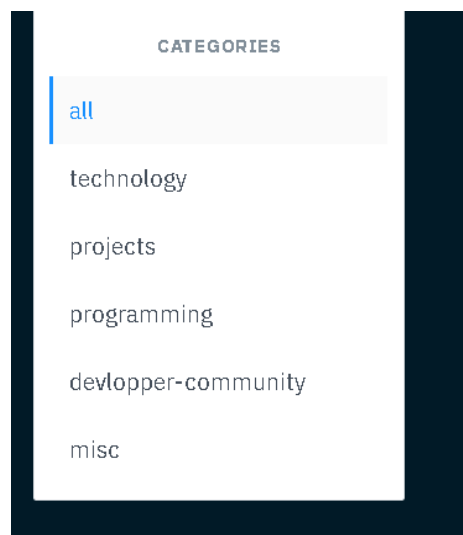that category by choosing the corresponding category from the category pane.

Fig.6.6 Showing different categories

6.1.6   Dark Mode:

A button is provided at the top right to switch the current UI to a dark user interface which enhances visual ergonomics by reducing eye strain.
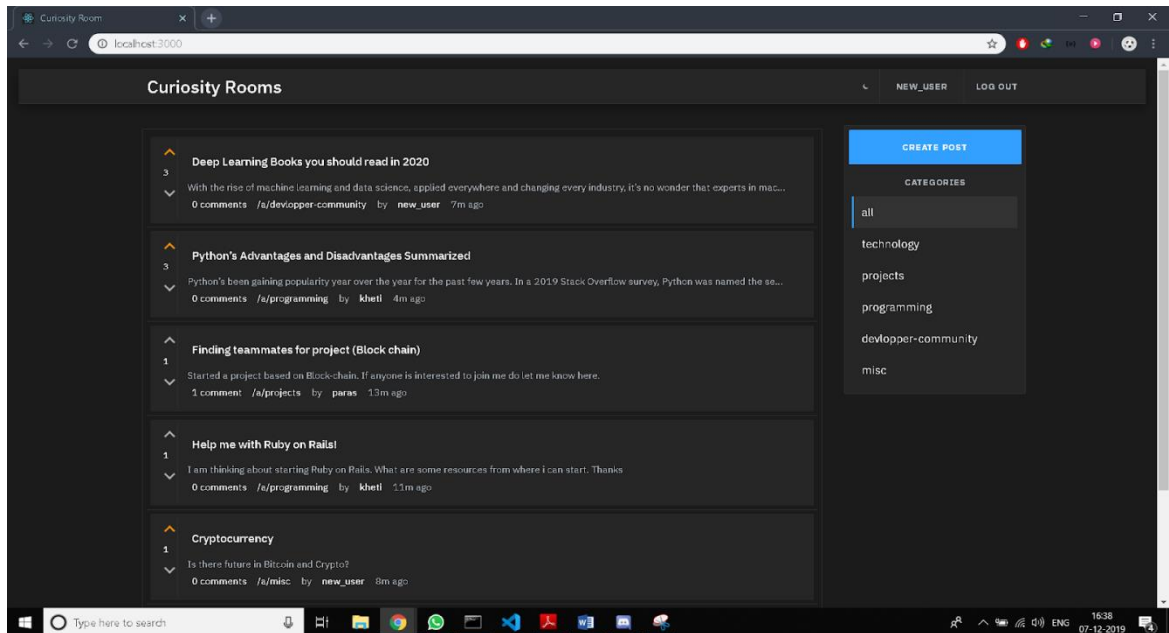

Fig.6.7 Dark Mode

6.1.7   Create a Post:

A user can create a post by clicking the create post button .Once user clicks the button, the create post window opens. It has the following fields:

1. **Category:**

The user can select the category from a predefined set of categories

2. **Title:**

The user can select the title for the post, the tile cannot be empt

3. **Text :**

This will act as the body of the post, the user can write text here or posts various links here
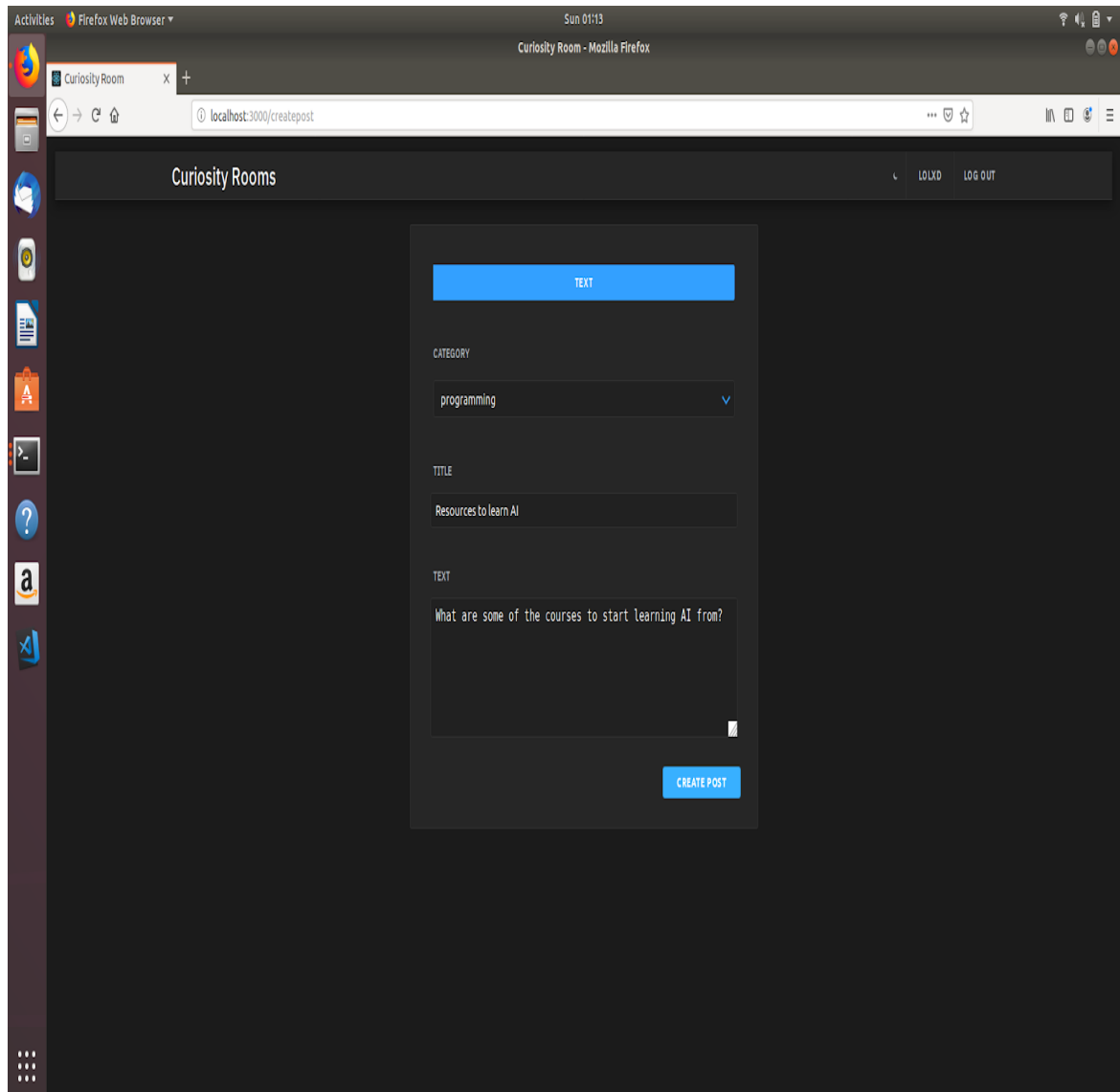

Fig 6.8 Create a Post

6.1.8 Post Moderation:

Generally, the post moderation is done by the moderators of the platform. But if the user engagement is high, it might be difficult for humans to moderate the posts and it would no longer remain efficient.

Thus rather than using human moderation at first step itself, we first a neural network to perform text classification so as to make moderation more efficient.

6.1.9 Text Classification:

For the task of text classification, we use a AWD LSTM model. LSTM stands for long short term memory , and specifically we use a weight dropped LSTM.

We use fastai to train and create our model , and the dataset has ben taken from various Kaggle datasets.We try to classify the text in three major categories:

1 Offensive:

This category contains text which might be vulgar , contain a lot of expletives ,etc

2. Hate Speech:

Hate speech is extremely offensive and that attacks a person or a group on the basis of sensitive attributes such as race

3. Okay:

This is the permissible text.This is the text we will allow in our platform

- **Training our AWD LSTM Model**
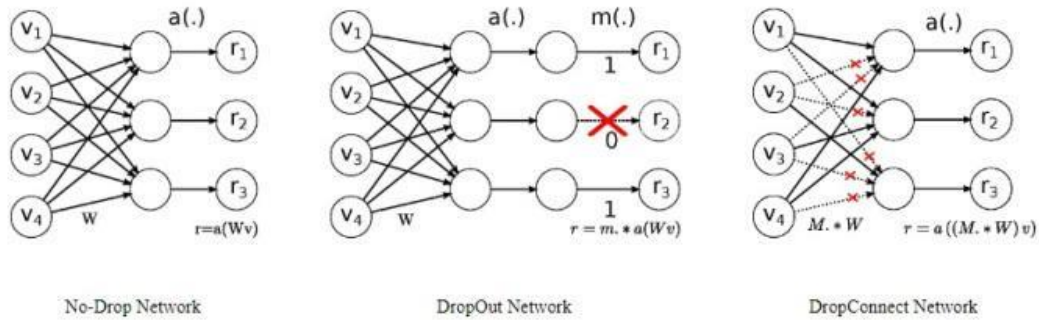
AWD LSTM using drop connect Layers.

Fig 6.9 Drop Connect

For training our model, we use multiple learning rates and gradually keep on unfreezing the layer.
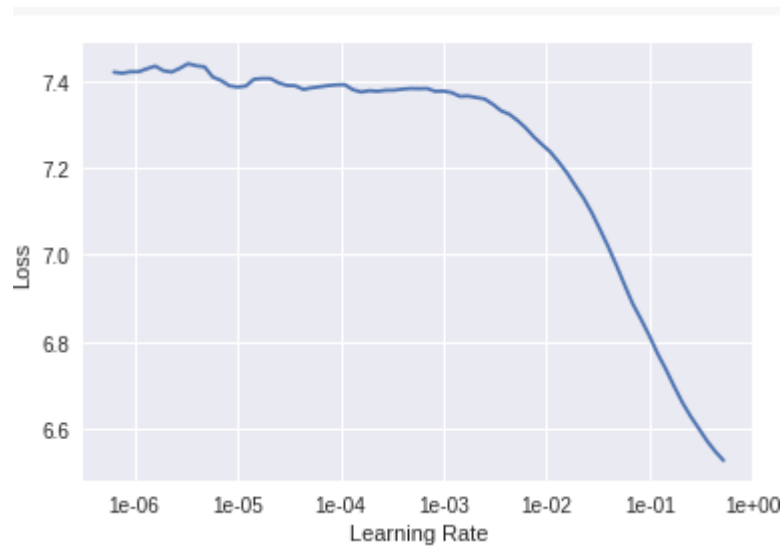
- **Various Learning Rates:**

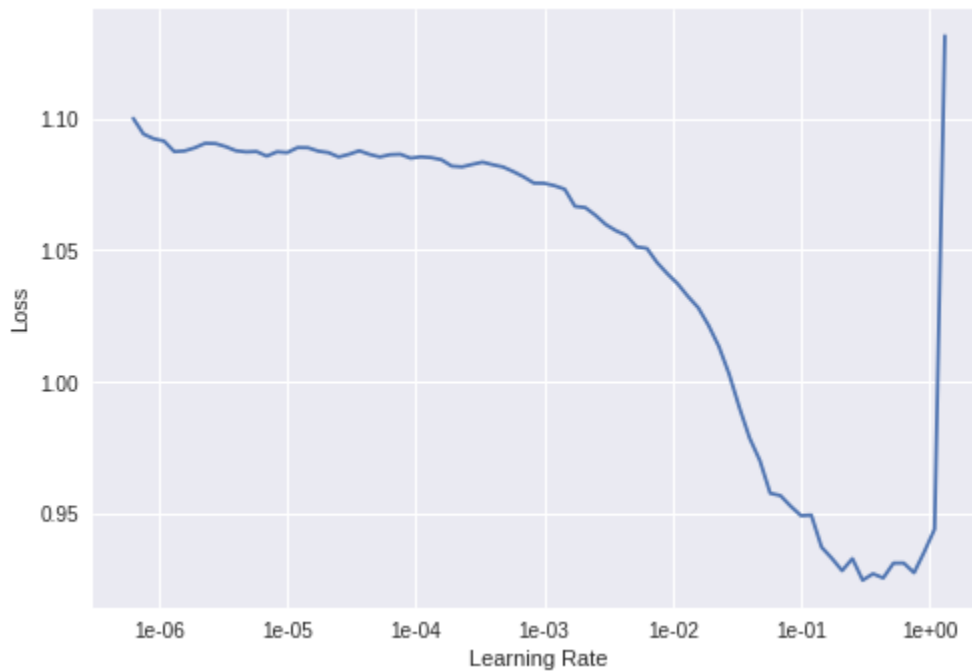Fig. 6.10 Learning rate when only output layer is unfreezed

Fig. 6.11 Learning rates when multiple layers are unfreezed.

We use an Encoder first which is pretrained on the Wikitext-103 dataset and then train on our own dataset.

Then once our encoder is trained, we use that encoder for our text classification, and at our initial attempt , we were able to achieve 89.8% accuracy.

- **REST API:**

Using flask, we create a REST API for the model, so that we could use our model in various application.

- **POST Moderation**

Whenever the uses presses the create a Post button, the text of our post is sent to our model, using a POST request and we receive the probabilities of the various categories as the output. If the probability of okay is greater than 50 percent , the post is allowed,

otherwise , an alert is shown to the user to warn him/her about the content of the message, the user may be able to send the post for further moderation.
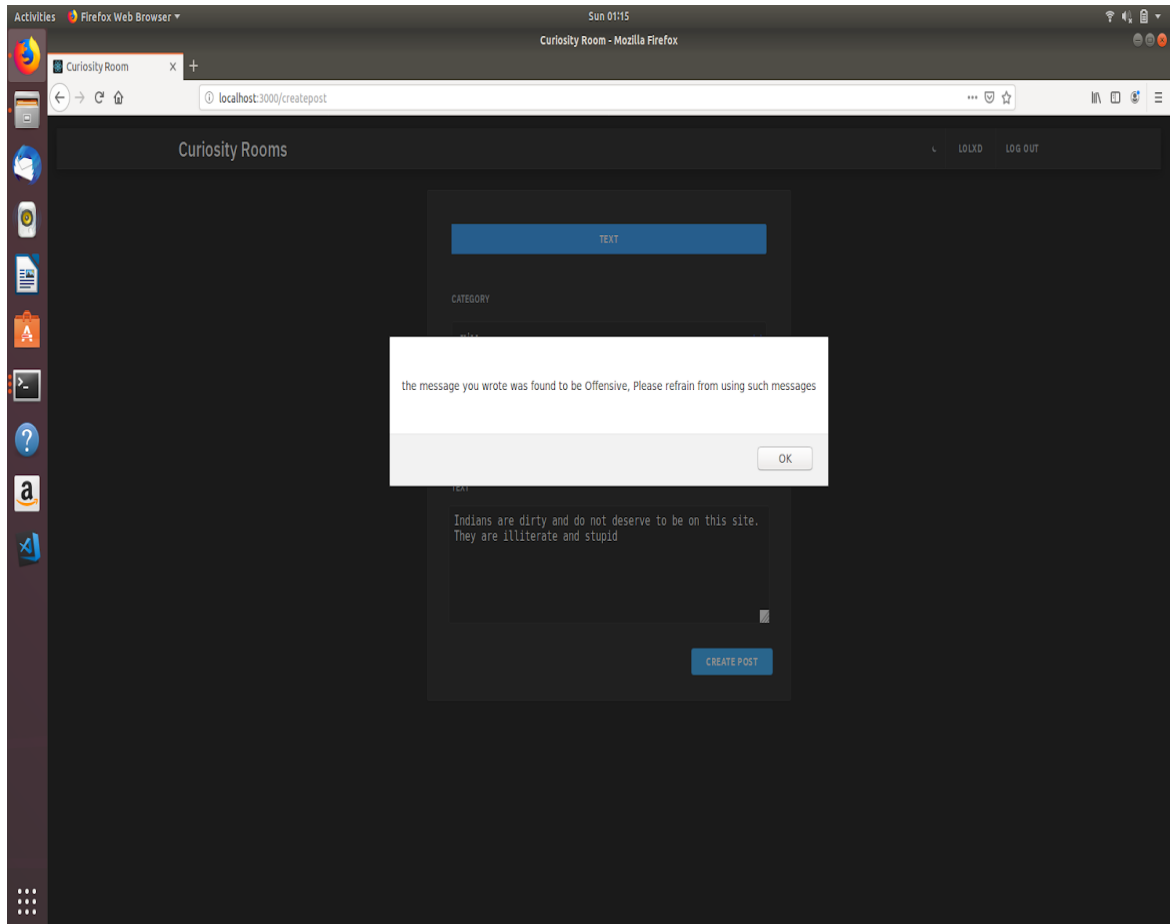

Fig 6.12 Text Moderation

- **Comment Moderation:**

To maintain a civilized conversation , not only the post but the comment itself must be moderated.

Similar to the post moderation, once a user posts a comment first it is sent to our model to classify, and if found to be offensive, the user is shown an alert.

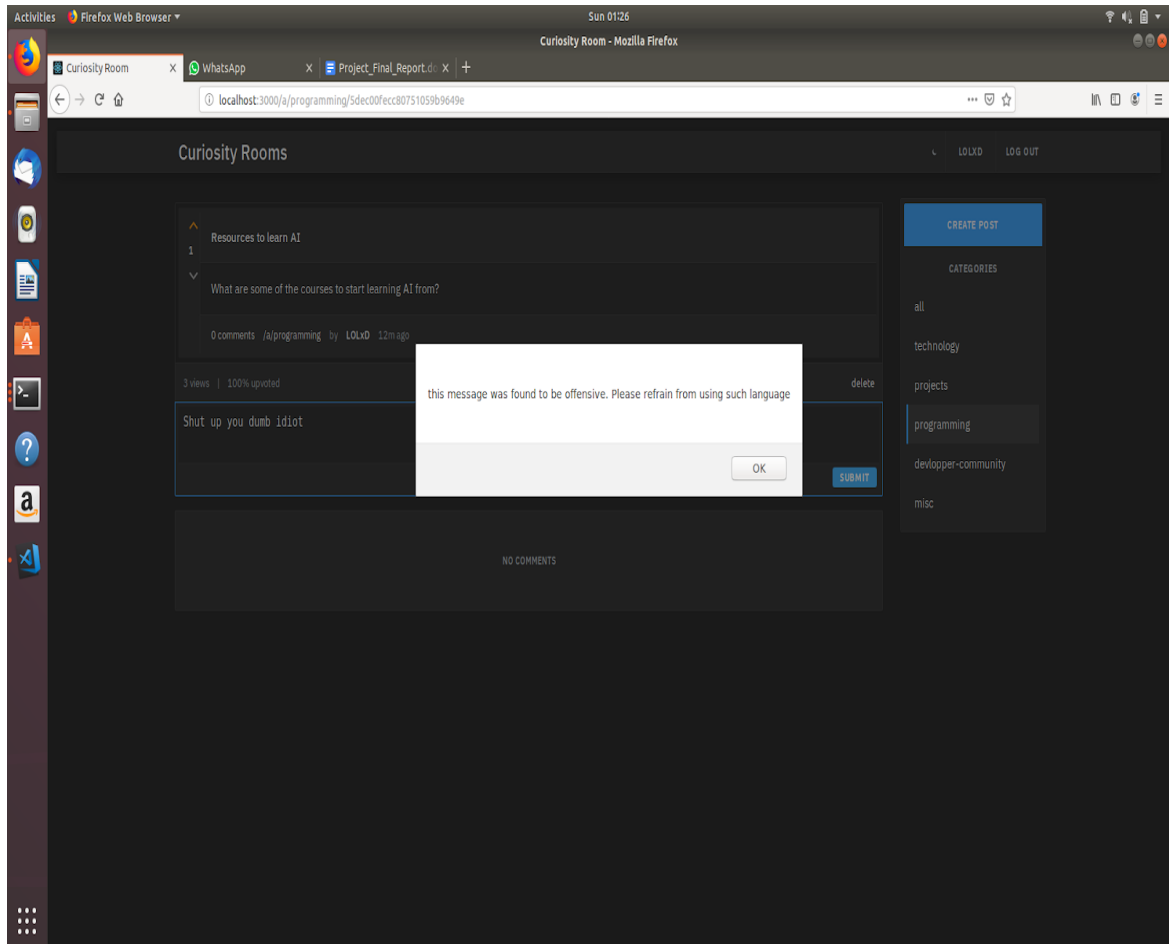Also if the user reports a comment , we may be able to use it to retrain our model.

Fig. 6.13 Comment Moderation

6.1.10 **Database:**

We use a No-sql database using MongoDB. We store our data in two files:

- **Posts:**

This stores the record of all the posts and we store the various attributes like title, category , text string, author ObjectID and created Date

	_id ObjectId	score Int32	views Int32	type String	title String	author ObjectId	
1	5dea1b31bbcefe1d50a2e009	0	4	"text"	"New Tech"	5dea16d3bbcefe1d50a2e008	"
2	5deb853d487ded1268144ed7	1	4	"text"	"Finding teammates for projec1	5dea01b1bbcefe1d50a2e006	"
3	5deb8596487ded1268144ed8	1	1	"text"	"Help me with Ruby on Rails!"	5dea0187bbcefe1d50a2e005	"
4	5deb8646487ded1268144ed9	1	1	"text"	"Cryptocurrency"	5deb5c30487ded1268144ed6	"
5	5deb868c487ded1268144eda	3	22	"text"	"Deep Learning Books you shoul	5deb5c30487ded1268144ed6	"
6	5deb8746487ded1268144edb	3	5	"text"	"Python's Advantages and Disa	5dea0187bbcefe1d50a2e005	"

		category String	text String	votes Array	comments Array	created Date	__v Int32
1	008	"technology"	"React is the Beast"	[] 2 elements	[] 0 elements	2019-12-06T14:41:13.937+05:30	2
2	006	"projects"	"Started a project based on B.	[] 1 elements	[] 1 elements	2019-12-07T16:25:57.072+05:30	2
3	005	"programming"	"I am thinking about starting	[] 1 elements	[] 0 elements	2019-12-07T16:27:26.952+05:30	1
4	ed6	"misc"	"Is there future in Bitcoin ar	[] 1 elements	[] 0 elements	2019-12-07T16:30:22.875+05:30	1
5	ed6	"devlopper-community"	"With the rise of machine lear	[] 3 elements	[] 0 elements	2019-12-07T16:31:32.745+05:30	3
6	005	"programming"	"Python's been gaining popular	[] 3 elements	[] 0 elements	2019-12-07T16:34:38.001+05:30	3

Fig 6.14 Posts Data

- **Users:**

This stores the information about the user such as username , password and user ID

	_id ObjectId	username String	password String	__v Int32	
1	5dea0187bbcefe1d50a2e005	"kheti"	"$2a$10$9j9hg4ahoVQbZE06fB9iF(	0	
2	5dea01b1bbcefe1d50a2e006	"paras"	"$2a$10$oZBwObDQAEIQ8fJ8TSilY(	0	
3	5dea16d3bbcefe1d50a2e008	"ashlin"	"$2a$10$0MEDak3u3EbP4E.B6lUin(	0	
4	5deb5c30487ded1268144ed6	"new_user"	"$2a$10$wBfxGrHJSrOUTf20dBPOa(	0	

Fig 6.15 Users Data