ShArena Project Report

About ShArena:

ShArena is an online Real time Team collaborating platform where each user has his own work space for each project, can interact with his team mates through a chat box and can view their work on a real time text editor

Why ShArena:

1)Real Time Collaborations is of utmost importance Now-A-Days with enhancing world and improving technology
2)Code sharing and getting doubts resolved on a piece of code is mostly done using mails which is very inefficient
3)Estimating the progress of your teammates and guiding them is very useful to support mutual growth
4)Collecting all the code at one place after working on different features is difficult and needs some easier and efficient way
5)Our main motivation came through this site but anyways it is a paid site but ours is not :p, please watch the video in the site to understand the motto of our project.

Key Features:

1)Friends and Groups
2)Chat-Box
3)Project Requests
4)Live Updating
5)Merge Feature

Features that are planned to be implemented:-

1)Friends and Groups for doing projects
2)Simple chat application
3)Live Editing
4) Special Feature named 'Merge Feature’

Features Actually Implemented:-

1)Friends and Groups for doing projects
2)Simple chatting application between 2 users with showing whether user is online/offline and typing/not typing and also a group thread for posting any updates ..
3)User ratings to rate friends based on their work.
4)Project requests based on the description of the project
5)Live Editing
6)Merge Feature with tag ‘@merge from_User title’ in the code and also with view/accept/delete features
7)Profile page to estimate the professional esteem of a user

Work allocation and Timeline

• 1ST WEEK:-
We had Completed basic structure of the project including Home page, creating accounts, login pages.
• 2ND WEEK:-
We have done friend-requests, groups Formation, Chat feature, group thread, animation for the Homepage.
• 3RD WEEK:-
We had Completed project-requests, Merge, Live Updating features by distributing work load equally among us.
• 4TH WEEK:-
We had Worked on the front-end(CSS of the HTML pages), User Rating, Profile Page , Resolving Bugs and Documenting Code.

A tour on our project(For better view with images refer to presentation):

Website page:

This is our website from where you can login or register yourself to our network

Home page:

It shows the username, his rating, his profiles picture and has links to the Friends page, Groups page, Profile pic upload page(a double upload makes the new pic replace the old one), Logout page.There is also a default picture if user hasn't uploaded one.
View Animation in homepage

Friends page:

User can send friend requests and accept/decline requests from other users. He can chat with users by clicking on their name in friends list and can also view their profile by clicking on their name elsewhere.

Chatting with other users:

Here you can chat with your friends. It is a real time chat showing whether the user is online or not and also notifies you when the other user istyping.

Profile page:

Here you can see other Users' Details, their rating based on review of his friends, projects he is involved in.

Rating page:

Here you can rate your friends based on the projects they are involved and their professional esteem.
Also rating has been done interactive through animation(Moving the cursor over the stars highlights the number of stars you have currently crossed)

View Animation in Rating page

Groups page:

i)Project list:

User can see the list of total projects he has been involved in.He can open the project to view his files, also others' and can also delete the project(don't worry a prompt will be popped up before deleting). He can also create a new project from the "+NewProject" link with a unique title and add members to it

ii)Send requests:

Here the user can see all the projects in the Data-Base that he is not involved in.Based on the description given he could match his interests and can send join-request to all the project members.

iii)Accept requests:

User can accept the join-requests of the other users for the projects he is involved in.He can view their profile and based on rating and other measures he can accept/decline him.

iv)Requests sent:

User can also see the list of all unaccepted requests.

Project open view:

i)Files uploaded:

Here the user can see the files he has uploaded in the project. He can either open or delete them. He can add new files by clicking on +Newfile link. He can also see the updates posted on the project by clicking on GroupThread link

ii)Un added friends list:

Here he can see the list of friends who are not in the project so that you can add them by clicking on ‘Add to this Group’ link

iii)Members list:

It shows the list of members associated with the project and you can view their files by clicking on ‘View files’ link

Group thread:

This is a group thread where everyone of same project can interact with each other,suggest changes to others’ work, help them in sorting out their problems and also can see the updates on the project if any.
You can add a post by clicking on the ‘+NewPost link above in the nav-bar.

Open file view:

This view is generated when you open YOUR file in the project. Programmers do make mistakes, so there are undo and redo buttons to our rescue. A searchable chat box makes it easy to contact your teammates whenever in need. Clicking on a name in chat box opens a chat window for both the users very similar to that of facebook.

Merge Feature:
Imagine you are working on a large project and your team mates decide to work on different features.
The main problem arises when you have to merge all the code. But don't worry ShArena is coming to your rescue. With the Merge feature code sharing has become lot easier!
Here there will be a person who decides to collect all the code.All the contributors work separately and when it comes to merging them, all the contributors send merge requests to the collector and the collector places @merge 'User_name' at the places where the code of a specific user has to come and accepts his merge request. That's it!! It's that simple!!! Not only this, there are lot more uses of this feature, this is just an example
P.S: When you fill a merge form with specific file of a user as destination a merge request will be sent to him with entire matter in the file from which you have sent him a request

View Merge Feature in action

Others' view page:
This view is generated when you open OTHERS' file. This view also supports live updating i.e, when the owner of this file is editing it right now, the changes can be seen live.

View Live update in action

Algorithms and Implementations:

1)Live Updating:

We are auto saving the file every time the user press a key. And for the user who wants to see the live changes, we are recursively sending the Ajax post requests to the django view and retrieve the saved data

2)Project Requests:

When any user sends join-request to a project we have saved a object with 2 fields(user and project) he has send request to.Then while rendering the projects page we extract the requests sent, requests he can accept and so on by using projectrequest's objects and project object's(by filtering the objects based of presence of user in the projectrequest and project model objects)

3)Friends and Groups:

We have implemented this feature by sending the ajax post request containing the data, from_user and to_user to django view which creates a merge request model with that data.Then for merging with the tag "@merge user title" we used a simple python script with basic string replacement methods

4)Chat-Box:

We have implemented this feature by using inbuilt Django-private-chat package where Dialog objects are created which are one-to-one relatioship on user and when the user wants to chat with the other user we invoke their dialog object which consists of messages till now and when user sends the message to others' it gets stored in their Dialog.

5)Merge Feature:

We have implemented this feature by sending the ajax post request containing the data, from_user and to_user to django view which creates a merge request model with that data.Then for merging with the tag "@merge user title" we used a simple python script with basic string replacement methods

References:

1)https://stackoverflow.com/questions/22063748/django-get-returned-more-than-one-topic
2)https://stackoverflow.com/questions/38097719/how-to-load-static-files-in-python-django
3)https://stackoverflow.com/questions/2470760/django-charfield-with-fixed-length-how
4)https://stackoverflow.com/questions/15845116/django-how-to-set-min-length-for-models-textfield
5)https://stackoverflow.com/questions/26745283/how-do-i-import-filenotfounderror-from-python-3
6)https://stackoverflow.com/questions/3805958/how-to-delete-a-record-in-django-models
7)https://stackoverflow.com/questions/21367320/searching-for-equivalent-of-filenotfounderror-in-python-2
8)https://stackoverflow.com/questions/14456503/how-to-get-a-particular-attribute-from-queryset-in-django-in-view
9)For Ajax reference - https://www.w3schools.com/xml/ajax_intro.asp
10)Tutorial for getting basic idea on how django works -
https://djangoforbeginners.com/
11)Djnago-friendship refernce -
https://pypi.python.org/pypi/django-friendship/
12)Django-private-chat reference -
https://github.com/Bearle/django-private-chat/blob/dev/README.rst
13)Working with ajax in django -
https://docs.djangoproject.com/en/1.11/ref/csrf/
14)For adding undo and redo buttons - https://github.com/jzaefferer/undo
15)For adding 5-star rating animantion -
https://www.cssscript.com/five-star-rating-system-with-pure-css-and-radio-button-hack/
16)Creating search list -
https://www.w3schools.com/howto/howto_js_filter_lists.asp

17)Allowing users to see pages only if he logged in -
https://stackoverflow.com/questions/24619629/django-do-not-allow-users-to-see-pages-when-they-are-not-logged-in
18)Dealing with manytomanyfield in a model -
https://docs.djangoproject.com/en/1.11/topics/db/examples/many_to_many/
19)Dealing with filefield in models -
https://simpleisbetterthancomplex.com/tutorial/2016/08/01/how-to-upload-files-with-django.html
20)Dealing with forms django -
https://docs.djangoproject.com/en/1.11/topics/forms/
21)Dealing with models django -
https://docs.djangoproject.com/en/1.11/topics/db/models/

Conclusion:

We have learned how to face real life situations where we need to google-out the things and find a way out for the problems we are facing, importance of GIT(version control) when we are working in groups to do bigger projects, handling errors, acquired team collaborating skills.