

**Lovely Professional University**

**INT213**

**PROJECT**

***ON***

**SOCIAL NETWORKING SITE**

Submitted to**: Sagar Pande**

Submission**: Nov, 2020**

Group Members**: Ravi Ranjan(roll no – 15)**

**Ashutosh Tripathi (roll no – 69)**

**Britipta Giri (roll no – 35)**

**BTech (CSE)**

**Section – K19HZ**

**ACKNOWLEDGEMENT**



Firstly we would like to express our special thanks to our teacher **Mr. Sagar Pande,** who has given us this opportunity to do this exciting and helpful project in python. Since this is our very first Project, we have given our best and while doing this project, we came across a lot new things and techniques that helped us to enhance our knowledge for further any Projects.

We are also thankful to our friends who have helped us in making this project throughout and innovative discussions with them.

We are truly making this project for our knowledge enhancement and not just for marks.

Thanks again to all who helped us.

Ravi Ranjan Ashutosh Tripathi Britipta Giri

BTech CSE BTech CSE BTech CSE

**Summary of the Project**

****

* Our project is on the topic social networking site.
* This project is very close to the social networking site **Twitter.**
* The name of this site is happytweet
* It allows a user to create an account, login through an existing account and logout aswell.
* It basically allows a user to post feeds.
* It can be very beneficial for sharing news and happening around him/her.
* It can also helpful in connecting with new people around the world.
* This site available with the features like posting news or others, liking system on the post, allows a user to comment, follow other interesting users.
* It also allows to delete unwanted posts or update the existing post
* It allows a user to update hi/her profile and can set pictures to his/her profile.

**PROJECT OBJECTIVES AND RESEARCH**

The very first thing as a team we figured out the basic needs of a social networking site that is connecting with the people around the people around the world and share thoughts and happening around them with others. So the best suited site for us that is easy to built is to make a site similar to Twitter.

This Twitter like networking site which we named as happytweet is built in the Django framework with certain libraries like crispy\_forms, sql, numpy and many more.

Django have many modules like Django.conf.urls, Django.contrib.admin and many more

The main objective of this project to established a connection between the two users through their thoughts and thinking and popularity. It can also be used to share news and aware others

What is happening around.

 **PROJECT DESCRIPTION**

* We used Django frame work for the creation of this site. Django is a high-level Python framework that encourage rapid development and clean pragmatic design. Django is basically work on backend of an app
* For frontend we use html and css. Bootstrap is mainly used for better result and to save time as well.
* In he very first page of this site that is log in page options like login, create an user, home, about buttons are available along with a form for login that take input user’s name and password and with correct credentials it leads a user to homepage.
* The first page also provides an option for creating an account. It also comes with a form that accepts user’s name, user’s email id, password and confirm password. It also contains a signup button which enables a user to create an account
* Home page consist of three columns. First column contains the user’s profile picture, user’s email id, his/her number of followers and following along with a tweet button option.
* Second column consist of post made by the user only. Post is shown along with the like button, comment button, delete and update option
* Third column allows a user to search any possible user who may have an account. Third column also show the name of the users who has posted.
* The top row of the home page consists of home button, about button, create a tweet, setting and logout button.
* Logout button leads a user to login page.
* Setting buttons lead to the user’s profile page which consist a form. The form is for updating the profile picture of the user, changing his/her name, or email id along with a submit button
* For the admin side of this site, all the data are stored using sqlite3.
* We can access the database only using super user’s profile.

**IMPLEMENTATION**

* We first created a project using the command Django-admin startproject project\_name.
* This allows a basic functioning of a any project. The main things that a project provide is setting.py file and urls.py file. setting.py file is used to connect the application further made inside the project. It also joins the base directory path of the other files present inside the apps like staticfiles, media, bootstrap4 and so on
* Urls.py file of a project used to connect with the urls.py of the created apps inside the project. further urls,py file of the apps joins with the views.py files of the pps and further views.py file is get associated with the frontend code that is basically written inside html or css or bootstrap.
* An app is created inside a project using the command Django-admin startapp app\_name.
* The app is basically created for the different functionality inside a project. each app contains its own urls.py and views.py that is further used to create site.
* For creating project and apps, basically command prompt or powershell in windows operating system is used.
* For example,
* step-1 -: django-admin startproject django\_project
* step-2 -: django-admin startapp blog
* step-3 -: write the function name available in the app in the setting.py file of the project under the block installed apps.
* Lets say the function inside the apps.py file of the app blog is BlogConfig. Then in the settings.py write as ‘blog.apps.BlogConfig’,
* And the url path is created between the urls.py file of the project an urls.py file of an app.
* For example - urlpatterns = [
* path('admin/', admin.site.urls), ] this url link to the admin path of the project.
* Admin site is nothing but the database where all the informations are stored.
* Admin site can only be accessed by a super user. Super is not created through the site but in the command prompt or powershell using the command python manage.py createsuperuser.
* After this command user is asked to enter his/her username email id and set password.
* Admin has the access to all the users who has created an account on that particular site
* There are many modules used in making of a social networking app, like
* Django.apps
* Dajngo.contrib
* Django.urls
* Django.contrib.admin -django’s admin site
* Django.conf.urls
* Django.contrib.auth – django’s authentication framework
* Django.contrib.auth.forms
* Django.db
* Django.db.models
* Django.utils.timezone
* Django.test
* Django.view
* Django.shortcuts
* Django.view.generic
* Django.contrib,auth.decorators
* Django.core.wsgi

In our project the first app is blog. The blog is going contain everything about tweets updates comment and everything we see on tweets and blogging. The we have users. The users app contains everything about users login authentication system reset password and everything about user logout.

we need to specify static and media paths. We will add the following lines to our settings.py file to do so

STATIC\_ROOT = os.path.join(BASE\_DIR, 'staticfiles')  
STATIC\_URL = '/static/'MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')  
MEDIA\_URL = '/media/'

add some more settings to help us tell Django what to do in those particular situations.

Now we defined our models file that is mode.py file by creating the models first. It is defined in the Django to tell Django what to save in the database and to define the relationship between different models and what characteristics they have

.****Now we will define a views.py file. it would contain all our views (how to render the files in the web browser). It directly passes data to the templates. Views.py file is too large. Here is some examples made in views.py file.

class PostListView(LoginRequiredMixin, ListView): - for viewing the post

class UserPostListView(LoginRequiredMixin, ListView): - for seeing the self post

class PostDetailView(DetailView):- for deleting post

class PostCreateView(LoginRequiredMixin, CreateView):- for creating post

class FollowsListView(ListView):- shows the followers

the file urls.py contain all the URLs for the website. It has an include which include all the URLs from the apps made in a project.

Now we focused on all the templates for displaying the logic we wrote in the user app. User app deals with all the authentication, profile, followers, search, and such other things.

Html file are created inside the templates for views the webpages

* Login.html
* Logout.html
* Password\_reset.html
* Password\_reset\_done.html
* Profile.html
* Register.html
* Serach\_result.html

In blog app templates are made to work on the pages of-:

* About.html
* Base.html
* Follow.html
* Post\_delete.html
* Post\_detail.html
* Post\_new.html
* User\_post.html

 After creating all these file and correctly rendering migration of database is made and then sync the database.

Our project has an SQLite3 database built-in which we will be using for development and testing.

So for migrations we ca run the following commands python manage.py makemigrations and  
python manage.py migrate

This command migrates all the files, users post along with the users detail of account creation to the database that can be views on the admin page.

The admin page can only be visited by a superuser. To create a superuser, we can run the command- python manage.py createsuperuser

Superuser will have access to the admin panel and can edit the database element. Superuser can also delete a user.

After all the implementations a server is automatically created. To run the surver command – python manage.py runserver is used which create a local address to the computer. The address the can be copied to the search engine available in the computer which will leads to the login page of our site

**Description of webpages**

The first page we land after visiting the server is the login page. Login page is contained with a form which allows a user to enter name and password. This form is created using crispy forms. When the credentials are correct user are allowed to navigate to homepage.

Login page also contains buttons like home login and register at the top of the page. Register button also contain a form which is used to create a new account for a new user. Register form take inputs of the username, email id, password and confirm password. After creating an account a message is displayed on the login page that an account is created successfully.

The second page is the homepage which at the top row contains button like Home, Create a Tweet Settings and Logout.

Logout button take the user from homepage to login page

Setting button navigate to page where a form is present. This form is used to change users name or email or change profile picture.

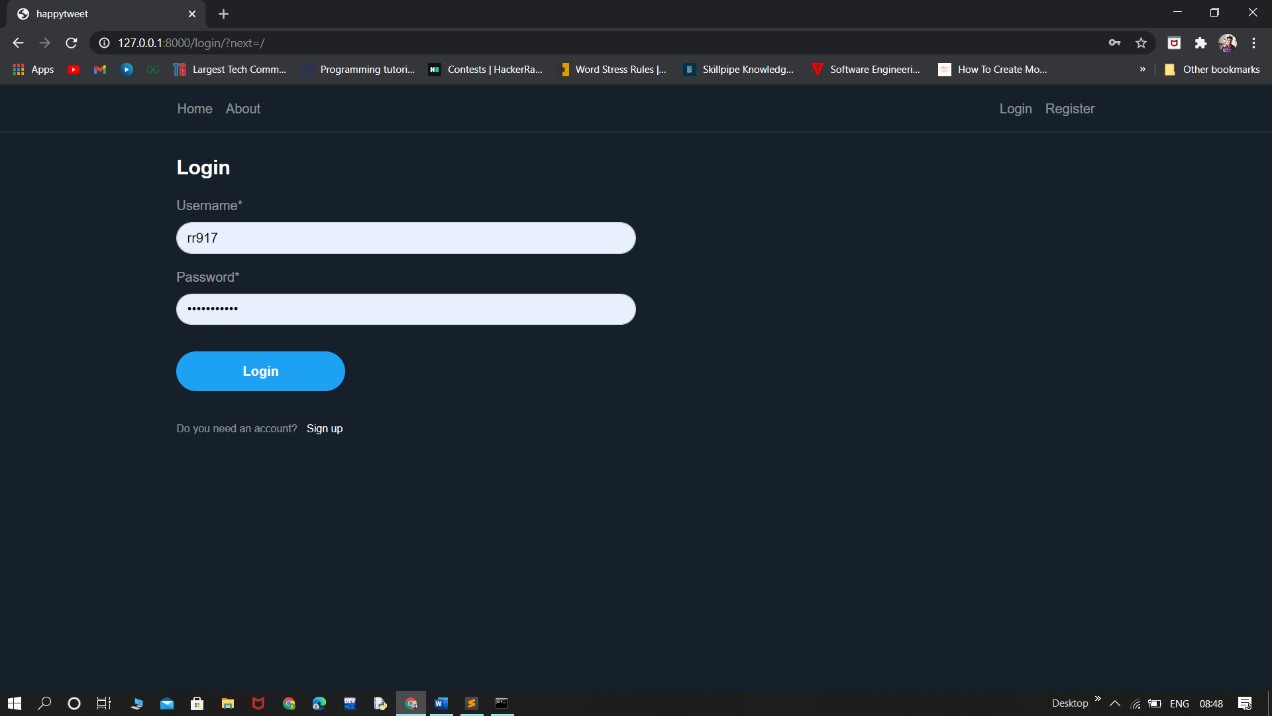
Below the navigation tab the name of the site that is Happytweet is shown. homepage is divided into three columns.

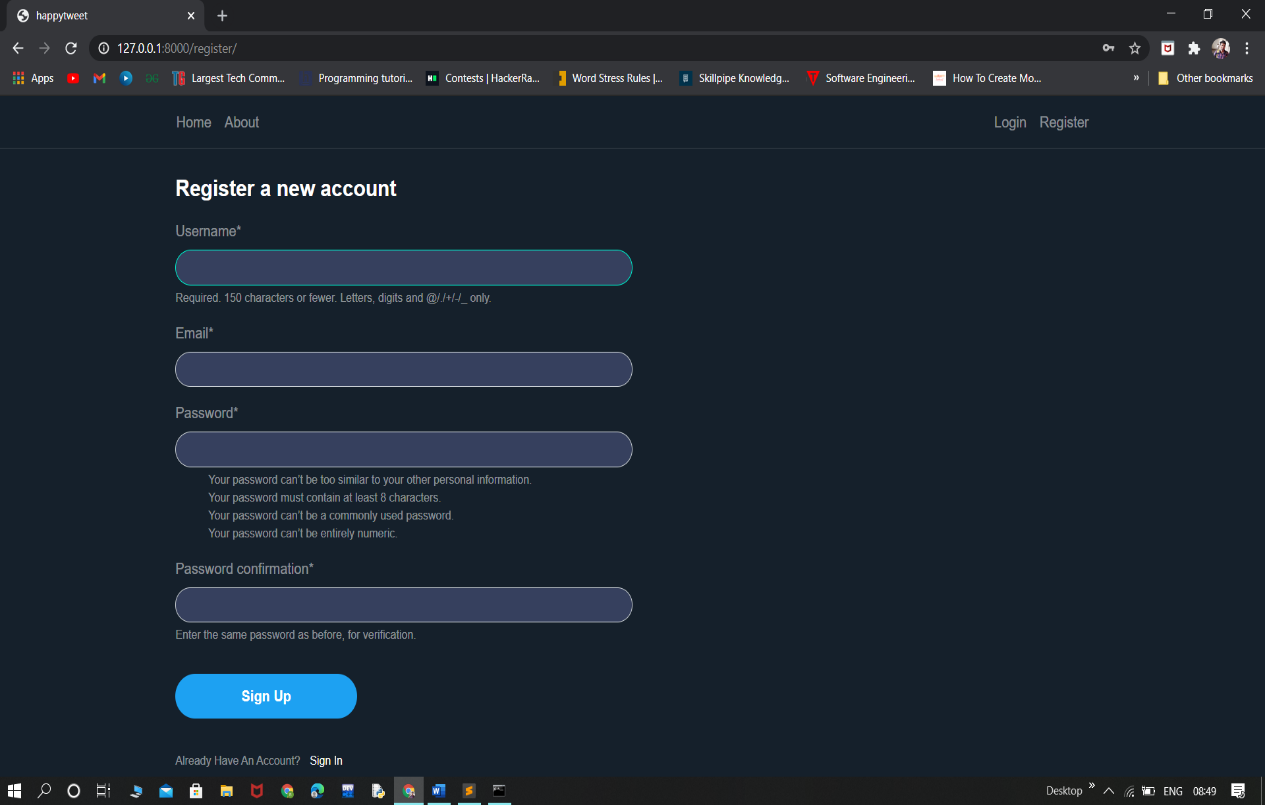
* The first column shows the site name, profile of the user, his number of followers and following and a tweet button. Tweet button navigate to a form where we can write anything and post it using submit button
* Profile of the user shows the name of the user and gmail id and a picture of the user if uploaded
* The second column consists all the posts that are posted by any users. Each post contains the name of the use who has posted. Each post come with a feature of liking it and also comment feature on which other users can comment
* each post also shows the time, date and day of the post
* A user can also delete hi/her post or can update post
* The post column also shows the number of pages at the bottom with next and last button. Next button navigates to the pages of more recent posts while last button navigates to the pages of older posts.
* The third column is comprise of search twitter where an existing user can be searched using his/her name or email id.
* The row below the search twitter is available with a block that shows an option Who to follow and shows the list of the user who has posted anything.
* We can also go to our friend’s profile by clicking on their profile available in the list.
* We can also see the people who is following us on going to the follower button in the first column and we can also see the people whom we are following by going on following button.
* The count of followers and following are also shown.

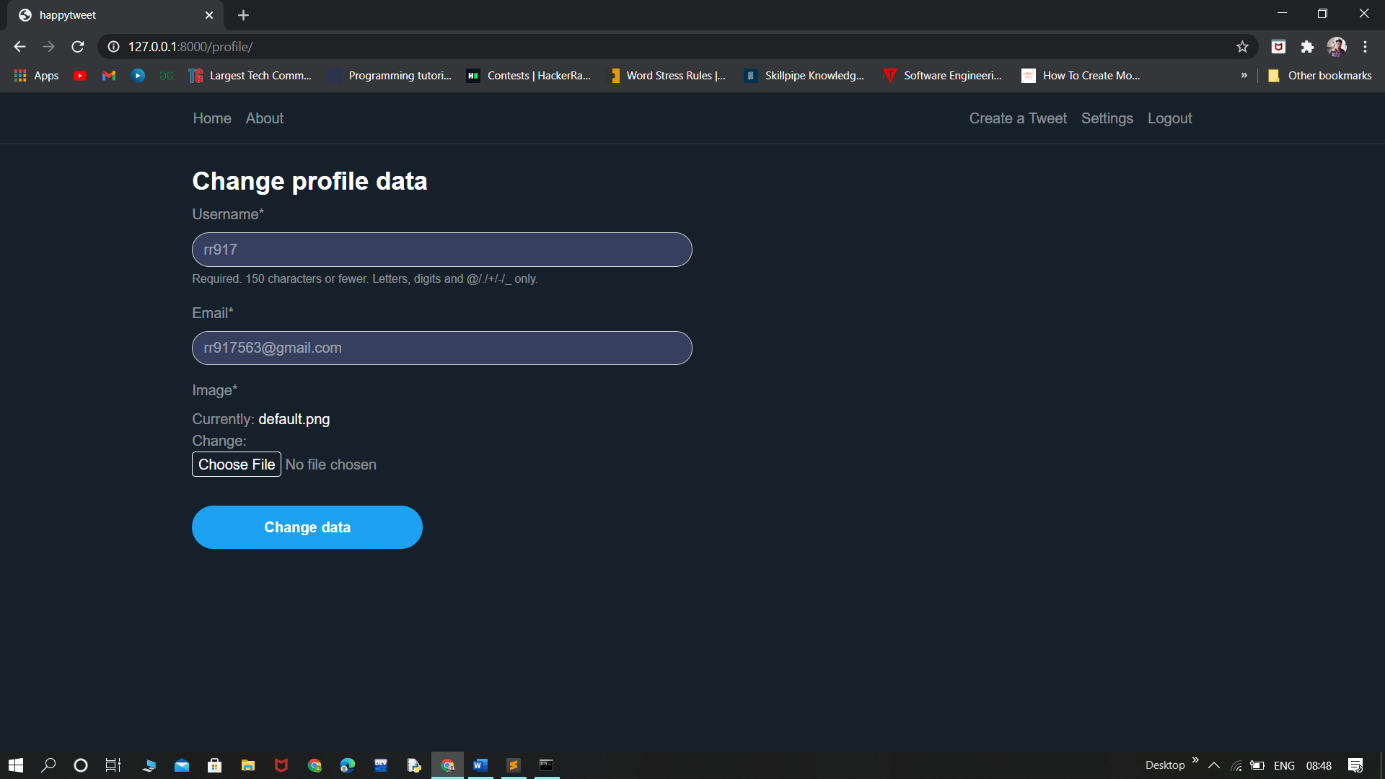
**Project Contribution**

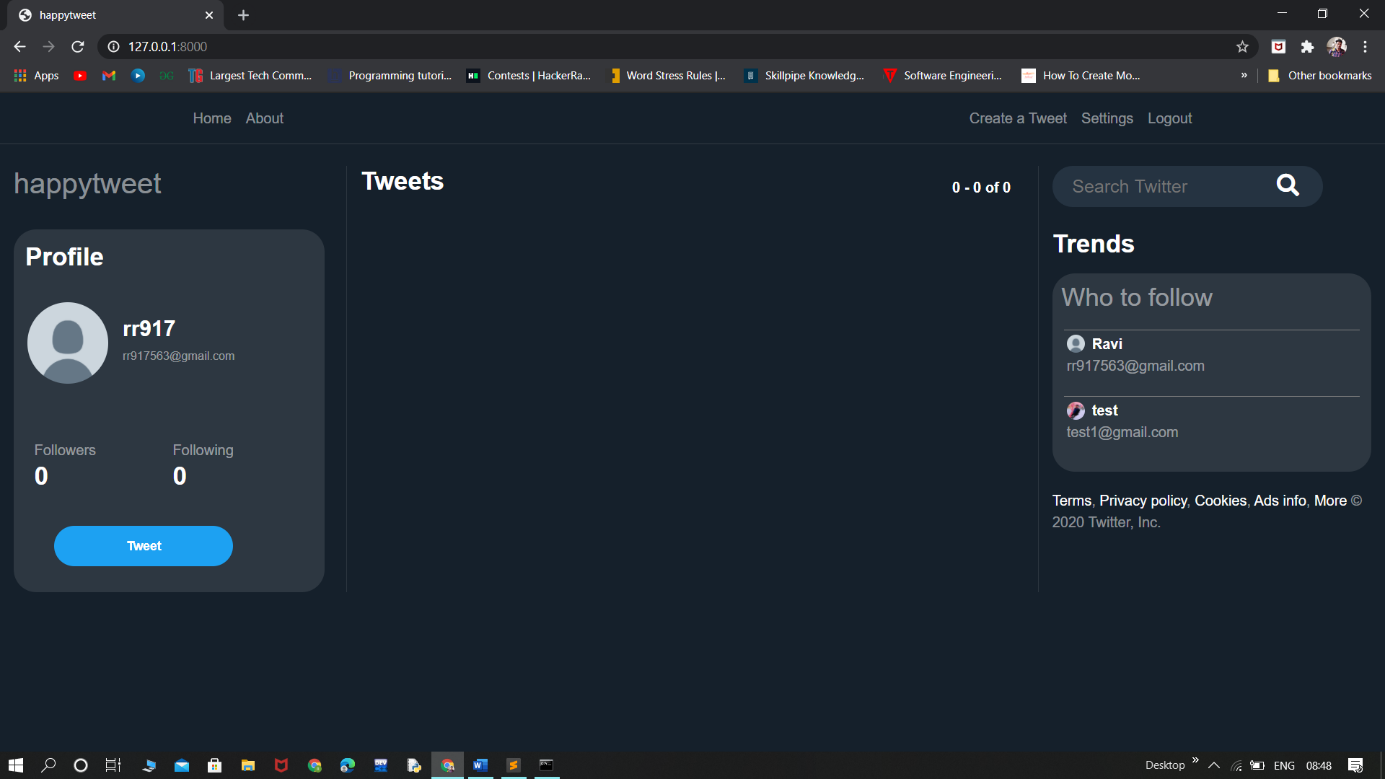
* We all three members have tried to give our best in this project
* We all have equally contributed in the project
* All the Discussions and implementation and modifications are done together
* We have equally shared our ideas and sources or anything related to this project
* Ravi Ranjan (roll no -15)- has done the registration part, login part along with the webpage designing and also merged all the codes together done by all three
* Ashutosh Tripathi (roll no -69) and Britipta Giri (roll no- 35)- have done on the homepage.
* Others functionalities like liking on post, commenting on post are done by Ravi Ranjan with the help of google and you tube.
* The setting part where a user can update pictures is done by Ravi.
* All the work related to this Project either Code Writing or Report Making or Presentation, we all have contributed almost equally from our side.
* We all are equally responsible for the pros and cons of this Project as we have given the equal efforts

**SCREENSHOTS OF THE WORKING PROJECT**

****

****





**THANK YOU**