Cloud Computing
Assignment 2(Instagram)
By-Ankit Gour
MSc-Big Data and analysis (September 2019-2020)
Student ID-2982713

Table of Contents

1. About the Assignment

This application is based on Instagram. This application is a replica of Instagram and it consist of all the basic functionality which Instagram is already having. In this application I have made few changes in UI. This application will have user to access all the function. This application is build using python, html, CSS and java script. This application is also using google app engine to save data on a no SQL database.

This application was built in seven steps from logging to comments and logout.

2. Brackets

1) Bracket 1:

- In this Phase of development I have created a login logout page for the user.
- The data model used here is MyUser using ndb module.
- All the data is saved on the localhost and can be retrieved from there.
- In this data model the key has been set as user email.
- The data model MyUser have 3 attribute that is email_address, following, follower.
- The following and follower will contain the list of other users email.
- The following attribute will contain the list of user email who are been followed by that user.
- And the follower will contain list of users email who are following that user.
- All the data types used in this is String Type.

2) Bracket 2:

- In this bracket we have added a post.
- Added an image to a blobstore.
- Have restricted user to add the image in jpg and png only.
- Have added a section to add image caption/text.
- At the time of addition of image, in the post a key is generated, by the user which has entered a user key.
- The data module post has caption, image, image URL and image id.
- Every field has a StringProperty except image which is a blobKeyProperty.
- For addition of image in blobstorage we have created another py file uploadhandler.py in which we have imported blobstore, ndb, blobstore_handlers, BlobKey, get_serving_url and images with datetime properties.

3) Bracket 3:

- In this bracket we have created a home page.
- In which the user in which he has added all his posts along with the count of the following and the followers.
- This is a detailed view of the current users following list and its followers list as well.
- Along with this we have given a search button in which current user can search different users.
- When he will search a user he will find a list of list of users in the Instagram app.
- Here in this list if we click on any users profile, the homepage of that user will be opened in a new window.
- On that window, the posts of that user will be seen,
- It will also have its followers and following.

- If the current user is not following the searched user a follow button will be made visible else a following button will be seen on its page.

4) Bracket 4:

- In this bracket, if the follow and unfollow button, will be taken into account.
- Here if the following button is clicked it will start unfollowing the other user
- Else when the follow button is clicked it will following the other user.
- Here, when the follow button is clicked his email id is saved in the followers list of that user.
- And in that user's following list, the email id of the current user is saved in its followers list.
- When the following button is clicked the current user's email id is removed from other user's follower list.
- And other user's email id is removed from the current user's following list.
- In this phase of development we also create user a functionality where he will be able to see all the current 50 post from all the user on the main page .
- All the post on the main. Page will be in the chronological order and in vertical timeline.
- User can only view recent 50 post from different user also from him.
- All the post are been fetch from the post datastore.

5) Bracket 5:

- In this phase of testing the user will be able to view all the people they are following and people who are following them.
- In this phase we will add another ability to the user in the home page .
- They can now clock now of people they are following/number of people following him.
- This will bring a new page in the application. Where we can see the list of users who are following the user and list of user email whom they are following.
- On the list page we can see the list of user email and we can now click on that user to find out more about the user like what they have posted and how many people they are following and by how many they are followed.
- This bracket will bring the image of Instagram picture sharing and the connection benefits.

6) Bracket 6:

- This phase of development brings the comments in the picture.
- A user can give a comment to a post as many times as he wish to do so.
- A comment can be given to a different users post which can be viewed to all the users added to that particular user.
- In this a the home page, at the bottom you can see a view comment button, which will redirect you to a new page.
- Here you can see all 5 comments from the post.
- And there will be a button comment, which will bring the ability to the user to comment on the post.

- The post comment is limited to 200 words as per the documentation.
- The entire comments will be saved another data store comment.
- In the datastore comment we will be storing the image URL, comment, user who has given the comments.
- After adding comment, you will be able to see that comment on the last page.
- All the comments, will be in chronological order.
- The comment will be shown as first the user who has given the comment and then the comment.

7) Bracket 7:

- In the view comments section, you will be able to see, only 5 recent comments.
- There will be a button, which will enable user to view more 5 recent comments, which will bring in total to the 10 recent comments.
- When the button is clicked it will expand the table in the timeline.

3. Python Files

1) Myuser.py:

- In this py, we are creating myuser class with ndb model.
- And in this class, we are creating 3 attributes email_address, following and follower.
- All the 3 attributes are having StringProperty.
- This data model indicates the Instagram user model.
- In this we are importing ndb from google.app.engine.

2) post.py:

- In this py we are creating Post class with ndb model.
- In this py we are importing ndb from google.app.engine
- And in this class we are creating 5 attributes which are caption, image URL, image, imageinfo and timepost.
- All the attributes are having String Property except image, which is having the blobkeyproperty.
- All the attributes in this class are in form of a list.
- So when the key is generated, it will create only single data entry, for single user.
- All the post by the user will be saved in 1 entity in the form of list.
- In the image URL, we are saving the URL link to hat image which can be used in terms of retrieval.
- Timepost is an attribute, in which we are saving the time of the post when it has been made.

3) Comnt.py:

- In this py we are creating a class comnt().
- In this class we are importing ndb model from google.app.engine.

- In this class we are creating 3 attributes which are image URL, comment, user.
- In this all the fields are string property, there is no user defined key generation in this ndb datastore.

4) UploadHandler.py:

- This py is for uploading the file, to the datastore In the form of blobkey.
- In this py we are creating a class called uploadHandler().
- In this we are importing blobstore, get serving URL, blobkey, blobstore handler from google app engine.
- In this we are creating a URL Link which will push our image into blobstore.
- At the end we are redirecting this class to the main class.

5) Main.py:

- In this py we are importing myusers, post, comnt, uploadHandler.
- We are also importing webapp2 and JINJA and ndb.
- In this first we are creating a JINJA environment, for our program.
- At the end, we are sending all the class to webapp2.

5.1. MainPage class:

- In this class, we will be creating login logout URL and we will displaying posts.
- In this class we are first creating a URL link for a user to login or logout.
- If a user is logged out, he will be seeing a page or a link to login.
- When a user is logged in, he will be able to see all the posts, he has created and the people he is following have created.
- In this there will be a navbar, which will be giving access to the their home page, to add a post, to search a user and to log out.
- The post which have been shown below will be in chronological order following vertical timeline.
- The total number of posts a user can see is limited to 50, which would be the recent 50 posts.

5.2. Postimg class:

- In this class, we will be calling an uploadHandler.py. to add a post.
- This class will permit user to add a post and save it into datastore.
- In this class we will be creating some template values, which will be passed with template in add.html page.

5.3. Disppostown class:

- In this class, is a link to home page, where user can see, all his posts.
- Here we will be creating, a page for a user so that he will be able to see, their posts, how many people they are following and people they are followers with.

5.4. Seruse class:

- This is the class which will enable the user to search a user in the entire myuser database.
- In this we will first query all the list of users, than filter them.
- In this a user will enter the email address and a loop will search that user in the entire myuser database.
- If a user is not found, it will show no user present.
- If user clicks on the search button without entering an email id it will give the list of users in the Instagram.
- When user sees the result, he clicks on the users email id and it goes directly to that users profile.

5.5. Prof class:

- In this we are creating a class which will show other user's profile.
- In this we are showing their posts made by that user and the number of people he is following and number of people following him.
- There is a button, follow and following which will bring the user's ability to follow or unfollow the user.
- In this class all the posts shown are in chronological order.

5.6. Flowuser class:

- When the follow button on other users profile is clicked, this class is triggered.
- In this class, we are adding user's email id into other users followers list.
- Along with this others user's email id will be added to their following list.
- In this we are creating an HTML page inside self.response.out.write().
- This HTML page, will bring users ability to go back when they have started following that user.

5.7. Unflowuser class:

- When following button on other users profile, is clicked this class is triggered.
- This this class we are removing users email id, from other users followers list.
- And other users email id will be removed from users following list.
- In this class we are creating a HTML page in self.response.out.write().
- This html page will bring users ability to go back when they have unfollowed that user.

5.8. Viewflo & viewflo1 class:

- These 2 classes does the same work to count the follower.
- In this class we will count how many people are following the user.
- And when it is triggered from other users profile then it will display how many people are following that user.

5.9. Viewing & viewing1 class:

- These 2 classes does the same work to count the number of following.
- In this class we will count how many people user is following .
- And when it is triggered from other users profile, it will count how many people that user is following.

5.10 Postcomnts & viewcoments class:

- These 2 classes play the same role.
- When this class is triggered, a user is able to see all the comments, of a particular post.
- In this class user is having ability to view recent 5 comments.
- The user has given one more ability that when a button is clicked for viewing more comments, he will be able to see 5 more recent comments that will bring the total to 10 comments.

5.11 Viewcoments class:

- In this class user will be able to add comment to the post.
- In this the user has given a text box which has a limit of 200 words.
- When a comment is added 3 things are saved at the same time.
- User email id, image URL and comment are the 3 things.

4. UI Decisions

- n this I have used bootstrap, for better UI outcomes.
- Also I have used, navbar to enhance the abilities of the UI.
- Using navbar has helped me order my pages in a definite order and manner.
- I have used the colouring properties and the html syntaxes in a manner that it satisfies the documented view of Instagram.
- The use of buttons and redirections which has helped me to develop a good UI.
- For images I have used cards, from bootstrap which helped in bringing the image view in a proper manner.