

# Cloud Computing: Project Report

## Django CMS: Image Database

By-

P.Srikanth-1401036

K.Dilip Kumar-1401027

Biswa Bushan Soren-1401015

G.Athul Sai Shravan-1401018

### Goals:

To develop a content management system (CMS) using Django framework where the users should be able to:

- ❖ View all images.
- ❖ Insert new images and associated meta data (tags).
- ❖ Search for images that relate to certain meta data.
- ❖ Delete images.
- ❖ Perform the above listed operations in a Web interface.
- ❖ Keep images in persistent storage, allowing shutdown and restart of the service.

### Procedure:

Start the server using the command: `python manage.py runserver`

Then enter the URL in the browser: `127.0.0.1:8000/images`

#### Instructions:

##### What is Mysite?

Mysite is an app that allows you to upload and store all of your images on the cloud.

##### How do I add image?

First, create a new album by filling out the form on this page. Once an album is created you will be able to add/upload images.

##### What are some Album logo best practices?

- Have a resolution of 512x512
- Use common image formats such as .JPG, .JPEG, or .PNG
- Remain under the 2MB limit.
- Square images look best

#### Project Done By:

- P.Srikanth-1401036
- Biswa-1401015
- Athul-1401018
- Dileep-1401027

❖ Log in:

- New users need to be registered enrolling their username, email and password.
- The users should log in with their username and password to authenticate themselves and enter their CMS page.
- If either the username or password are invalid then it will throw an invalid login message.

❖ Home page:

- The users are provided with the choice to add albums and uploaded images. He can also view the uploaded images.
- The user can also run an image or album search using meta data tags.

❖ Adding album:

- The user can create an album with the attributes following artist, album title, genre and album logo.

Django Code:

```
class Album(models.Model):
    user = models.ForeignKey(User, default=1)
    artist = models.CharField(max_length=250)
    album_title = models.CharField(max_length=500)
    genre = models.CharField(max_length=100)
    album_logo = models.FileField(upload_to='media')
    is_favorite = models.BooleanField(default=False)
    def __str__(self):
        return self.album_title + ' - ' + self.artist
```

❖ Adding image:

- The user can add images by entering the image title which becomes the meta tag for that image.
- He can upload the file from the system directories using the browse option.

Django Code:

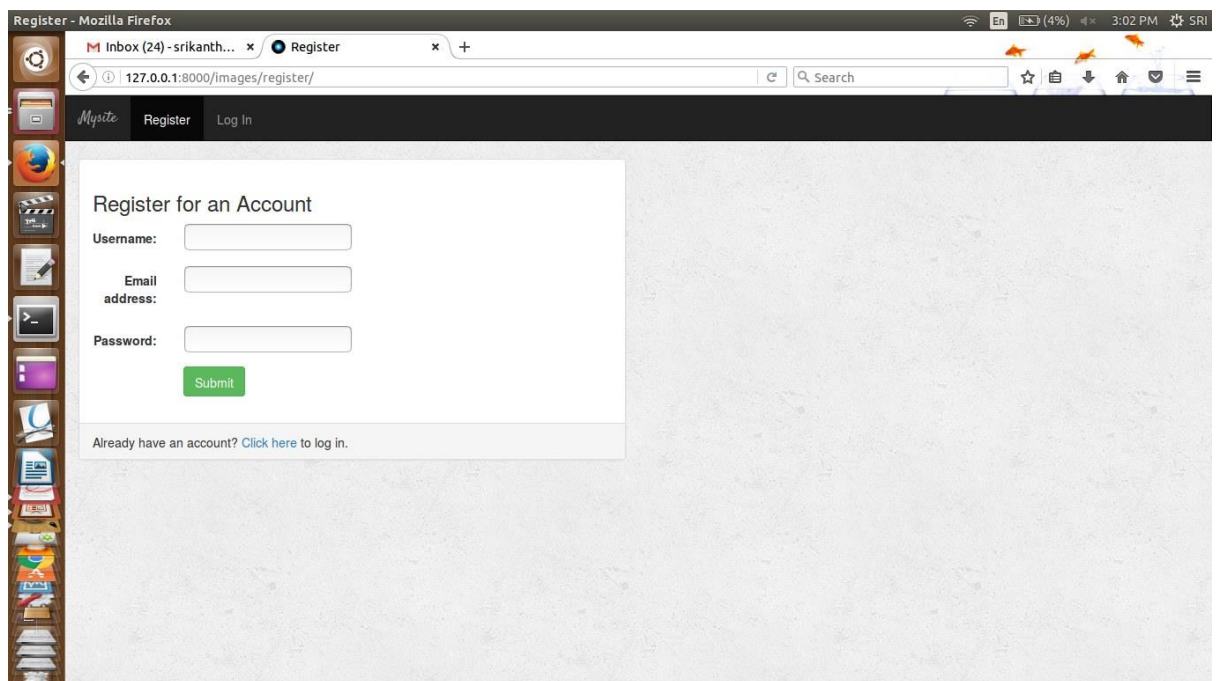
```
class Images(models.Model):
    album =
models.ForeignKey(Album, related_name='image_set', on_delete=models.CASCADE)
    image_title = models.CharField(max_length=250)
    image_file = models.FileField(upload_to='media')
    is_favorite = models.BooleanField(default=False)
```

```
def __str__(self):  
    return self.image_title
```

- ❖ Search image:
  - The user can search for the images and albums using the meta data provided during their creation.
- ❖ Delete image:
  - The user can delete a particular album or a particular image from the database providing the album title or image title respectively.

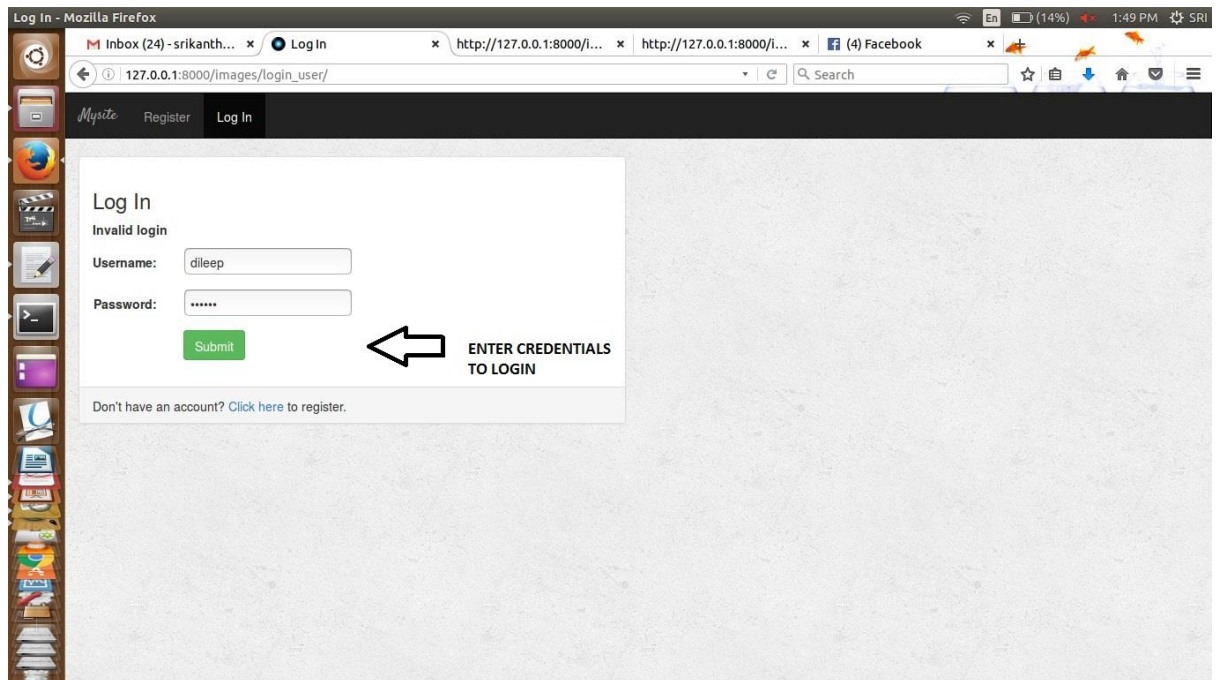
## Input and Output:

STEP-1: The user registration.

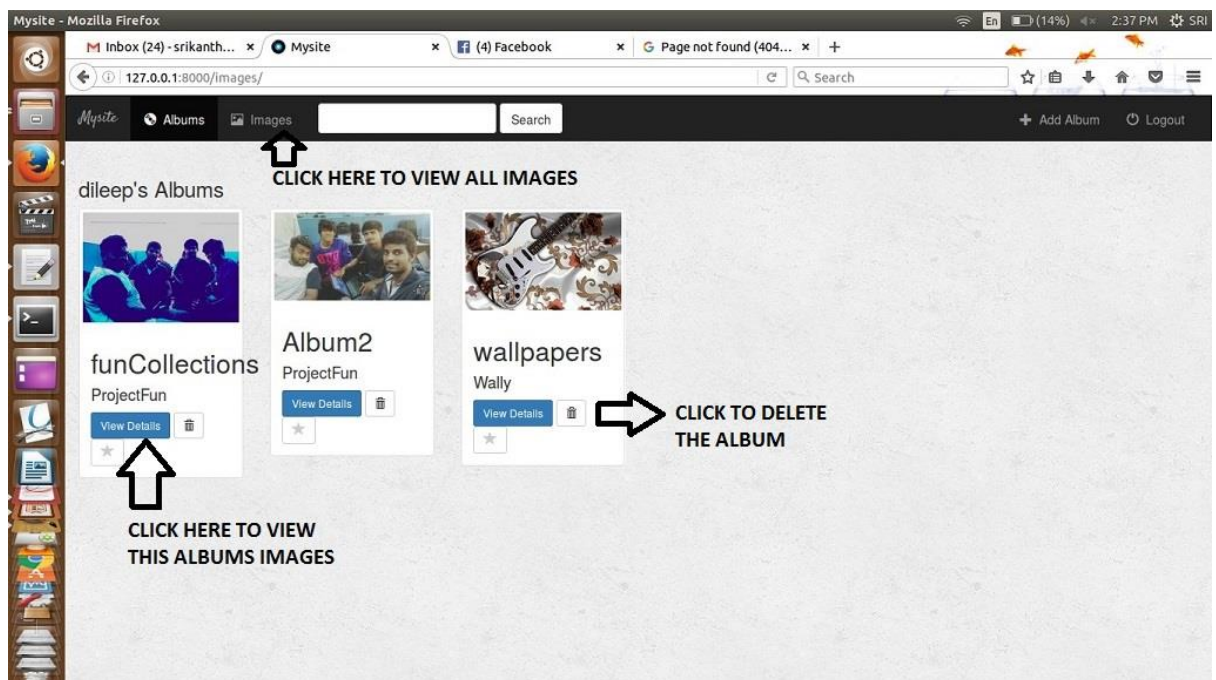


The screenshot shows a web browser window titled "Register - Mozilla Firefox". The address bar displays "127.0.0.1:8000/images/register/". The page has a dark header with "MySite" and navigation links "Register" and "Log In". The main content area features a "Register for an Account" form with three input fields: "Username:", "Email address:", and "Password:". A green "Submit" button is located below the "Password:" field. At the bottom of the form, there is a link: "Already have an account? [Click here](#) to log in."

STEP-2: The user enters his log in credentials.

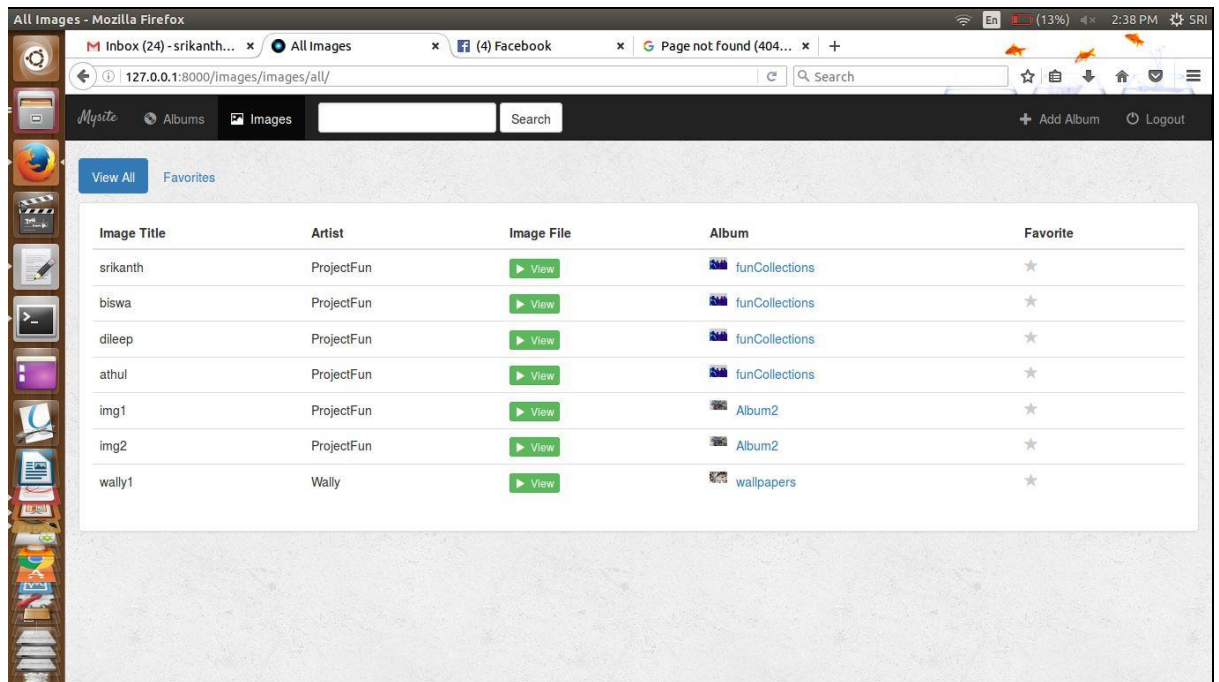


STEP-3: The homepage of the user appears.

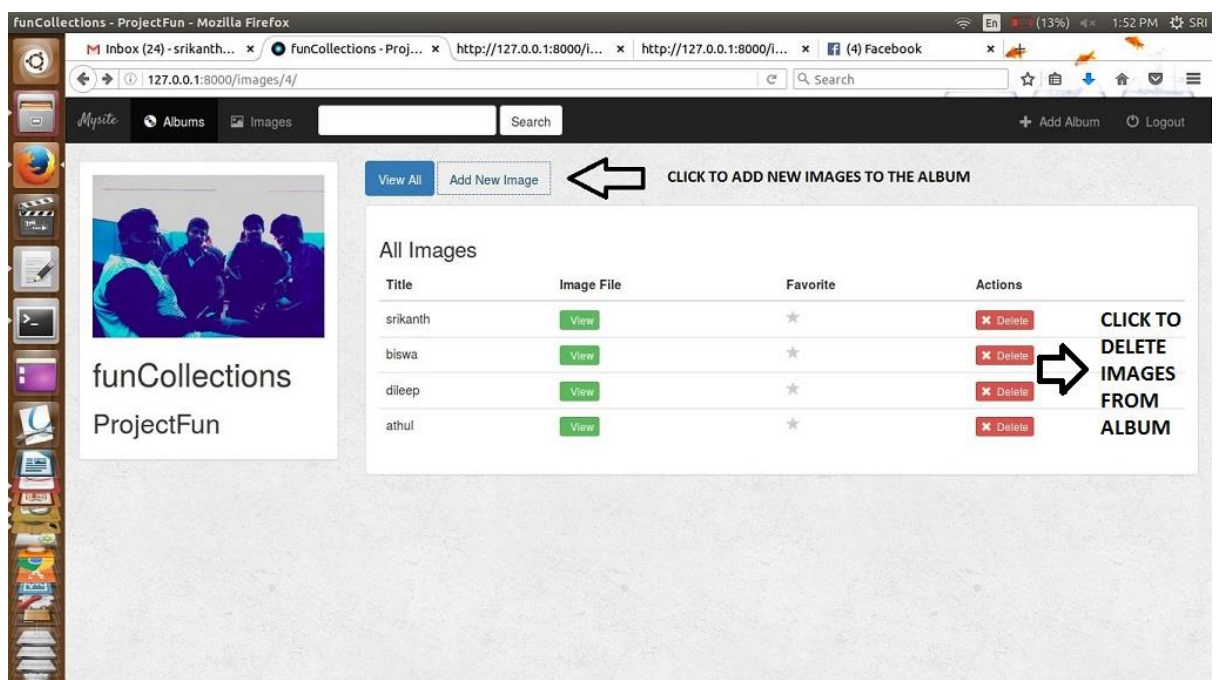


STEP-4: The user can directly delete the entire album.

STEP-5: If the user goes to view all the images in all the albums database.



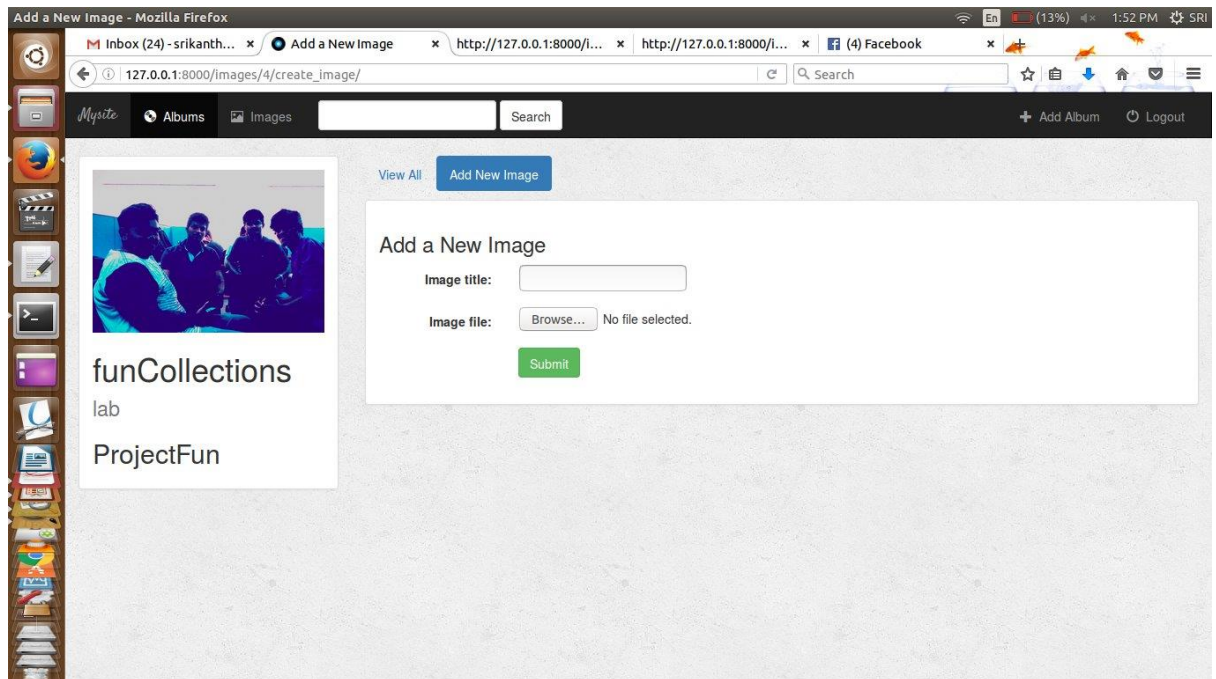
STEP-6: If the user clicks to view the images of a particular album.



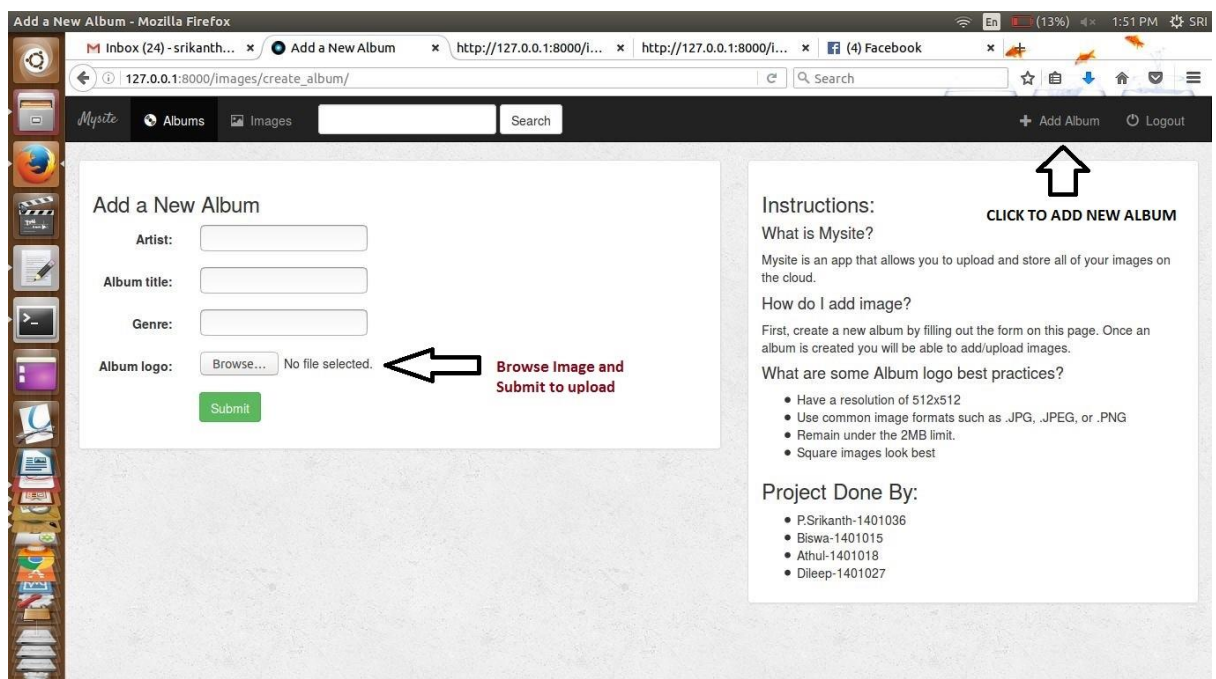


STEP-7: The user can also delete images within the album.

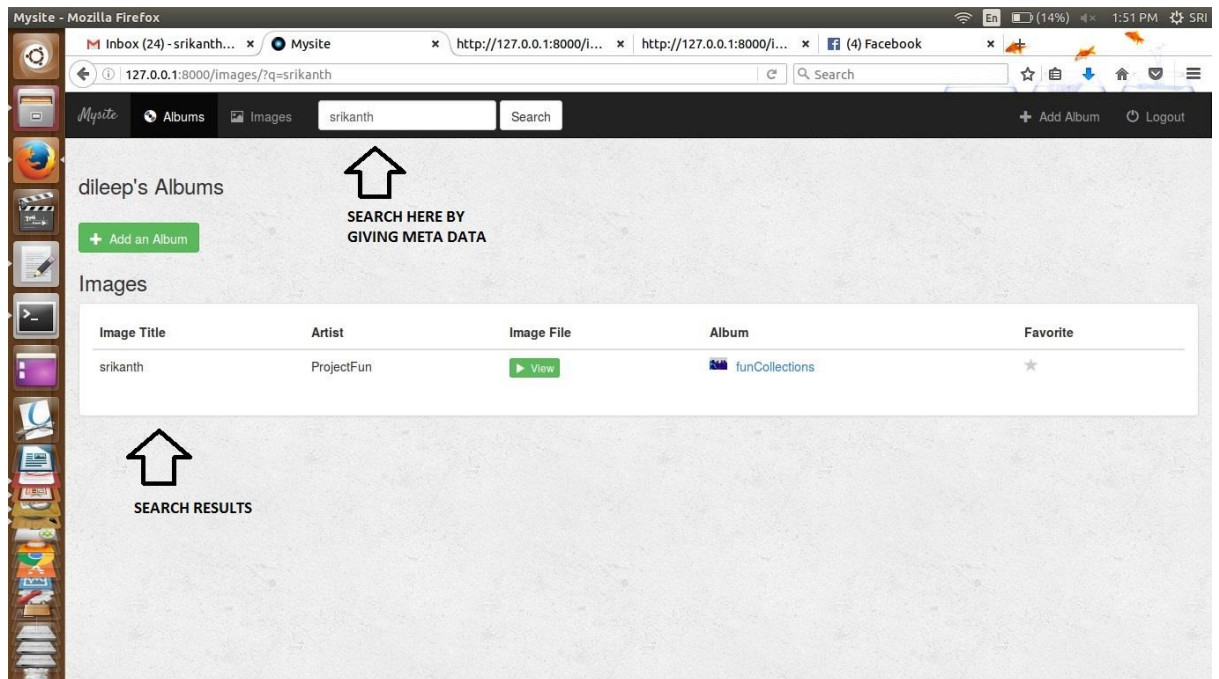
STEP-8: If the user chooses to add new images.



STEP-9: If the user wants to add new albums.



STEP-10: If the user wants to search for a particular album or a particular image.



THANK YOU