**SAILIFAQ FEATURE: FILE UPLOAD USING AMAZON AWS S3**

1.To run the FAQ Project

Ensure that the pre-requisites are completed:

(a) Laravel installed.

(b) Github connected via VCS.

(c) Heroku login completed.

(d) Laravel Configuration.

2. Do git clone <https://github.com/sailis/sailifaqnew.git>

The repository includes the form for uploading file on laravel.

3.copy .env.example to .env

4. Follow the steps for laravel authentication:

5. Type the following commands in the Terminal:

**laravel new**

This will create a new laravel application.

**sudo apt-get install php-sqlite3**

This will install php sqlite 3 database.

**touch database/database.sqlite**

This will connect sqlite as the current database.

Include the following in the config>database.php file:

top:  
**$heroku\_db\_url = parse\_url(env('DATABASE\_URL', "postgres://forge:forge@localhost:5432/forge"));**

inside the list of databases:

**'pg-heroku' => [  
'driver' => 'pgsql',  
'host' => $heroku\_db\_url['host'],  
'database' => substr($heroku\_db\_url['path'], 1),  
'username' => $heroku\_db\_url['user'],  
'password' => $heroku\_db\_url['pass'],  
'charset' => 'utf8',  
'prefix' => '',  
'schema' => 'public',  
],**

Now, type the following commands in the terminal:

**php artisan migrate  
php artisan make:auth**

**echo web: vendor/bin/heroku-php-apache2 public/ > Procfile**

This will create a Procfile with web:vendor/bin/heroku-php-apache2 public written inside it.

Ensure that you mention public.

Now, we create a Heroku app and create authentication for that app:

**heroku apps:create $app\_name**

Mention the name of the app that you want to create.

**heroku addons:create heroku-postgresql:hobby-dev --app $app\_name**

This will install the Postgre addon for your app.

**php artisan --no-ansi key:generate --show**

This will show the APP\_KEY for your app.

Copy the **APP\_KEY**.

Now type

**heroku config:set APP\_KEY=**

And Paste the APP\_KEY after the =

This will assign the APP\_KEY as the one that you pasted

**heroku config:set APP\_LOG=errorlog**

This will set an errorlog for your app.

**heroku config:set APP\_ENV=development APP\_DEBUG=true APP\_LOG\_LEVEL=debug**

This will set the application environment for your app.

**heroku config:set DB\_CONNECTION=pg-heroku**

This will specify that your Database Connection is pg-heroku as we mentioned in the database.php

Now, Push these changes to the Master branch of your app on heroku.

**git push heroku master**

Now, migrate these changes to your app.

**heroku run php artisan migrate**

If you get an error,

try

**heroku run php artisan migrate --app (appname)**

Now,

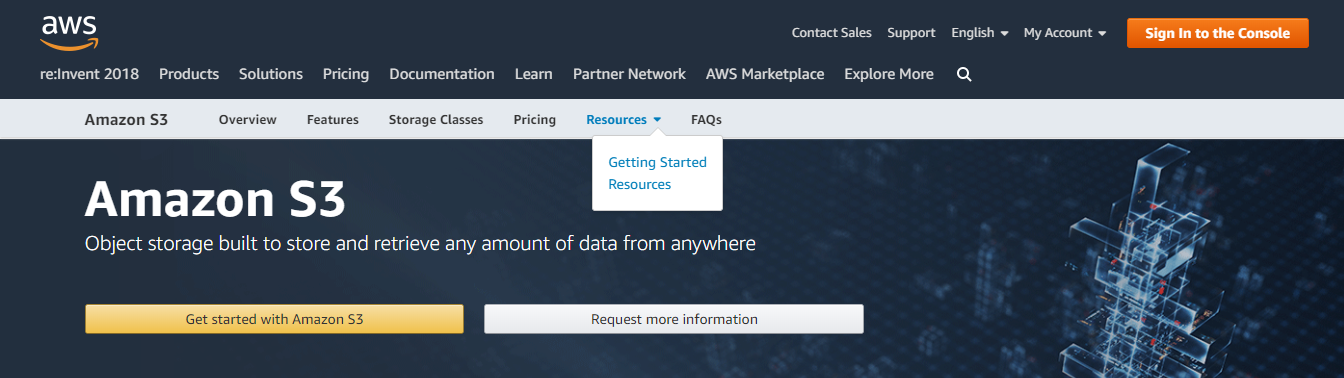
we begin the instructions for **AMAZON AWS S3**.

Go to

**https://aws.amazon.com/s3/**

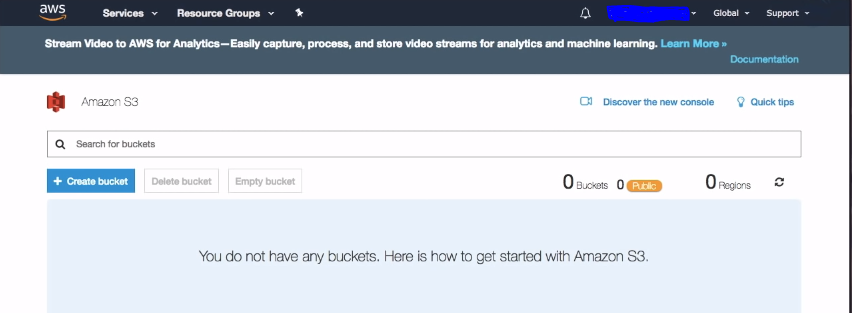
And Click on the "**Get Started with Amazon S3**" button.

And Create an Account**.**

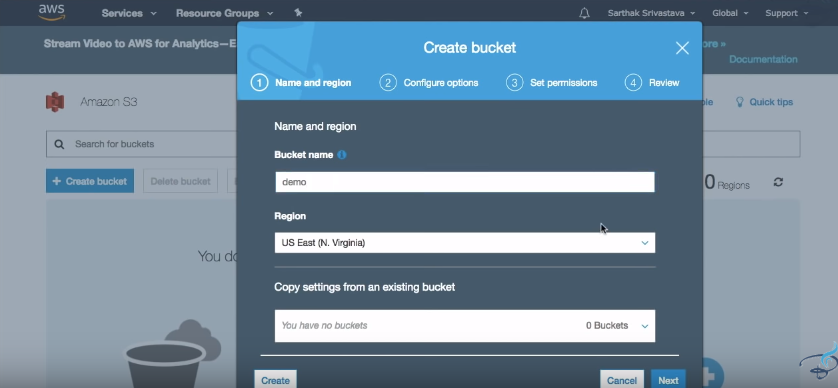
****

Now, after you Login,

Click on the **CREATE BUCKET** button



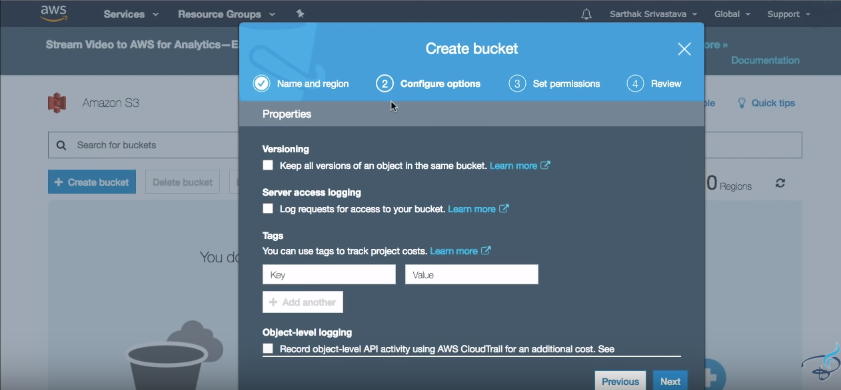
Now fill in the details in the prompt

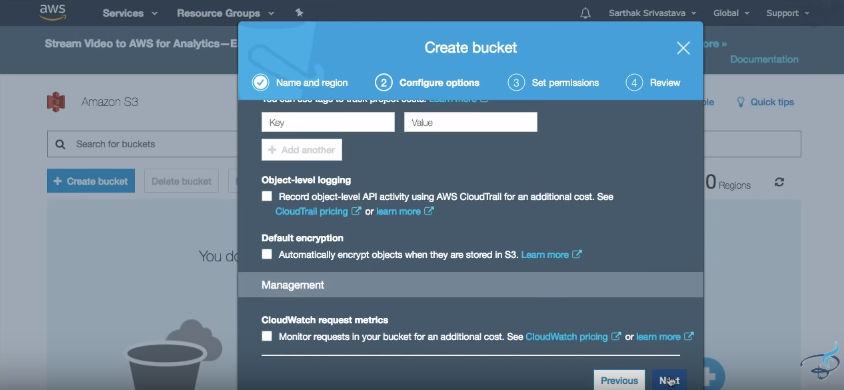


Name your bucket.

Choose your geographical region.

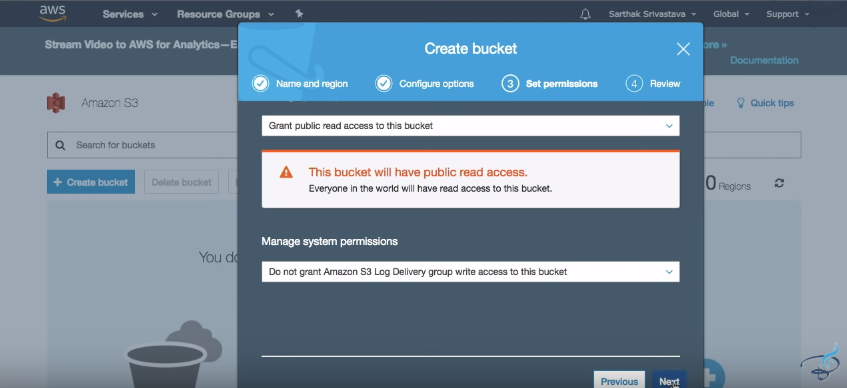
And press NEXT.



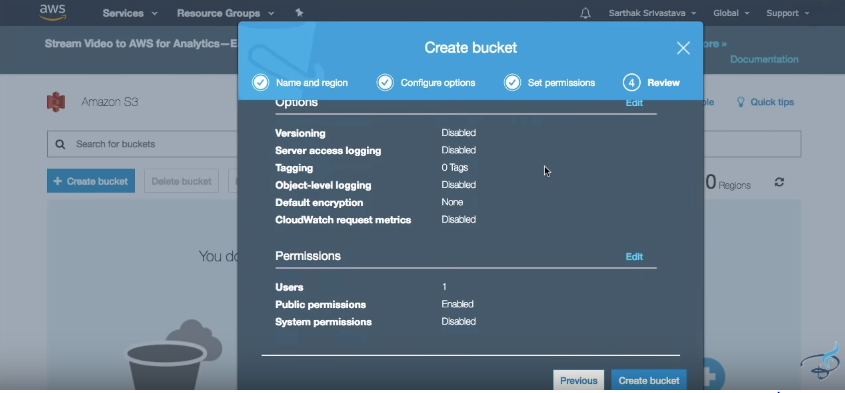


Press NEXT.

Change the Permissions as follows,



This will give public read access to all the contents of your bucket.

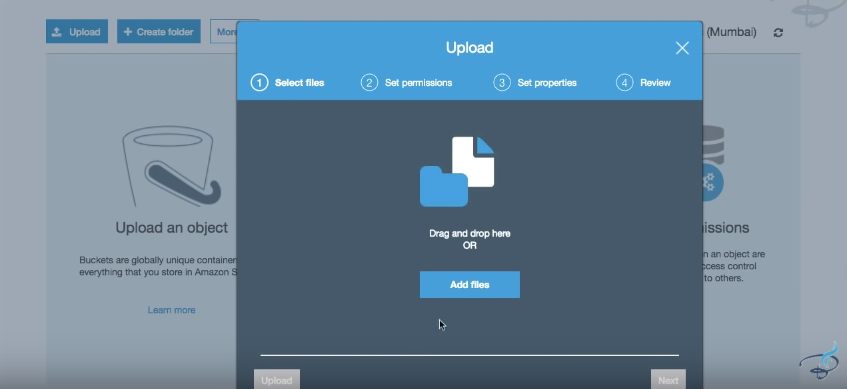


Now, Press the CREATE BUCKET button.

Now, you have successfully created your own bucket on amazon aws.

You can view your bucket.

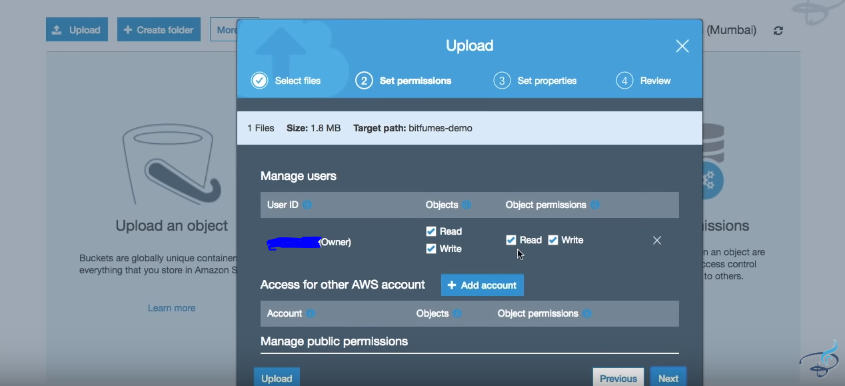
Now, click on the GET STARTED option.



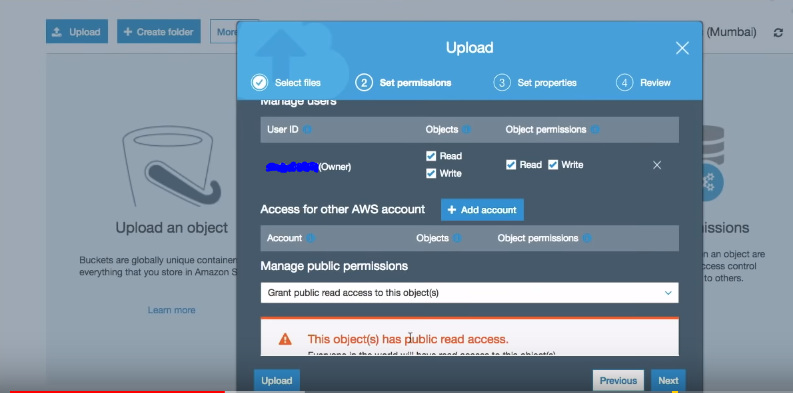
Here, you can add files from your computer to the bucket.

Test the bucket by uploading files from your computer.

Give the file the permission to be READ and WRITE.



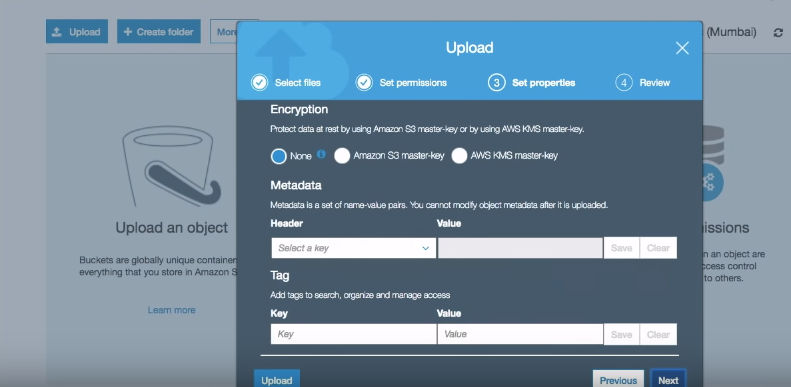
Grant public read access to the object.



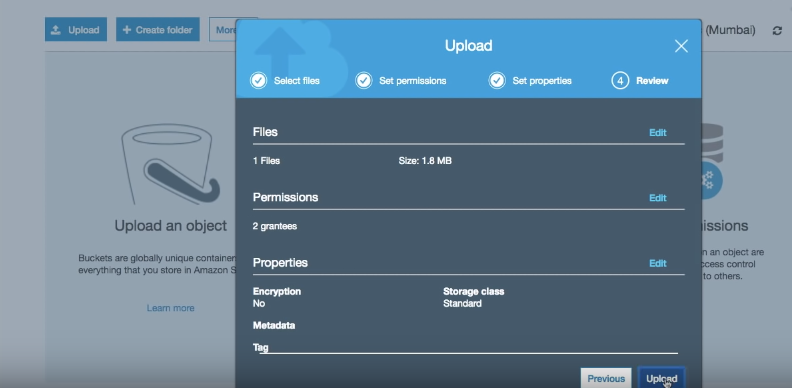
Click on NEXT.

Keep the Encryption details as it is.

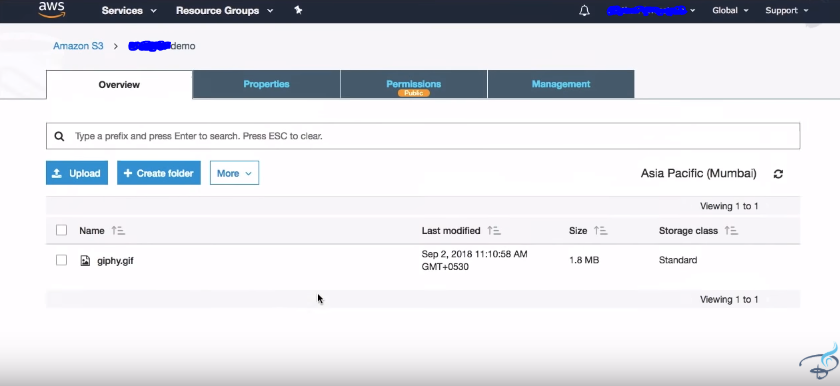
Click on NEXT.



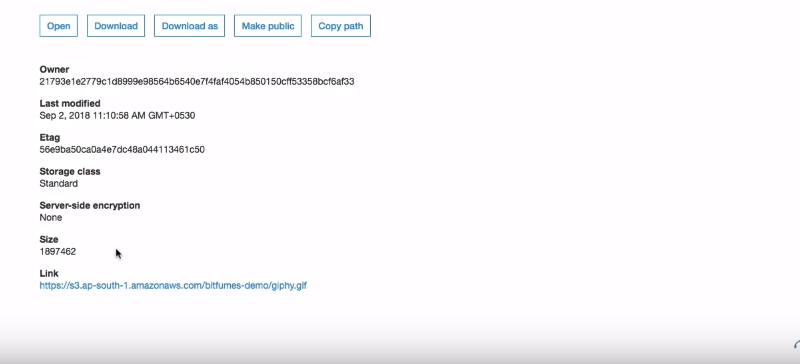
Now, Click on UPLOAD.



Your file will now show up inside your bucket.



When you click on the file, you can now see the details of your file and you can Open or Download your file from the bucket.



Now,

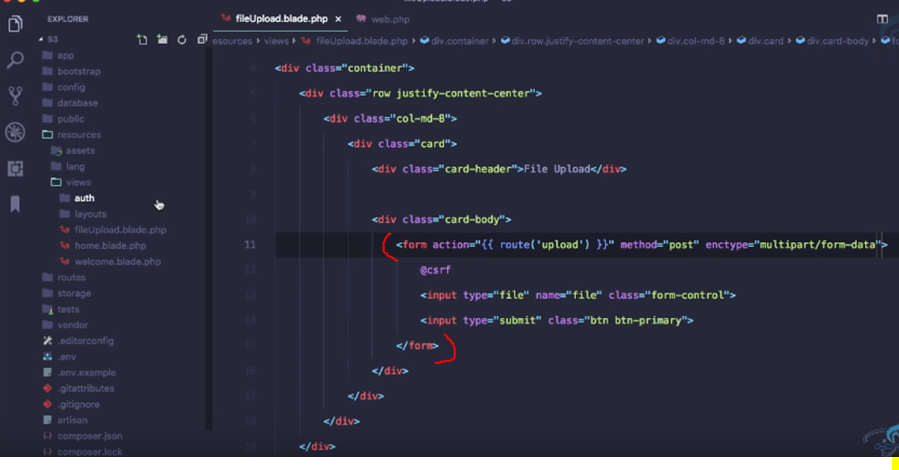
Go to **PHPSTORM**.

Now for the form for uploading a file can be created in one of the two ways:

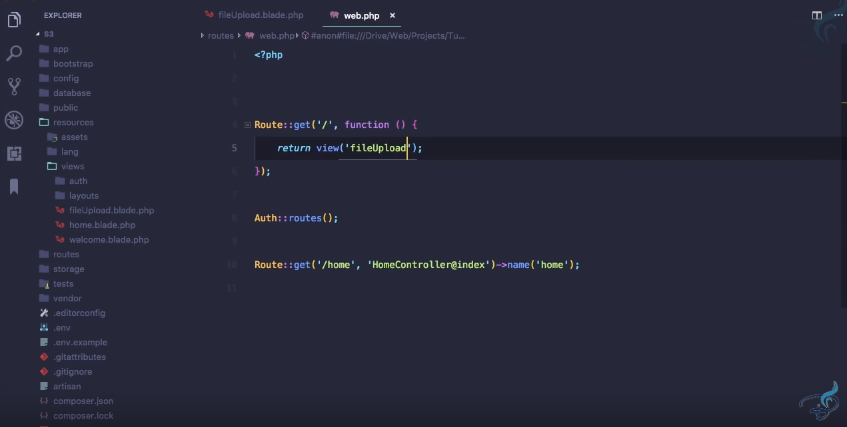
(a) Create a file named 'FileUpload.blade.php'

Copy the code from your 'welcome.blade.php' and Paste it in 'FileUpload.blade.php'

And make the following changes.

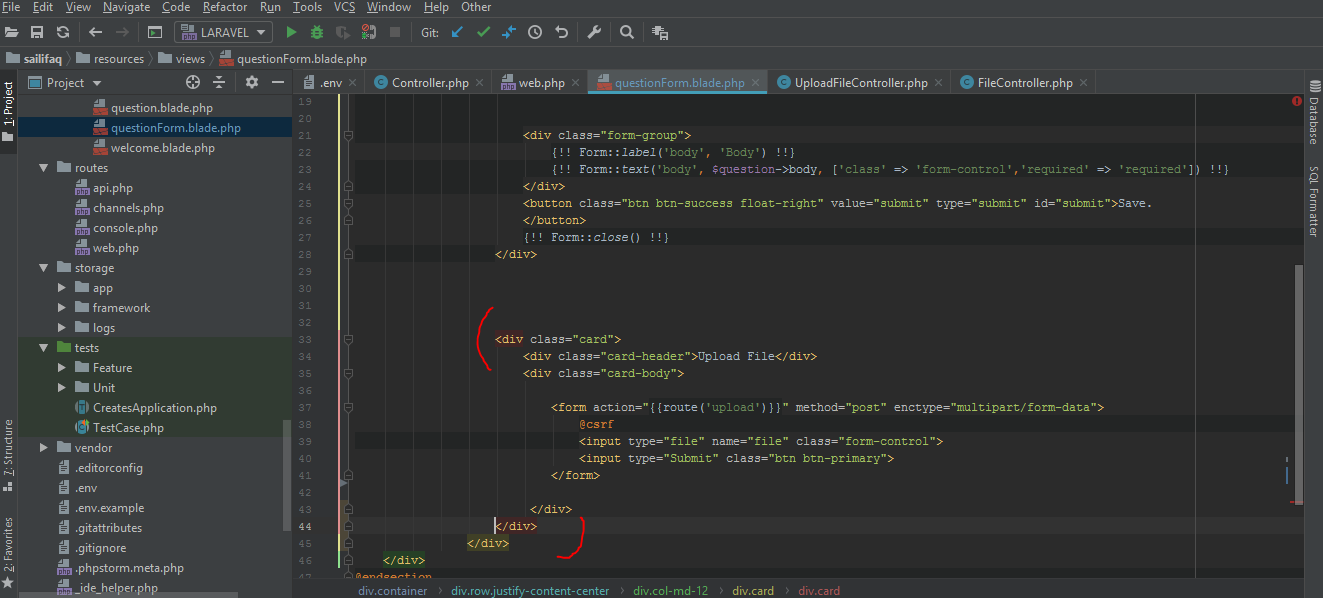


Now, make the following changes to your web.php:



This will change your current welcome page to the FileUpload page.

(b) You could also insert the FileUpload Form in your questionForm.blade.php



This will include your FileUpload Form under the Question Upload Form.

Similarly, you can add this piece of code inside your answerForm.blade.php to give an option to the user to upload a file along with their answer.

Open the routes>web.php file and include the following in your list of routes.

Route:: *post*('upload',**function**() {  
  
 request()->file('file')->store(  
  
 'my-file',  
 's3'  
 );  
  
})->name('upload');

This will create a route which specifies that :

-Name of file uploaded and stored will be 'my-file'

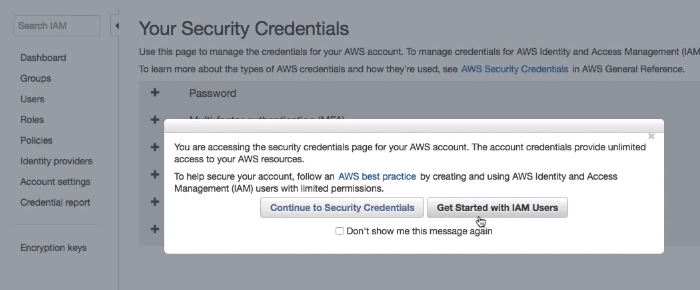
-The file will be uploaded and stored on 's3'

Now, include the following in your .env file:

**AWS\_ACCESS\_KEY\_ID**= ###############  
**AWS\_SECRET\_ACCESS\_KEY**= ###############  
**AWS\_DEFAULT\_REGION**=#region#  
**AWS\_BUCKET**=##############  
**AWS\_URL**=http://#bucketname#.s3-#region#.amazonaws.com

Now to obtain these credentials, Go to your profile and click on MY SECURITY CREDENTIALS

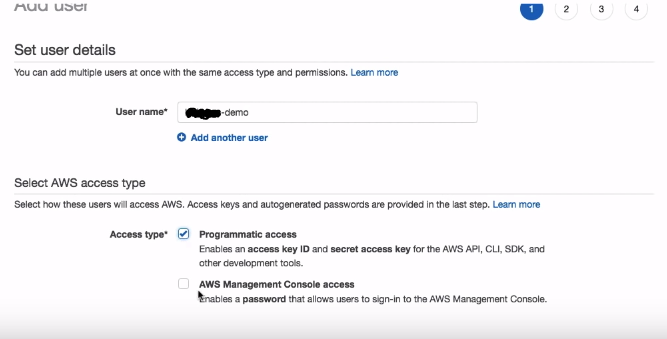
This will give you the following prompt.



Click on Get Started with IAM Users.

And Click on ADD USER.

Name the User and give it Programmatic access.



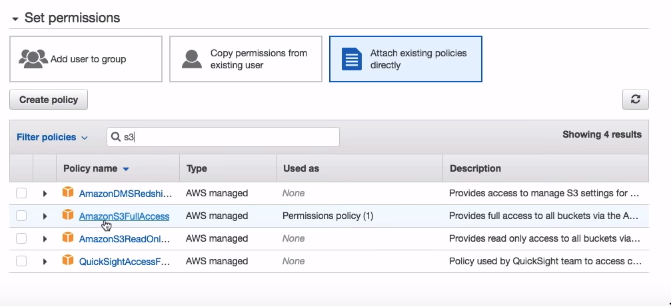
Click on NEXT.

NOW, Click on ACCESS EXISTING POLICIES DIRECTLY.

This will give you several options.

Search for 's3'.

Click on AmazonS3FullAccess and click on NEXT.



Click on CREATE USER.

Now, you will be able to access your

ACCESS KEY ID and SECRETACCESS KEY.

Copy them and paste them inside your .env file accordingly.

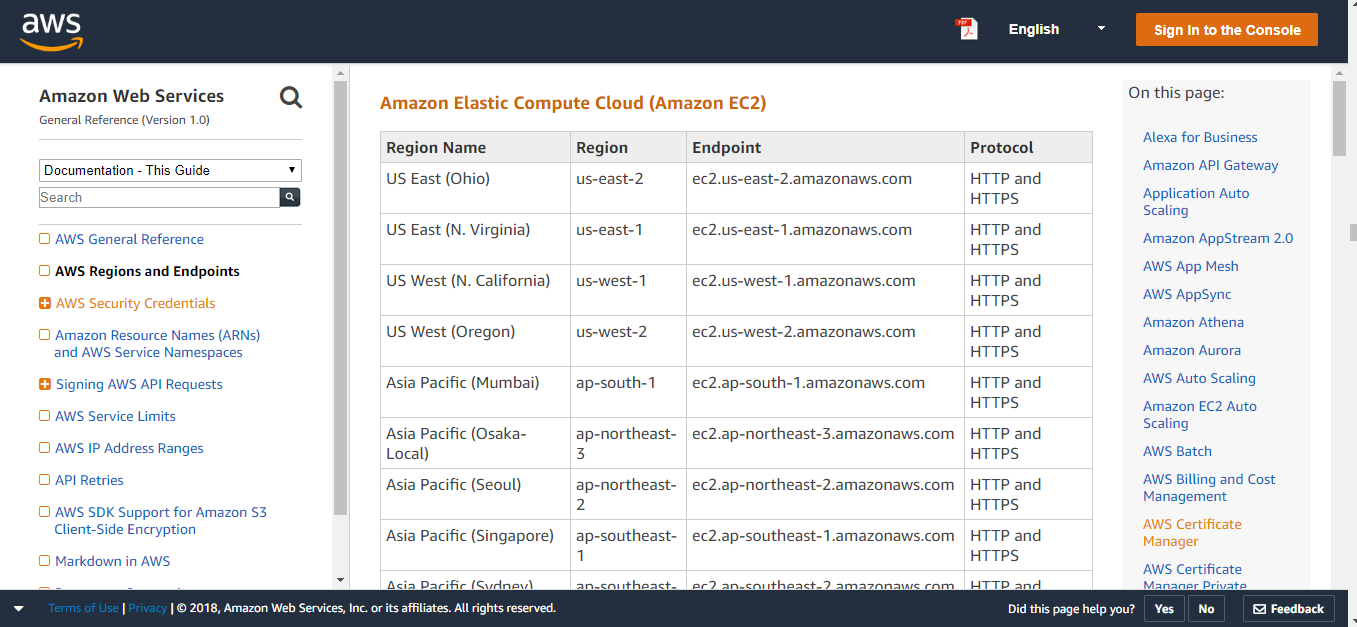
Also, enter the bucket name.

To find the correct format of mentioning your region, go to

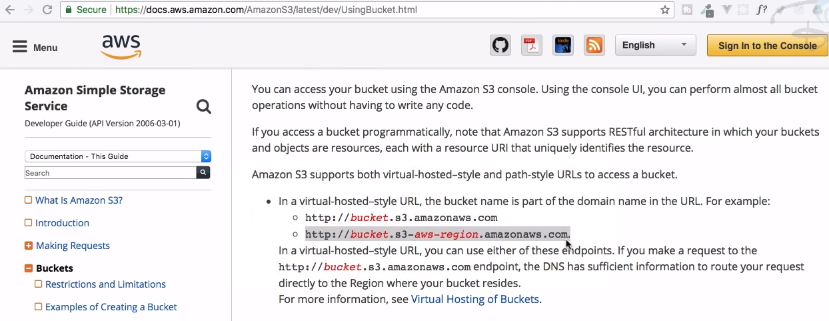
https://docs.aws.amazon.com/general/latest/gr/rande.html#ec2\_region

Now find your geographical region and follow the format.

for eg.



To enter your AWS\_URL, follow the format

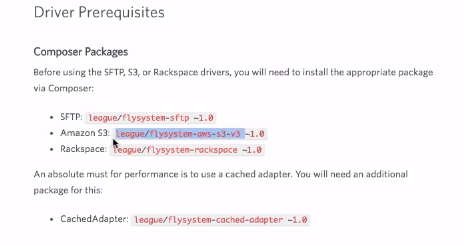


Finally,

Go to

https://laravel.com/docs/5.7/filesystem

and scroll down to copy the package needed for connecting to Amazon S3 from your app.



Now, type in your terminal

**composer require league/flysystem-aws-s3-v3**

If you encounter an error,

try

**composer require league/flysystem-aws-s3-v3 --dev**

If you still encounter an error which says that you require other packages to install this package,

try installing the following packages:

**composer require phpspec/phpspec**

**composer require henrikbjorn/phpspec-code-coverage**

**composer require league/flysystem**

Now try installing the aws package again.

**composer require league/flysystem-aws-s3-v3**

It will work now.

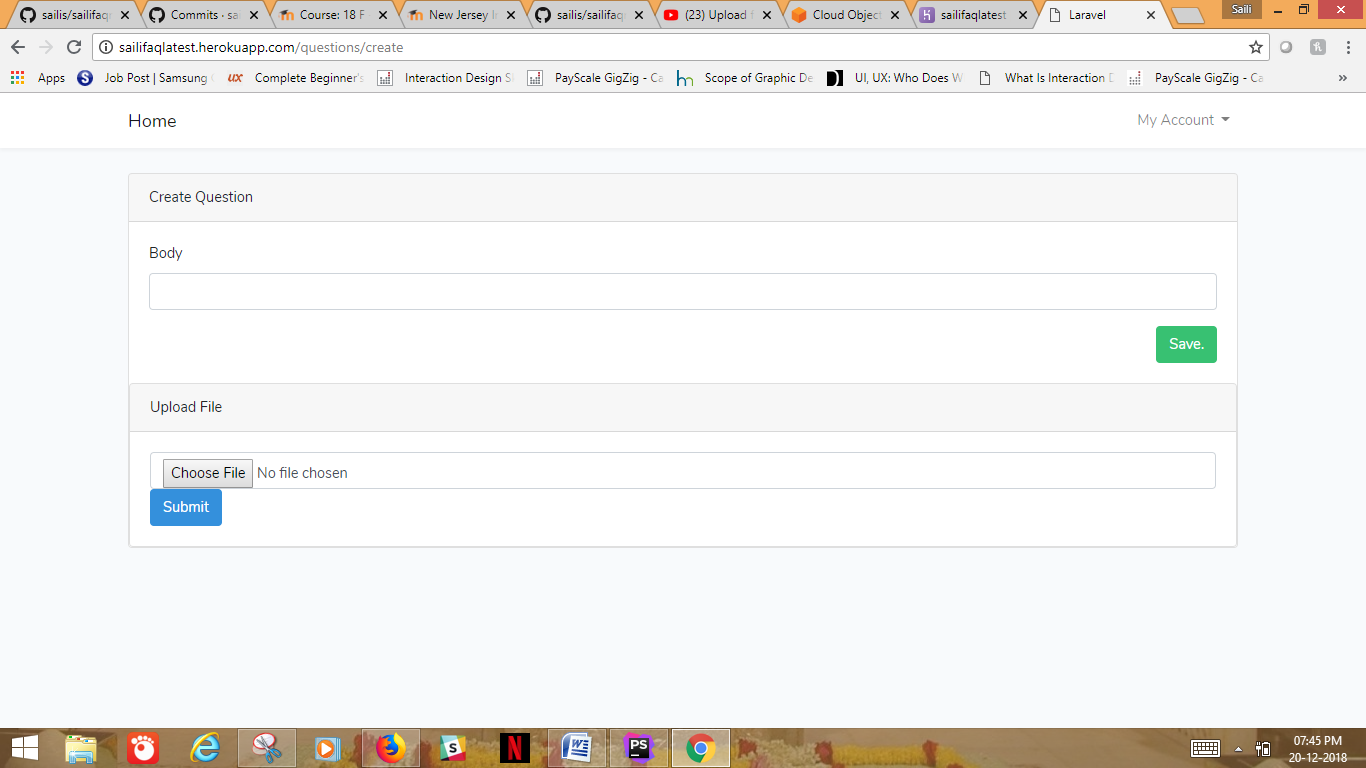
Now,

**Finally**

**Go to your Heroku App.**

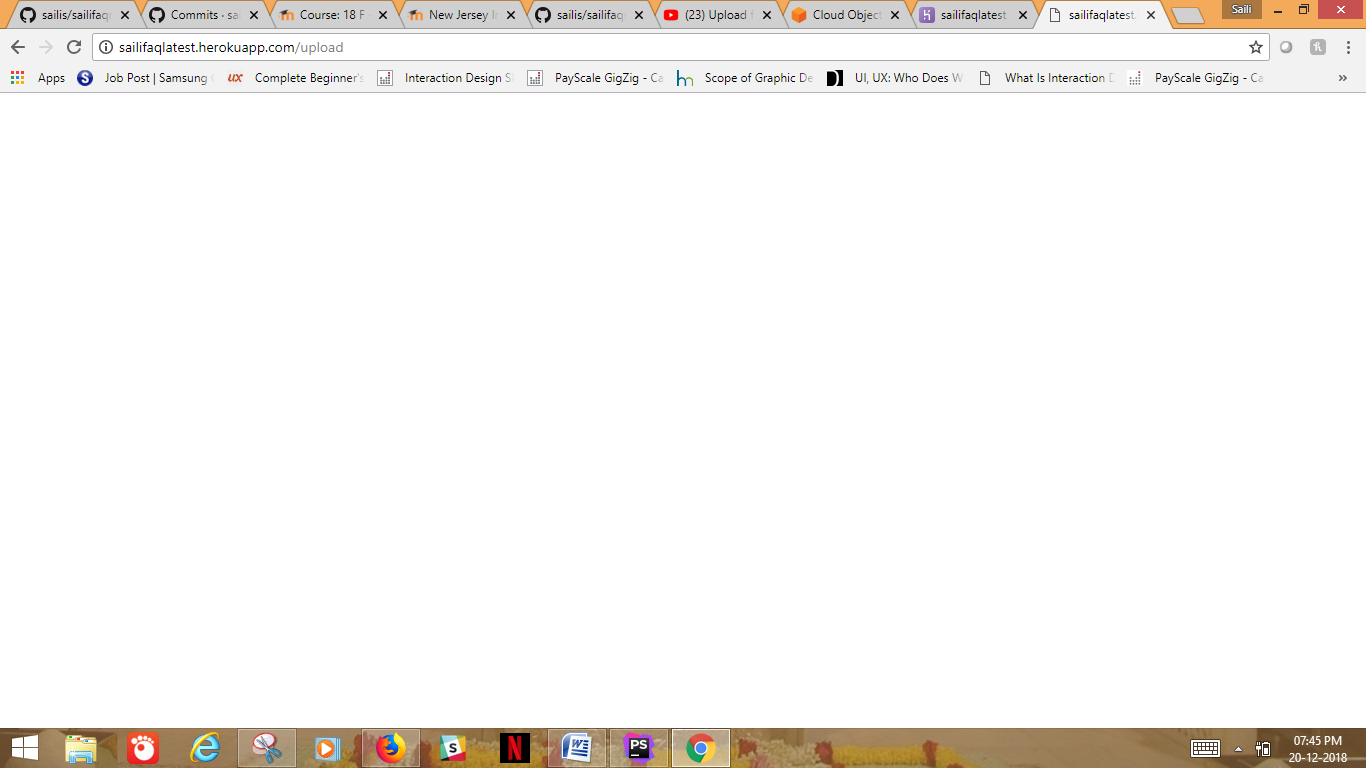
**And Open it!**

**You will get your laravel app with a file upload form.**

****

**After Uploading the file,**

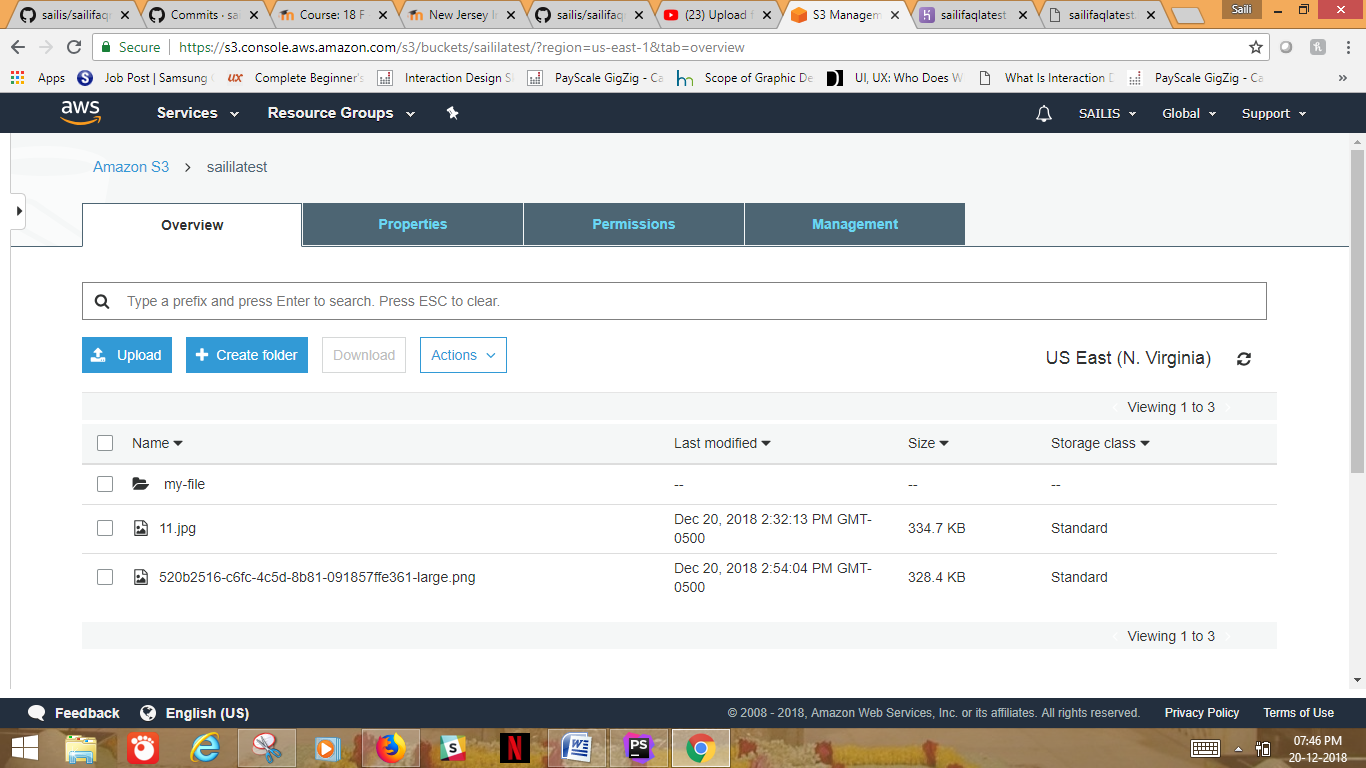
**You may get a blank screen.**

****

To verify if your Upload has worked,

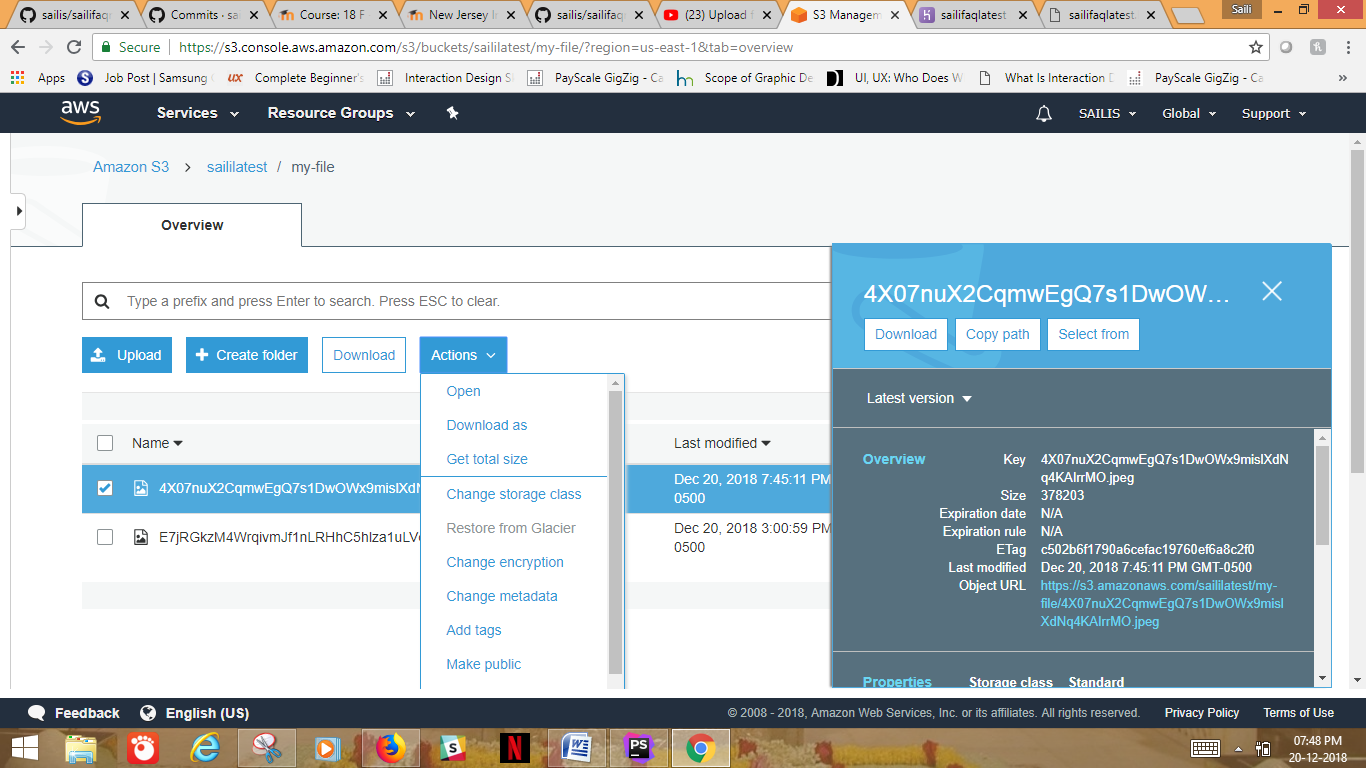
Go to your Amazon AWS S3 console,

click on the Bucket and you can see the File(s) you uploaded.

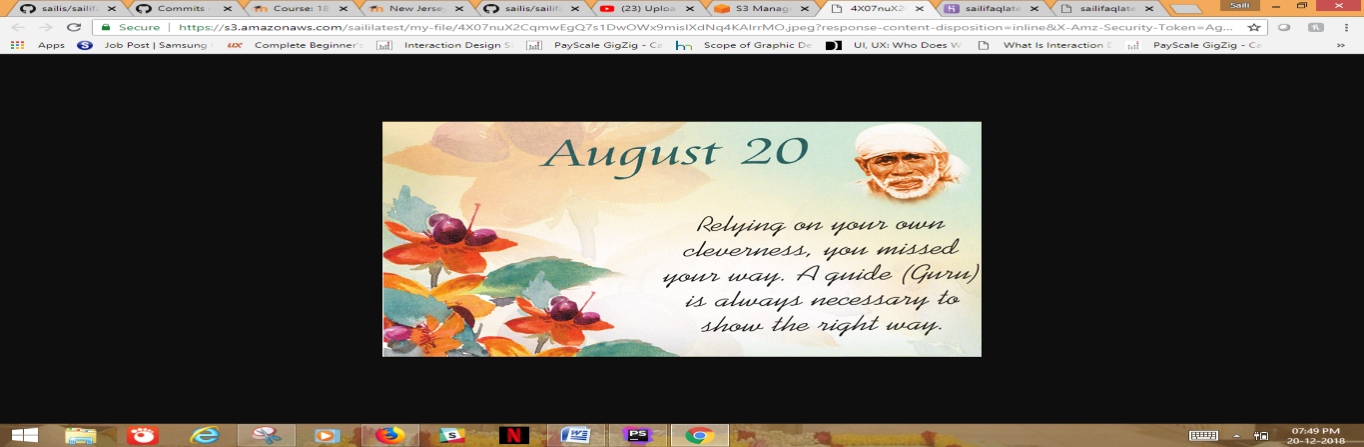


When you click on the file,

You have the option to OPEN, DOWNLOAD etc



When you OPEN your file, it will open in a new tab:



**YAYIE! IT WORKS!**

**FEATURE**:

**Upload file in Question**: This allows the user to include a file(text, picture or video) along with their question to support their doubts. This way, if the user has a doubt related to the picture, they can upload the picture to point out their issue.

**Upload file in Answer**: This allows the user to upload a file(text, picture or video) along with their answer to support their thoughts. This way, the user can explain their answer via a picture.

**USER ACTIONS:**

(a) The user can Register or Login to create their profile on the website.

(b) The user can create or update their profile.

(c) The user can post (ask) questions on the website.

(d) The user can view and edit and delete the question that they posted.

(e) The user can upload files (text, picture or video) to support and explain their answer.

(f) The user can view and answer a question posted by other users.

(g) The user can also delete the answer that they posted.

(h) The user can upload files (text, picture or video) to support and explain their question.

**EPIC:**

A user can access the website to view and ask questions and also answer questions posted by other users.

**USER STORIES:**

1. As user ,  
   I want to register my account ,  
   so that I can access the website.
2. As user ,  
   I want to login to my account ,  
   so that I can access the website.
3. As user who wants to ask a question ,  
   I want to post a question ,  
   so that I can get an answer from other users.
4. As user who wants to ask a question ,  
   I want to upload a file ,  
   so that other users can understand my problem and I can get an answer from other users.
5. As user who wants to answer a question ,  
   I want to post an answer ,  
   so that I can solve the problems of other users.
6. As user who wants to answer a question ,  
   I want to upload a file ,  
   so that I can solve the problems of other users with details about my answer.