**# Steps before cloud deployment:**

We need to change our code a bit so that it works unhindered on the cloud, as well.

a) Add a file called ‘gitignore’ inside the project folder. This folder contains the list of the files which we don’t want to include in the git repository.

Add a file called ‘Procfile’ inside the ‘reviewScrapper’ folder. This folder contains the command to run the flask application once deployed on the server:

Open a command prompt window and navigate to your ‘reviewScrapper’ folder. Enter the command ‘pip freeze > requirements.txt’. This command generates the ‘requirements.txt’ file.

**# Deployment to Heroku:**

• After installing the Heroku CLI, Open a command prompt window and navigate to your project folder.

• Type the command ‘heroku login’ to login to your heroku account.

• After logging in to Heroku, enter the command ‘heroku create’ to create a heroku app. It will give you the URL of your Heroku app after successful creation.

• Before deploying the code to the Heroku cloud, we need to commit the changes to the local git repository.

• Type the command ‘git init to initialize a local git repository as shown below:

• Enter the command ‘git status’ to see the uncommitted changes

• Enter the command ‘git add .’ to add the uncommitted changes to the local repository.

• Enter the command ‘git commit -am "make it better"’ to commit the changes to the local repository.

• Enter the command ‘git push heroku master’ to push the code to the heroku cloud.

• After deployment, heroku gives you the URL to hit the web API.

• Once your application is deployed successfully, enter the command ‘heroku logs -tail’ to see the logs.