1. LOGIN Details
   1. Github <[anaxxtsprojxxxs10@xmail.com](mailto:anaxxtsprojxxxs10@xmail.com)> <GithubAxxxxx@1>
   2. Heroku <[anaxxtsprojxxxs10@xmail.com](mailto:anaxxtsprojxxxs10@xmail.com)> <HerokuAxxxxx@1>
2. Install python and ensure that the add to path of environment variables is checked
3. Install git
4. Create SSH key
   1. Create a folder C:\Anupam\GIT\_heroku\_repository
   2. Git bash on <GIT\_heroku\_repository>
   3. In terminal run <pwd> and it will show current directory
   4. Now run <mkdir .ssh>
   5. Now <cd .ssh>
   6. Now we will create ssh key. Run <ssh-keygen.exe> hit enter
   7. Hit enter again
   8. Enter password 2 times <I used word as password>
   9. There are two files created in <C:\Users\nuey\.ssh>
5. Heroku toolbelt
   1. Go to <https://devcenter.heroku.com/categories/command-line>
   2. Download by clicking Install The Heroku CLI
   3. Set path to Heroku should be checked (all 3 checked)
   4. Close the git bash and reopen after toolbelt is installed.
   5. In git terminal prompt type <heroku –version> and it should show you version
6. Virtual Environment
   1. Create a virtual environment using conda or PyCharm. In our example it is <C:\Users\nuey\anaconda3\envs\heroku\_repository>
   2. When you open a cmd terminal and run <conda info –envs> it will show all virtual environments
   3. Open a cmd and run conda activate heroku\_repository and it will activate virtual environment
7. Create a django project
   1. Run <django-admin startproject mysite>
   2. Cd in mysite <cd mysite>
   3. Run <python manage.py runserver> and see in browser
8. Create App
   1. Run python manage.py startapp address\_book
   2. Add app in settings ,py
9. Git version control setup
   1. At this stage, in the folder, django app and projects are created but no local repository is created and linked to Git. which we will do in below steps
   2. On mysite folder, git bash here
   3. Run <git config –global user.name “ab will also do”>
   4. Run <git config –global user.emal "[analytsprojects10@gmail.com](mailto:analytsprojects10@gmail.com)">
   5. Run <git config –global push.default matching>.
   6. Run <git config –global alias.co checkout
   7. Run <git init>
   8. Now add all files to index (making all files ready for commit) by git add . <this period at the end means all files to be added>
   9. Now commit all these changes <git commit -am “all files added in one step”>
10. Pushing to Heroku (note, we only committed using git bash, not pushed)
    1. Login to heroku.com <[analyticsproxxxxs10@xmail.com](mailto:analyticsproxxxxs10@xmail.com)><HerxxxAxxxxx@1>
11. Heroku CreateApp
    1. Open a normal cmd terminal and type <heroku login>. A new browser will open and prompt to login. On successful browser login, the cmd terminal also shows analytcsprojects10@gmail.com
    2. Now add SSH keys. Run <heroku keys:add> and it will find the keys from anywhere on computer where SSH was created. No need to go to that path
    3. Now create an app on Heroku by running <heroku create>. It will create app and you can copy the first past and paste on browser<<https://frozen-anchorage-65212.herokuapp.com>>
    4. Now we have to push our code to heroku.
12. Install Dependencies on heroku
    1. First the secret key needs to be added from settings.py to heroku.
       1. Open settings.py and copy key without quotes.
       2. Login to heroku dashboard and look for frozen-anchorage-65212 and click on it go to settings/Reveal Config vars
       3. Add the key without quotes and in SECRET\_KEY as it appears in settings.py
    2. Install gunicorn (which is a web sever heroku uses)
       1. Open another cmd terminal and activate virtual environment <conda activate heroku\_repository> run <pip install guinicorn>
       2. Run <pip install django-heroku>
       3. Run <pip install python-decouple>
    3. Modify settings.py
       1. Open settings.py
          1. Import django\_heroku
          2. from decouple import config
          3. Import dj\_database\_url
          4. At bottom add django\_heroku.settings(locals())
          5. Create one .env file at <C:\Anupam\GIT\_heroku\_repository\mysite>
          6. In the .env file, pase <SECRET\_KEY = 'django-insecure-lw\_(t$=s-$@)o@#1xg!v\*8kddxdg2bzt(lk1\_hr)462oeqmb(x'>
          7. Delete the SECRET\_KEY line from settings.py and add <SECRET\_KEY = config(‘SECRET\_KEY’)>
          8. Create a .gitignore file at and add .env in it
    4. Whitenoise
       1. Create a folder called static at same level as manage.py
       2. Add \_\_init\_\_ in static folder
       3. In settings file add <STATICFILES\_DIRS = [os.path.join(BASE\_DIR, 'static')]>
       4. In MIDDLEWARE add 'whitenoise.middleware.WhiteNoiseMiddleware',
       5. At bottom, after STATICFILES\_DIRS add <STATICFILES\_STORAGE = 'whitenoise.storage.CompressedManifestStaticFilesStorage'>
    5. Heroku profile and requirements.txt
       1. Two files (proc file) and requirements.txt file needs to be created
       2. Proc file tells heroku what type of app is this.. This is a web app
       3. Create Procfile at <C:\Anupam\GIT\_heroku\_repository\mysite> and add <web: gunicorn mysite.wsgi> in Procfile
       4. Now create requirements.txt file
          1. Cd the cmd terminal at <C:\Anupam\GIT\_heroku\_repository\mysite> and run <pip freeze > requirements.txt>
    6. Database stuff
       1. I had to place back the SECRECT\_KEY
       2. Run python manage.py runserver
       3. The migration and admin is not set as yet
       4. Run <python manage.py migrate>
       5. Run python manage.py createsuperuser <super super>
    7. Now ready to push the code to heroku
       1. So in settings.py removed the security key and uncommented security\_ket config line
       2. Commit the code to git (not push)
          1. Run <git add .>
          2. Run <git commit -am “push all files to git”>
       3. Now push the code to heroku

—------------------------------------------------------------------------------------------------------

Youtube https://www.youtube.com/watch?v=5d8AQFF0Ot0

1. 10 steps
   1. Ready django application
   2. Install additional packages
   3. Create runtime.txt
   4. Create Procfile
   5. Requirements.txt
   6. Create a github repo
   7. Create app on heroku platform
   8. Create a postgres db on heroku
   9. Modify setting.py for postgres db
   10. Modify settings.py for static files
2. Workflow
   1. Create django app
   2. Push django app to Github repo
   3. Pull django app on heroku from github repo
3. The django app is <C:\Anupam\GIT\_heroku\_repository\mysite> and is running
4. Pip install gunicorn
5. Pip install whitenoise
6. Pip install python-decouple
7. Pip install psycopg2
8. Create runtime.txt at <C:\Anupam\GIT\_heroku\_repository\mysite> and add <python=3.8.12>
   1. Create Proc file at <C:\Anupam\GIT\_heroku\_repository\mysite> and ad
      1. release: python manage.py migrate
      2. web: gunicorn mysite.wsgi --log-file=-
9. Generate requirements.txt
   1. Pip freeze > requirements.txt
10. Create a new heroku app from heroku site <mysite-forzen-heroku>
11. Now create a database for this new app
    1. You can add postgres and complete from site.
    2. From settings / view credentials you can get the database details
    3. Open settings.py and change the database from sqllite to postgres
12. Now add for static

import os

import django\_heroku

STATIC\_ROOT = os.path.join(BASE\_DIR,'staticfiles')

STATIC\_URL = 'static/'

ALLOWED\_HOSTS = ['mysite-forzen-heroku.herokuapp.com']

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.postgresql',

'NAME': ‘dpt4g1ifin9af’,

'USER': 'czgbyqxjsobgui',

'PASSWORD': '160f7478b7b21e1cad7db140937397523f57ecb2a62a2b93ede3441d9298ba33',

'HOST': 'ec2-52-70-186-184.compute-1.amazonaws.com',

'PORT': '5432',

}

}

django\_heroku.settings(locals())

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

1. Create git local repository for mysite
   1. Git bash on <C:\Anupam\GIT\_heroku\_repository\mysite>
   2. Run <git init>
   3. Link to remorse git repository <$ git remote add main <https://github.com/ab10projs/heroku_repository.git>>
   4. Git add .
   5. Git commit -am “first commit”

**To work and push the changes to Live site from Local repo to Heroku site**

1. Make changes to the setting.py file for security\_key by uncommenting and commenting so that it works in local environment
2. Once the code is working fine on local, make the changes to setting file for Heroku
3. Push the changes to github
4. Logon to Heroku
   1. Go to deploy
   2. In App connected to Github, disconnect and then hit search button and it will show all repositories to connect.
   3. Deploy the master branch

—------------------------------------------------------------------------------------------------------

Youtube <[Python Django Tutorial: Deploying Your Application (Option #2) - Deploy using Heroku - YouTube](https://www.youtube.com/watch?v=6DI_7Zja8Zc)>

1. Virtual environment <heroku\_repository>. On a cmd just write conda and it will recognize commands
2. The virtual environment is at <C:\Users\nuey\anaconda3\envs\heroku\_repository>
3. The django project is at <C:\Anupam\GIT\_heroku\_repository\mysite>
4. Open a cmd and activate heroku\_repository and cd to <C:\Anupam\GIT\_heroku\_repository\mysite>
5. Create a new github repository called heroku\_rep
6. Now git bash at <C:\Anupam\GIT\_heroku\_repository\mysite> and run <git init>
7. Now link this local repository with github <**git** remote add origin <remote\_repo\_URL>>
8. Run <git remote -v> and it will show the remote repository linked
9. Now add local files to index <git add .>
10. Run <git commit -am “comments”>
11. Run <git push origin master>