

Topic	DEPLOYMENT	
Class Description	<p>The kid will learn to build and deploy a small API based program on the cloud first testing it locally. This expands their expertise in Python deployment as they learn more advanced programming deployment of web application concepts related to cloud applications.</p>	
Class	ADV-C263	
Class Time	60 mins	
Goals 	<ol style="list-style-type: none"> 1. Understanding the concept of deployment. 2. Learning about Github and Render and deploying web applications on them. 	
Resources Required	<ul style="list-style-type: none"> • Teacher Resources: <ul style="list-style-type: none"> ○ Use Gmail login credentials. ○ Laptop with camera. ○ Earphone with mic. ○ Notepad and pen. • Student Resources: <ul style="list-style-type: none"> ○ Use Gmail login credentials. ○ Laptop with camera. ○ Earphone with mic (optional). ○ Notepad and pen. 	
Class Structure	Warm-Up Teacher-Led Activity Student-Led Activity Wrap-Up Project Pointers and Cues	5 Mins 15 Mins 30 Mins 5 Mins 5 Mins
<p>NOTE: Teacher should deploy and test at least 1 web app before conducting this class on GitHub and Render.</p>		

Class Steps	Say 	Do 
Step 1: Warm-up (5 mins)	<p>In the previous class 262, we learned & revised the concepts of PAAS, why we used PAAS. Learned how to handle API and create a web app out of it using Flask.</p> <p>Q What is the full form of API? A The API is nothing but an Application Programming Interface.</p> <p>Q Why did we use a route decorator in a flask? A We used a route decorator for directing to a specific webpage to run a function.</p>	
Teacher Initiates Screen Share		
Step 2: Teacher-Led Activity (15 mins)	<p>In the previous class, we learned about PAAS and created a web app that gives us real-time weather stats report.</p>	

	<p>In this class, we will be understanding the concepts of deployment and we are going to deploy the same weather web app.</p>	
Let's revise the previous class,		
Q	<p>What do you mean by FLASK?</p>	
A	<p>Flask is nothing but a small/micro web framework used to create small web apps. This framework is used for web development and API handling of websites.</p>	
Q	<p>Why do we use Flask?</p>	
A	<p>Flask is a microframework; it means it does not contain complex settings or file structures. This makes Flask a user-friendly framework for web development. Companies like Google, Twitter, and Facebook work heavily on the flask.</p>	
Q	<p>What is API?</p>	
A	<p>The API is an Application Programming Interface that helps web applications interact with databases.</p>	
Q	<p>What is a database?</p>	
A	<p>A database is a place where data is stored.</p>	
That's great. Impressive.		
FLASK		
As you know Flask is nothing but a small/micro web framework used to create small web apps. And it is a very popular framework because to create a simple web application you		

don't need to install lots of libraries or external tools.

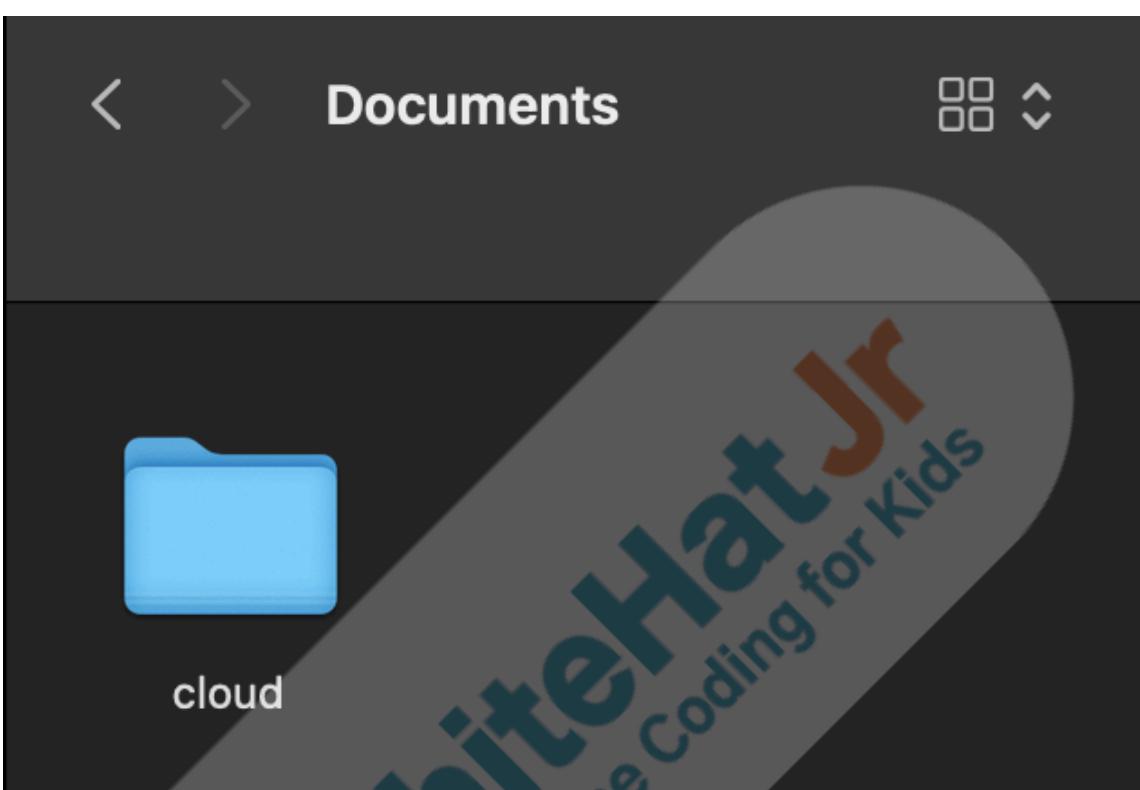
The flask is powerful. Because it gives features to create, update and deploy your web application in an easy way.



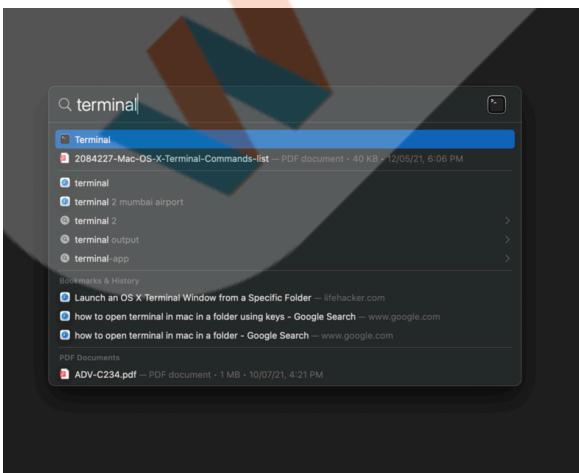
Let's run our web app which we have created in the previous class:

Testing the code For MAC:

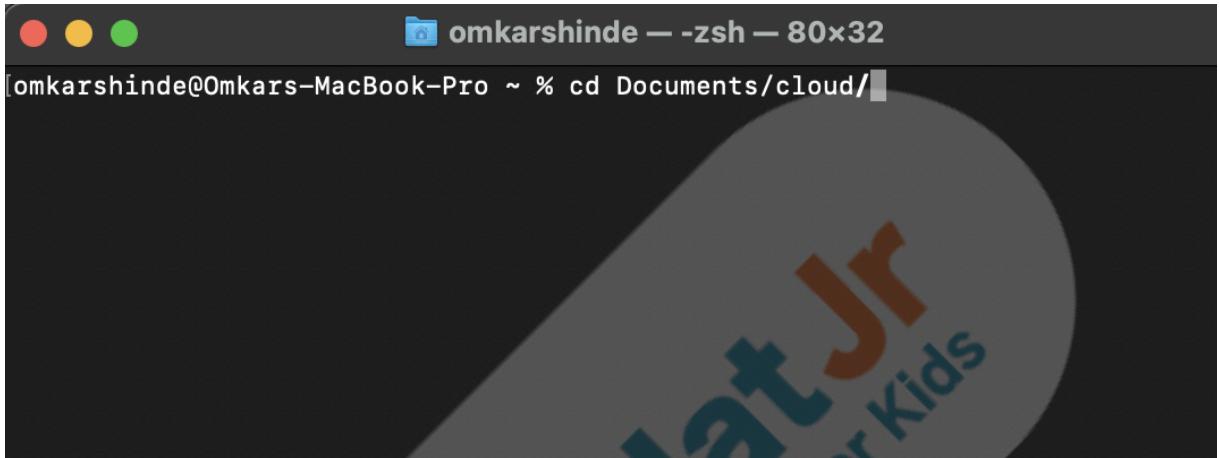
1. Go to the “cloud” folder in Documents directory.



2. Open the cloud folder.
3. And now to run the file in the terminal first open the terminal by pressing **command + SPACEBAR** key and type ‘terminal’.

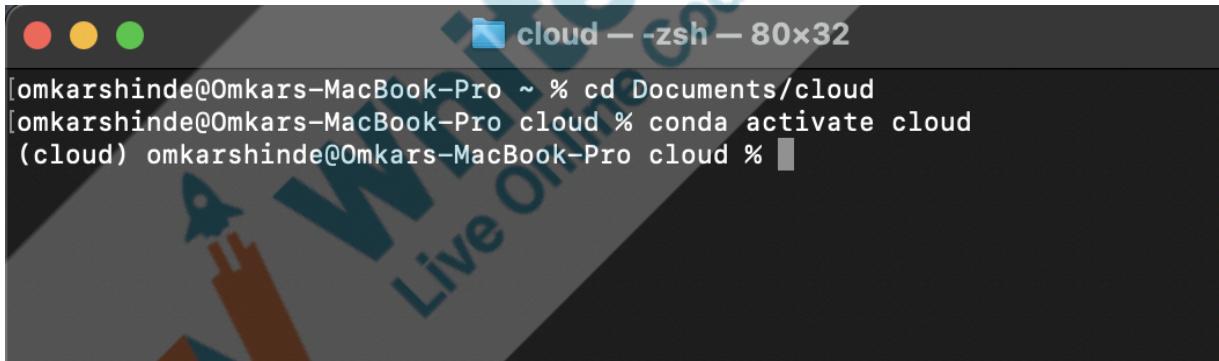


4. Then, locate the Documents folder using the **cd** command as **cd Documents/cloud/**



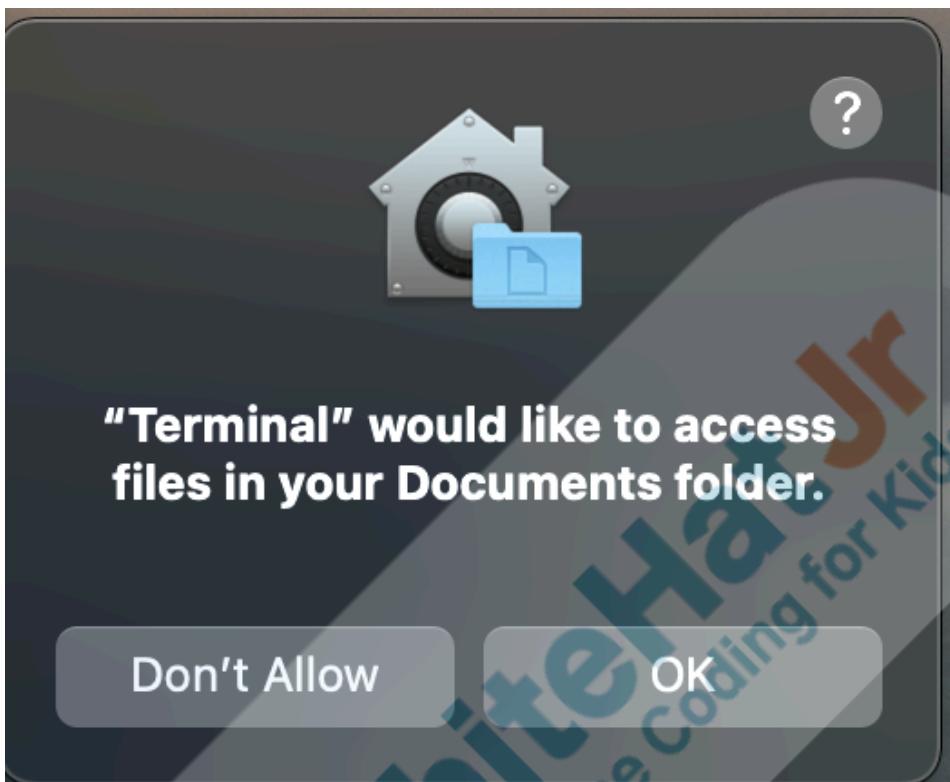
```
omkarshinde ~ % cd Documents/cloud/
```

5. Now, activate blockchain environment as **conda activate cloud**



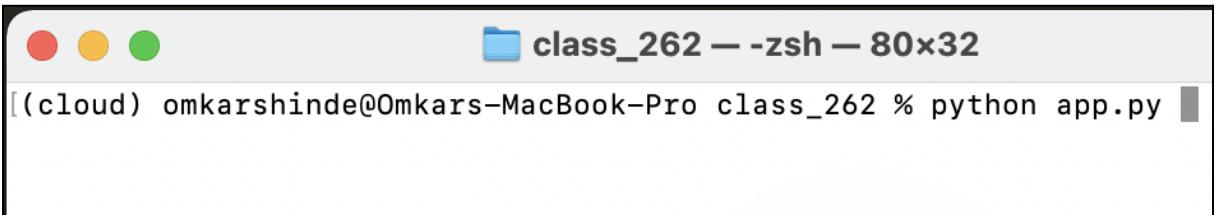
```
cloud ~ % cd Documents/cloud  
cloud ~ % conda activate cloud  
(cloud) omkarshinde ~ %
```

6. If you see this window please click on 'OK.'



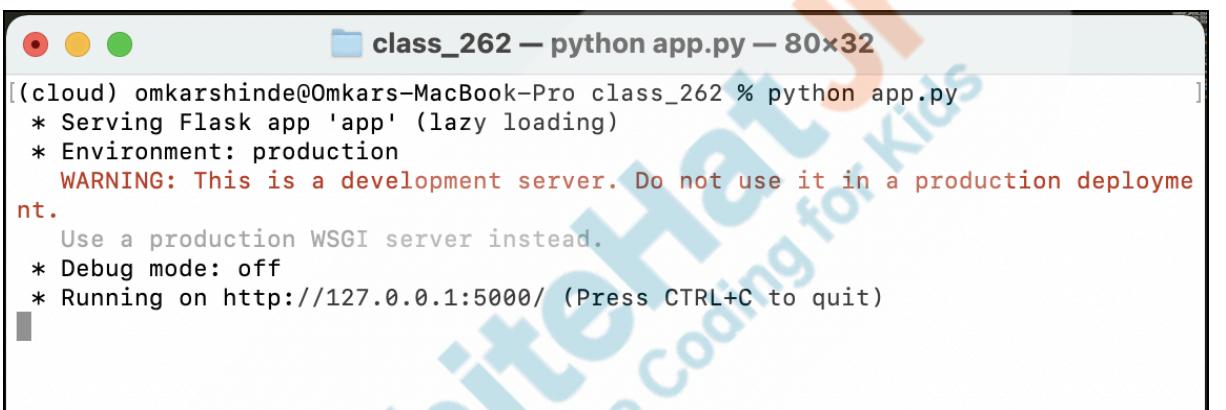
7. And then go to **class_262** folder by `cd class_262`

8. Now, run the Python file as **python app.py**



```
(cloud) omkarshinde@Omkars-MacBook-Pro class_262 % python app.py
```

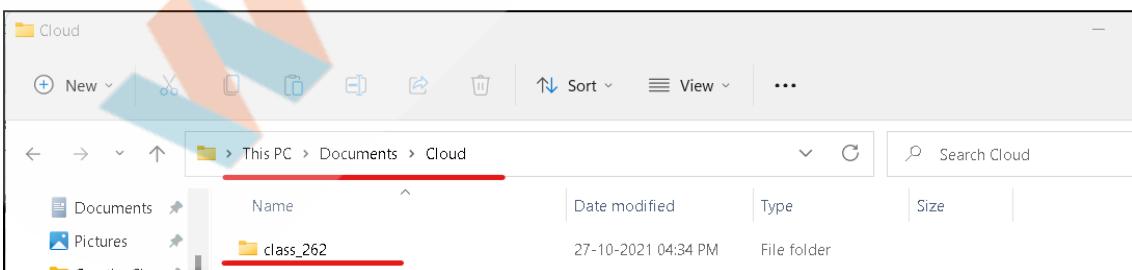
9. It will look like this:



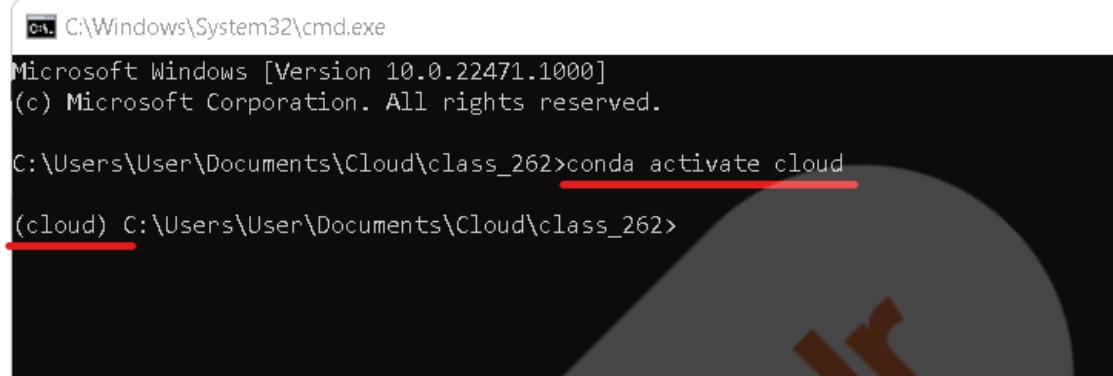
```
(cloud) omkarshinde@Omkars-MacBook-Pro class_262 % python app.py
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Testing the code for Windows:

1. Open the **CMD** in the **class_262** folder which is inside the **Cloud** folder which is located in the **Document/Cloud** directory.



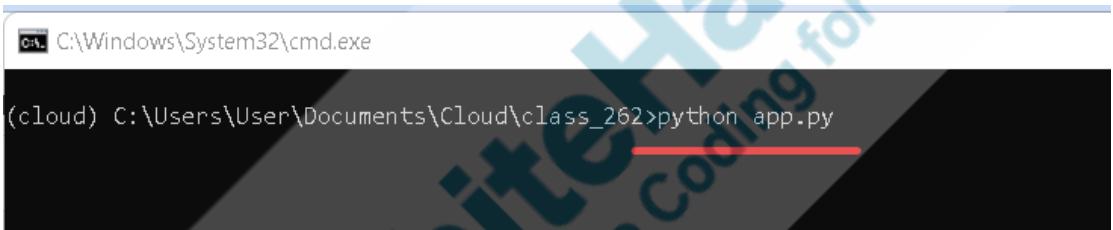
2. Now, active the cloud environment by using the command as **conda activate cloud**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22471.1000]
(c) Microsoft Corporation. All rights reserved.

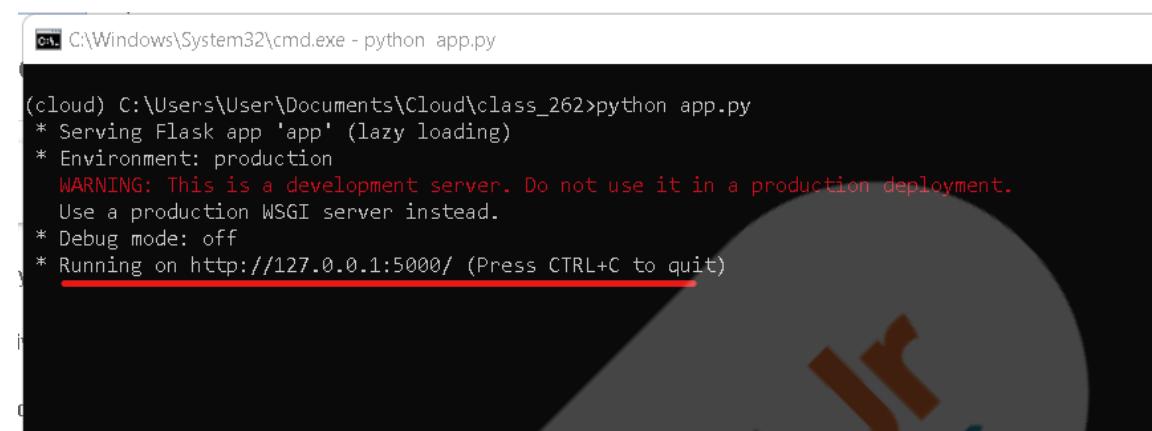
C:\Users\User\Documents\Cloud\class_262>conda activate cloud
(cld) C:\Users\User\Documents\Cloud\class_262>
```

3. Now, run the Python file as **python app.py**



```
C:\Windows\System32\cmd.exe
(cld) C:\Users\User\Documents\Cloud\class_262>python app.py
```

4. It will look like this:



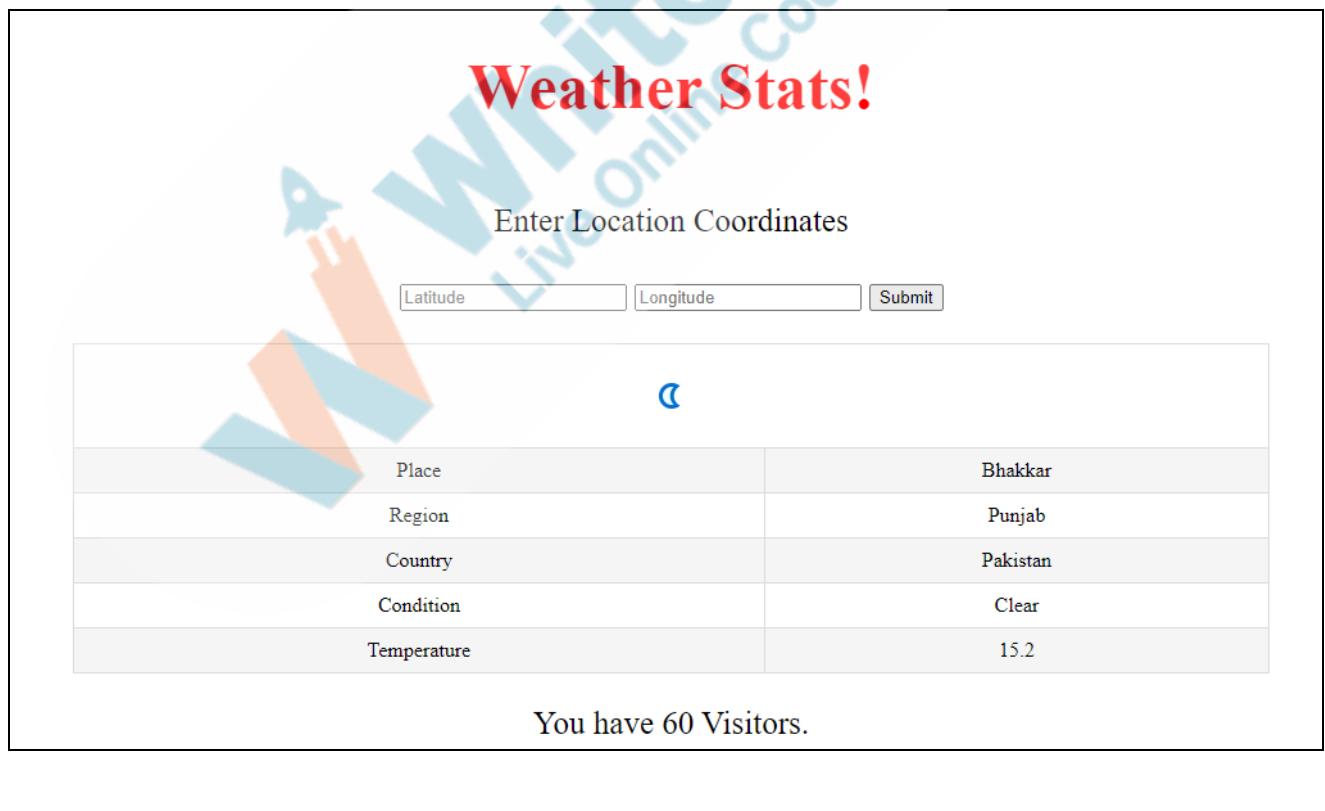
```
C:\Windows\System32\cmd.exe - python app.py

(cloud) C:\Users\User\Documents\Cloud\class_262>python app.py
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Same for both Mac and Windows:

Now, open your browser and paste this address: **127.0.0.1:5000**

You will see something like this:



Weather Stats!

Enter Location Coordinates

Latitude Longitude Submit

Place	Bhakkar
Region	Punjab
Country	Pakistan
Condition	Clear
Temperature	15.2

You have 60 Visitors.

Perfect.

As you remember in the last class, we have created a folder named **class_262**. And in that folder, we have stored all the required files in order to run our web app.

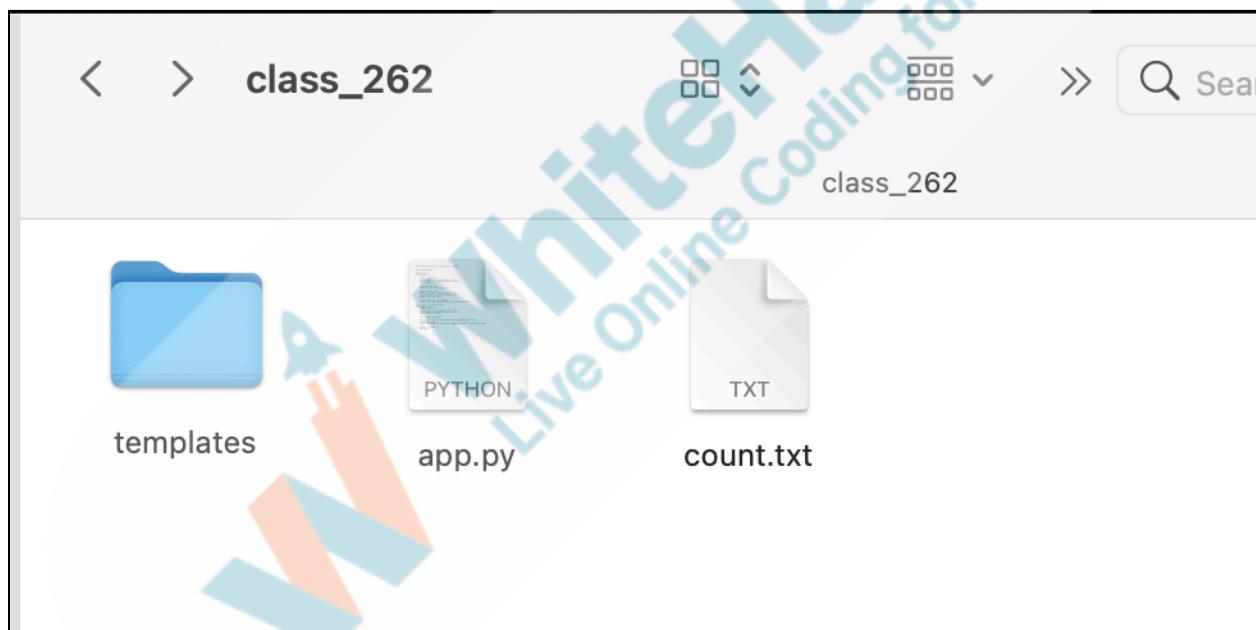
But as you noticed in the last class we have run our web app locally.

But in this class, our goal is to deploy the app on the cloud so anyone can see our web app.

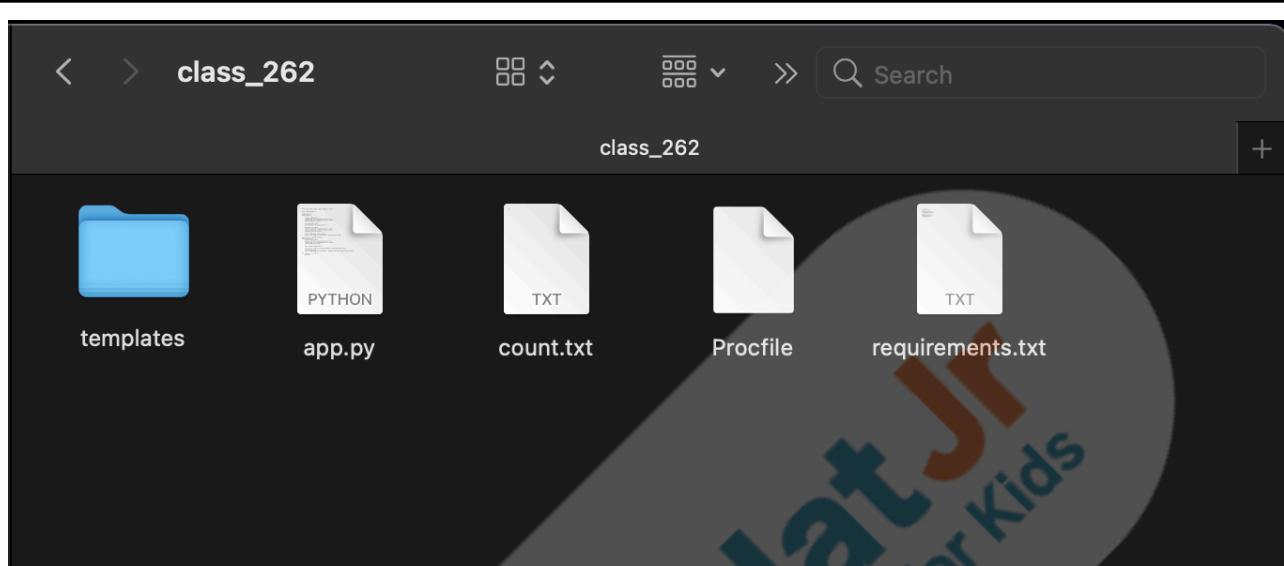
For this, we need 2 additional files which are **Procfile** and **requirements.txt** and we have to move them inside the **class_262** folder.

For me, these 2 files are in **Teacher Activity 1** and for you, it will be in **Student Activity 1**.

So, **class_262** folder will look something like this,

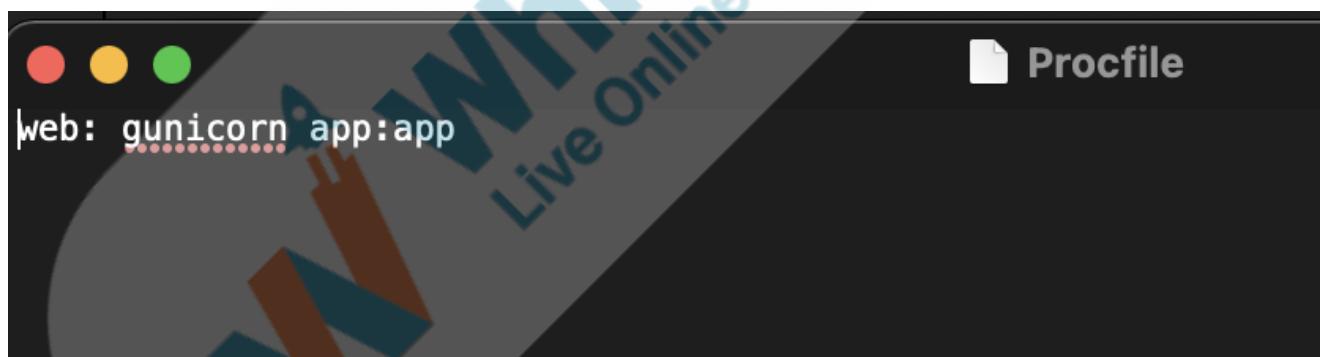


But after moving **Procfile** and **requirement.txt** file you can see something like this,



Procfile - This file is used to deploy web apps on **Render**, later in the class we will see about Render.

In this file, we mention which file should run when a web app is deployed on Render. This file tells Render which file to execute in order to run the app on the Render platform.



Here:

The **web** defines a web application.

The **gunicorn** is a small Python server which is required to deploy apps on Render.

The **app** is nothing but our Python file name where all our Python code is stored. If we had some other file name then that name would be here.

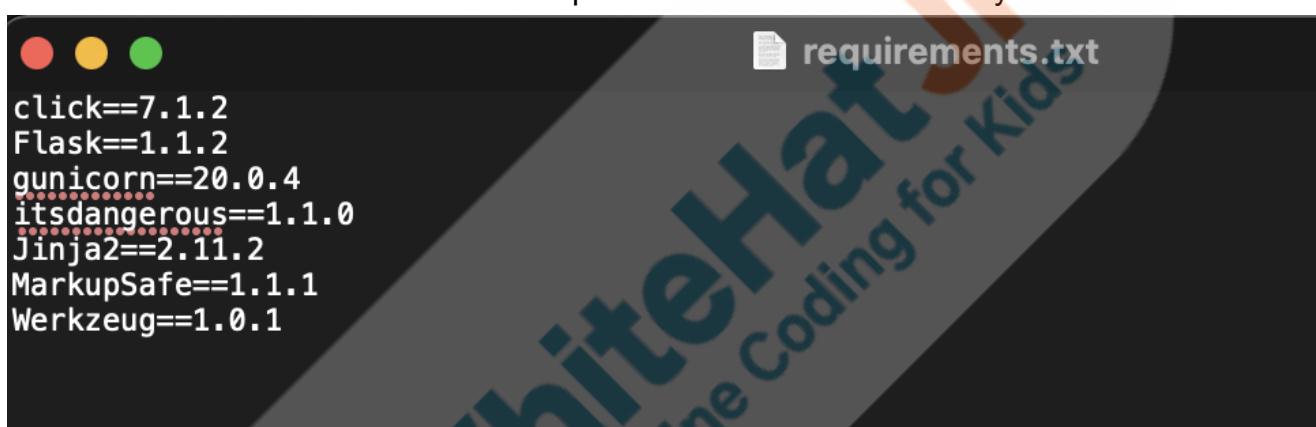
requirements.txt - As the name suggests in this file, we mention the libraries which will be

used for the deployment of the web app. So, if you want to use a new library then you need to mention it in this file.

For example, for running our Python code: first, we create an environment and then install libraries in it such as flask and then run our code.

Similarly, when we wish to deploy the web app on the cloud we need to mention all the used libraries in the **requirements.txt** file.

So, when you deploy the web app, the cloud platform will look into this **requirements.txt** file and install all the mentioned libraries as per the versions of each library.



```
click==7.1.2
Flask==1.1.2
gunicorn==20.0.4
itsdangerous==1.1.0
Jinja2==2.11.2
MarkupSafe==1.1.1
Werkzeug==1.0.1
```

Here, you can see we are using **click**, **Flask**, **gunicorn**, **itsdangerous**, **Jinja2**, **MarkupSafe** and **Werkzeug** libraries.

In our project, we are using the above libraries. So, you can say that the **requirements.txt** file holds the libraries name and its version required to run the program.

So, now our file structure is ready and all set to get deployed.

So, what do you mean by **deployment**?

Well, deployment is nothing but uploading a project on a server that will run our project over the internet. This means once you deploy anything on a server you don't need to think about anything deployment platform can take care to run your project.

We have multiple deployment platforms such as GitHub, Render, etc.

In today's class, we will be deploying our web app using platforms like GitHub and Render.

NOTE: Guide the student to upload all the code and assets files on GitHub and Render. And parallelly explain the concepts as per the lesson plan.

Teacher Stops Screen Share

Now it is your turn.

- Ask the Student to press the ESC key to come back to the panel.
- Guide the Student to start Screen Share.
- The Teacher gets into Fullscreen.

**Step 3:
Student-Led
Activity
(30 mins)**



First you have to download the **Procfile** and **requirements.txt** files from [Student Activity 1](#) and put them in the **class_262** folder in which you were working.

[**Student Activity 1- PREDEFINED FILES**](#)

[**Student Activity 2 - GITHUB**](#)

[**Student Activity 3 - GIT DOWNLOAD**](#)

[**Student Activity 4 - RENDER**](#)

[**Student Activity 5 - RENDER DOCUMENTATION**](#)

[**Student Activity 6 - CODE DIAGRAM**](#)

GITHUB

GitHub, Inc. is a company that provides a hosting platform. To host applications on GitHub we required a software named **Git**.

Git is nothing but software that helps you to deploy your project on GitHub.

It has multiple functionalities to manage and update your project in the git repository. Once you upload your project on GitHub you can modify it anytime and from anywhere.

In webdev phase, you have already used GitHub for hosting your websites.

In this class, we are going to upload our web app on GitHub. For that first, you need to create an account in GitHub if you haven't created one.

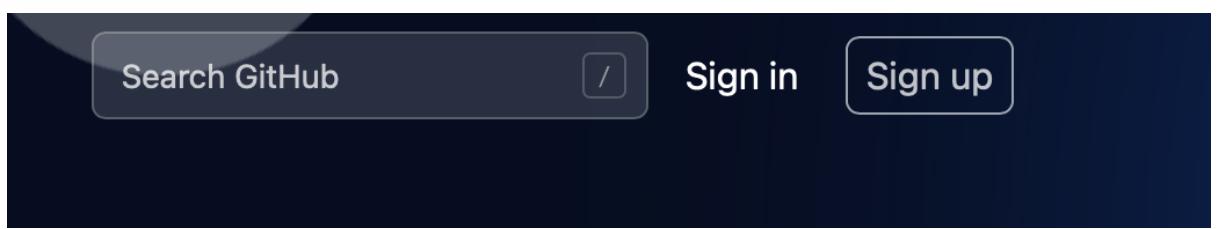
NOTE: If you faced any issues while deploying the project on GitHub or Render, please follow Teacher Reference Activity 2.

Create an account on GitHub:

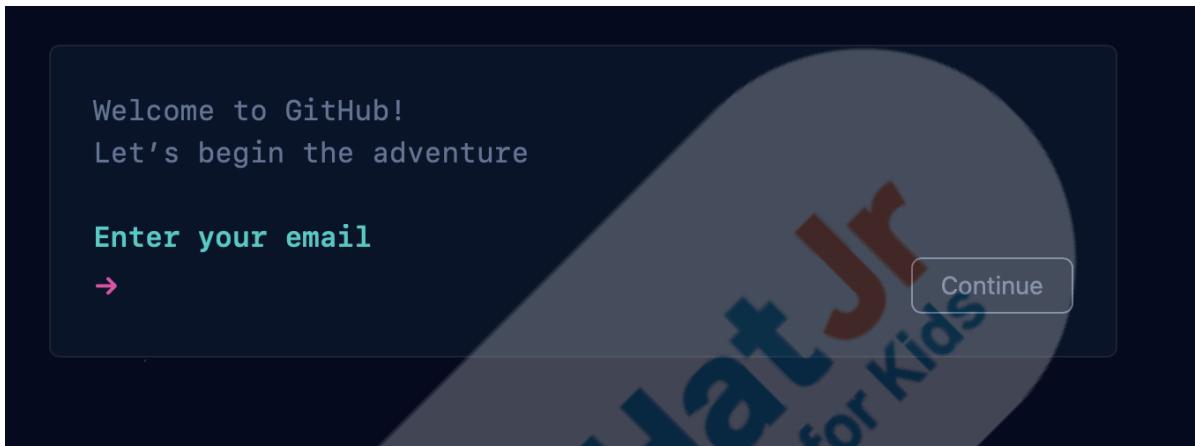
1. Open [Student Activity 2](#) and go to GitHub. You can see something like this:



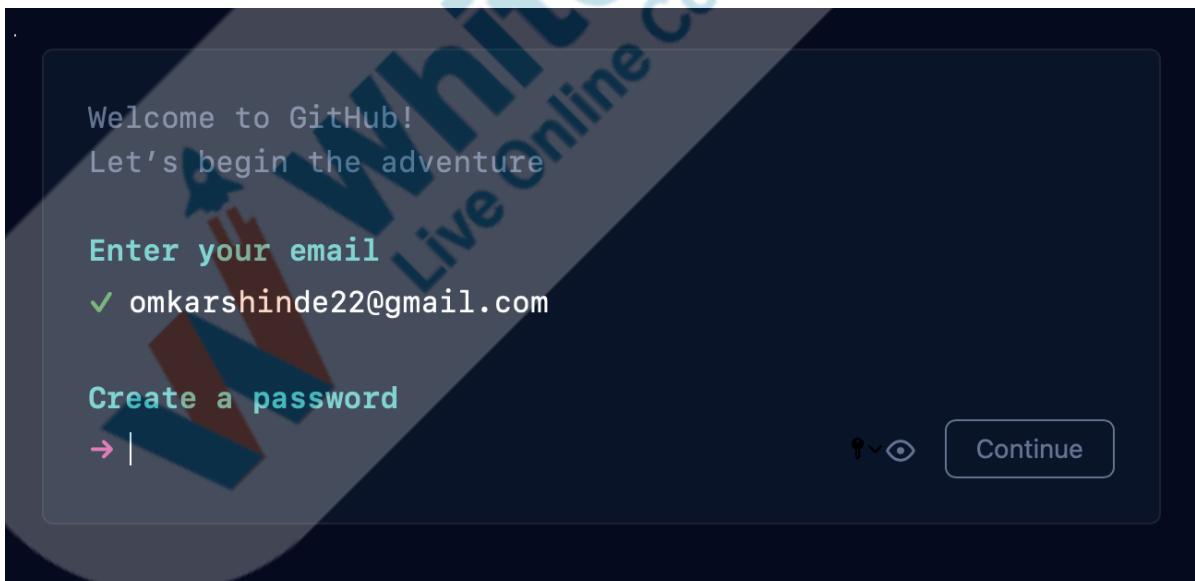
2. Now, click on Sign up.



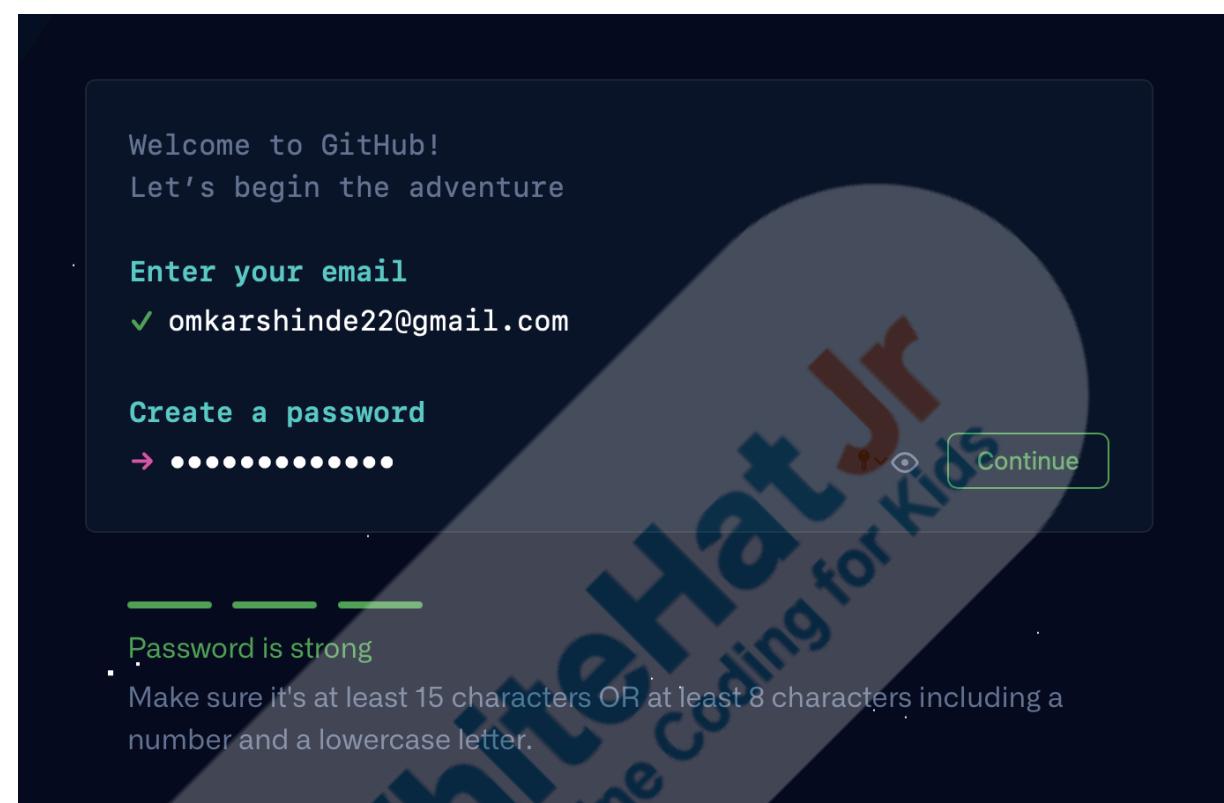
3. Enter the email address which you want to set for your GitHub account. And click on continue.



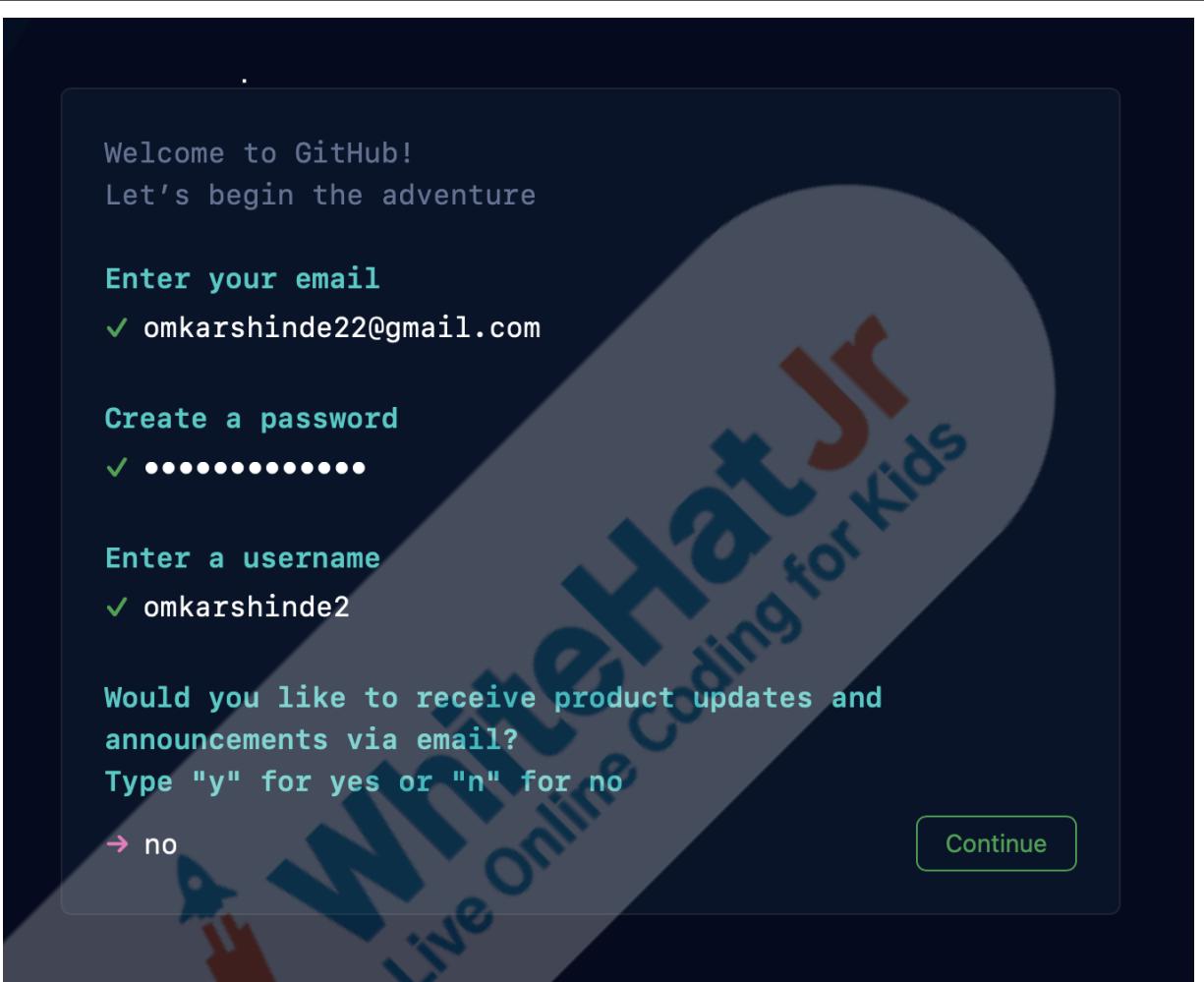
4. Now, enter your desired password for the GitHub account.



5. Now, click on continue.



6. Now, enter the username for your GitHub account.
7. And write NO for 'would you like to receive product updates and announcements via email?' If you want updates from GitHub you can write YES. It's up to you.



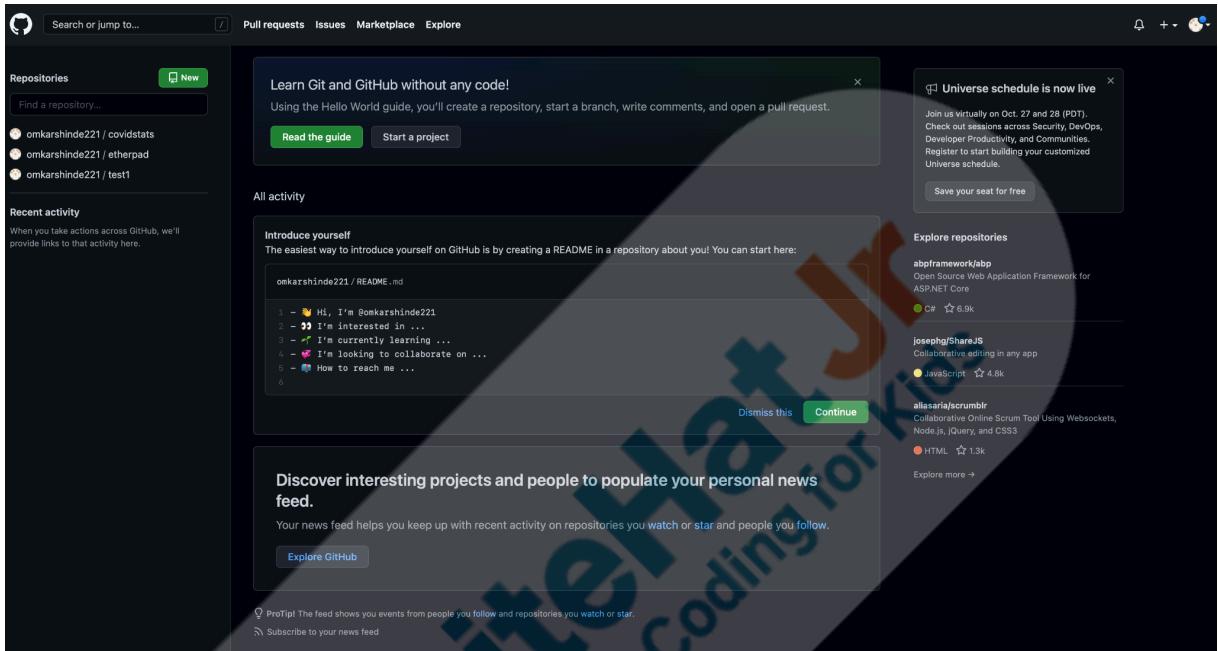
8. Now, verify your account by doing the following task. Such as choosing a spiral galaxy.



9. Great. Almost there. Now, click on create an account.

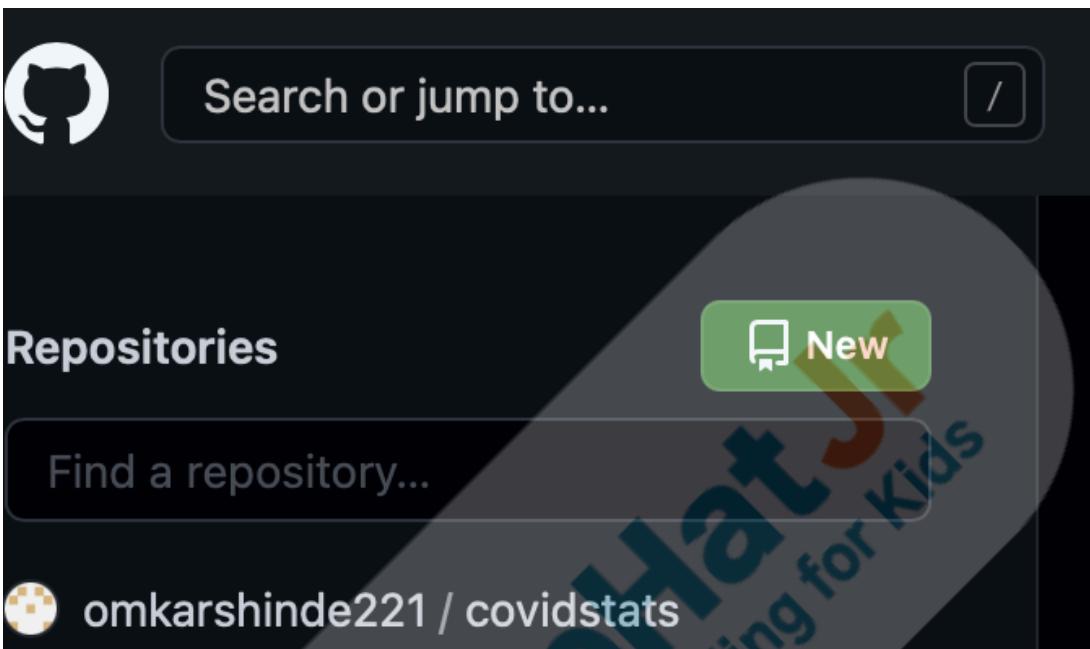


10. Once you click on create an account. You can see something like this.



11. Let's create your first GitHub repository. The repository is nothing but a place where you upload and store code files and assets of a specific project.

- To create a repository you need to click on the  button which is in green.



12. Then, you can see something like this:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner *  omkarshinde221 / **Repository name ***

Great repository names are short and memorable. Need inspiration? How about [turbo-journey?](#)

Description (optional)

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

13. Now, here you see the **Repository name**.

Repository name *

myfirstapp

Put '**myfirstapp**' as your Repository name. You can name anything as per your choice but for today's class use the name as **myfirstapp**.

14. Then, click on the **Create repository** button.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner *  **Repository name ***

omkarshinde221 / myfirstapp 

Great repository names are short and inspiring! [myfirstapp is available.](#) Need inspiration? How about [turbo-journey?](#)

Description (optional)

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

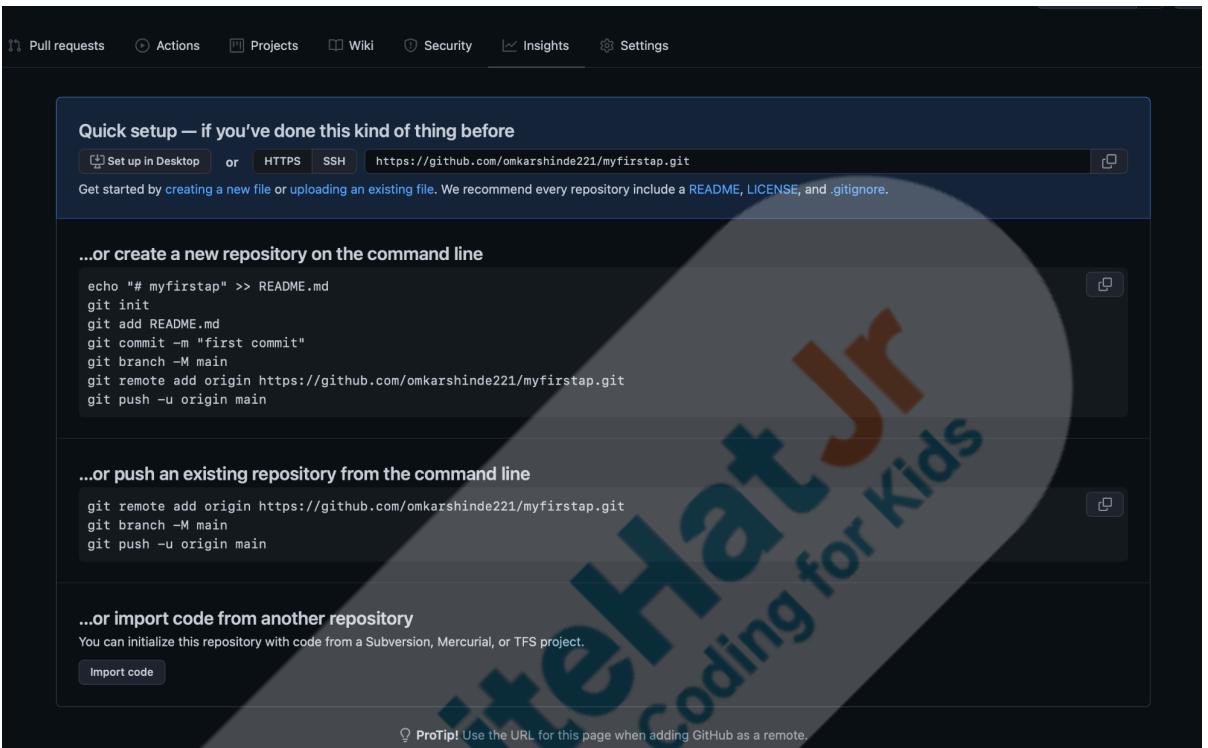
Add a README file
This is where you can write a long description for your project. [Learn more](#).

Add .gitignore
Choose which files not to track from a list of templates. [Learn more](#).

Choose a license
A license tells others what they can and can't do with your code. [Learn more](#).

Create repository

15. Then, you can see something like this:



Quick setup — if you've done this kind of thing before

[Set up in Desktop] or [HTTPS] [SSH] <https://github.com/omkarshinde221/myfirstap.git>

Get started by creating a new file or uploading an existing file. We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# myfirstap" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/omkarshinde221/myfirstap.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/omkarshinde221/myfirstap.git
git branch -M main
git push -u origin main
```

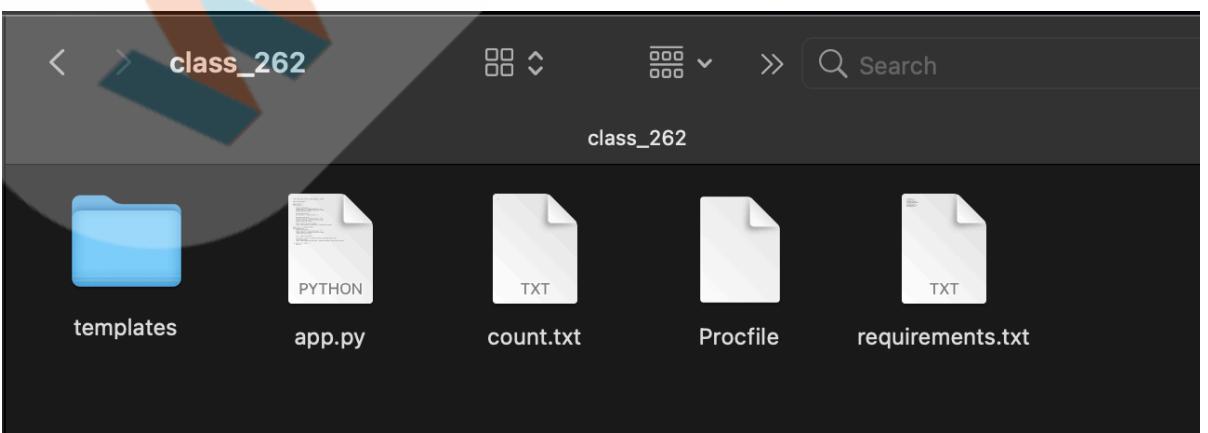
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

ProTip! Use the URL for this page when adding GitHub as a remote.

16. Congratulations. You have successfully created your GitHub repository. Let's add our web app to this repository.
17. Keep this webpage open in the browser. Do not close it.
18. Now, go to the **Class_262** folder which is stored in the **Documents\cloud\class_262**. You can see something like this.



19. Now, our goal is to upload this web project to the GitHub repository created.
20. But for that, we need to download **Git**. Using git we can upload our web app on the GitHub website.
21. Open **Student Activity 3** you can see something like this:



Downloads

macOS **Windows**

Linux/Unix

Older releases are available and the [Git source repository](#) is on GitHub.

GUI Clients
 Git comes with built-in GUI tools ([git-gui](#), [gitk](#)), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

Logos
 Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

Git via Git
 If you already have Git installed, you can get the latest development version via Git itself:

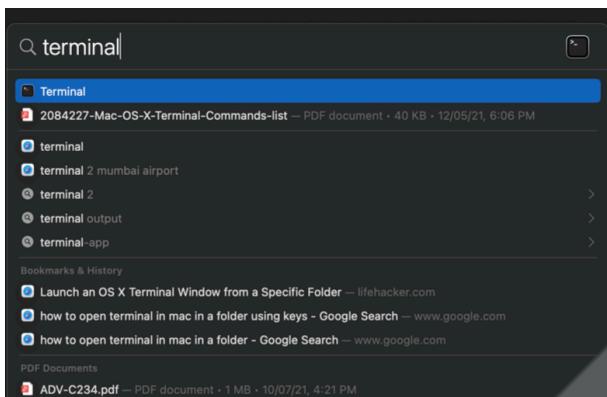
```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the [web interface](#).

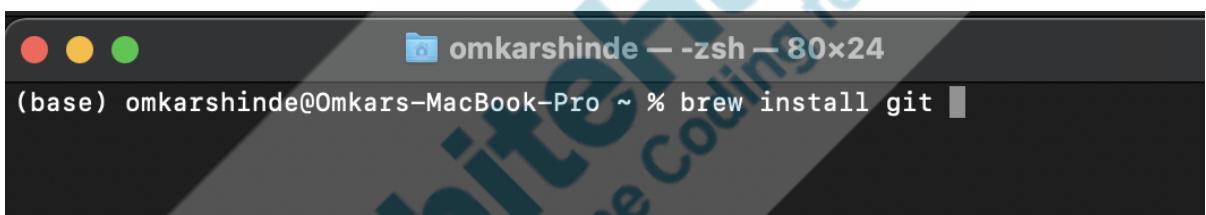
22. Then, choose your operating system.

FOR MAC installing Git:

- Open the terminal by pressing **command + SPACEBAR** key and type '**terminal**'.



- Then, write the below command to install git in your MAC.



- It will install **git** in your MAC and you can see something like this:

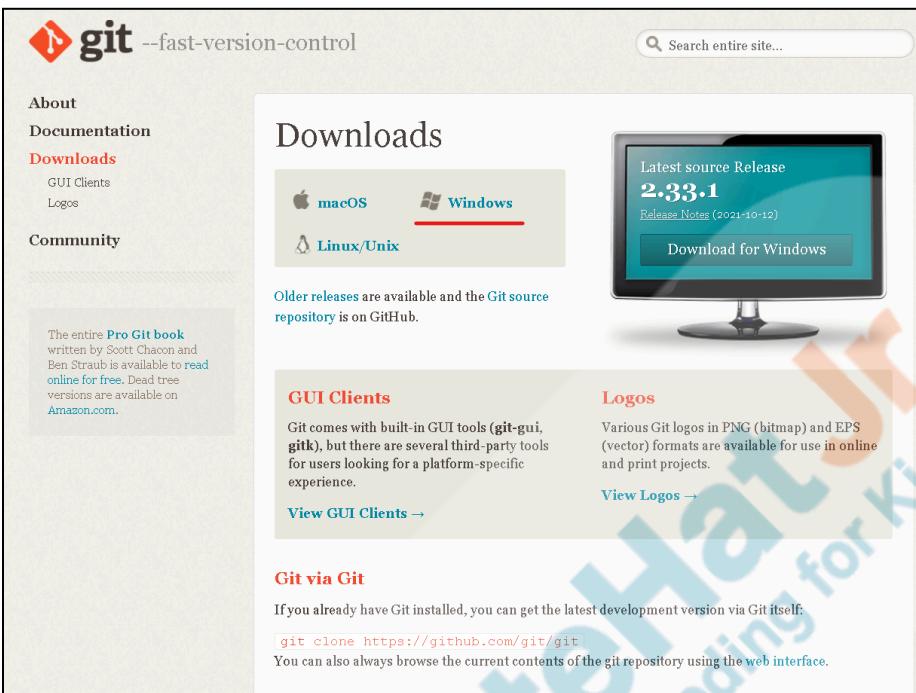
```
[base) omkarshinde@Omkars-MacBook-Pro ~ % brew install git
Updating Homebrew...
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> New Formulae
colima      fheroes2      ly          qwt-qt5      tom11
cpufetch    liqoctl      mmtabbarview  rbw
==> Updated Formulae
Updated 547 formulae.
==> Renamed Formulae
selenium-server-standalone -> selenium-server

==> Downloading https://ghcr.io/v2/homebrew/core/gettext/manifests/0.21
#####
==> Downloading https://ghcr.io/v2/homebrew/core/gettext/blobs/sha256:a025e143fe
#####
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:a025e143fe
#####
==> Downloading https://ghcr.io/v2/homebrew/core/pcre2/manifests/10.38
#####
==> Downloading https://ghcr.io/v2/homebrew/core/pcre2/blobs/sha256:c2f04108058a
#####
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:c2f04108058a
##### 100.0%
```

- Once it's done then it means now you have git installed in your system.

For WINDOWS installing Git:

- Click on Windows to download git for windows OS.



The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads

[macOS](#) [Windows](#) [Linux/Unix](#)

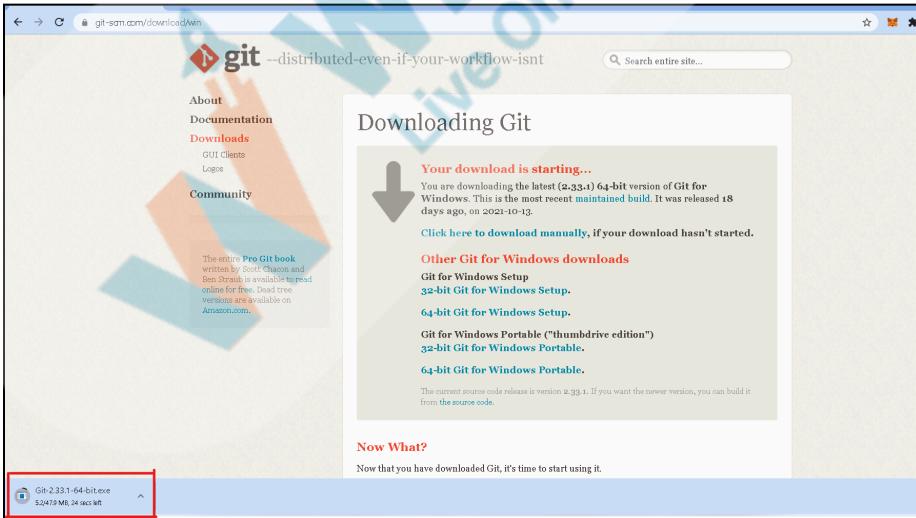
Older releases are available and the Git source repository is on GitHub.

GUI Clients
Git comes with built-in GUI tools (`git-gui`, `gitk`), but there are several third-party tools for users looking for a platform-specific experience.
[View GUI Clients →](#)

Logos
Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.
[View Logos →](#)

Git via Git
If you already have Git installed, you can get the latest development version via Git itself.
`git clone https://github.com/git/git`
You can also always browse the current contents of the git repository using the [web interface](#).

2. Downloading will start automatically.



Your download is starting...
You are downloading the latest (2.33.1) 64-bit version of Git for Windows. This is the most recent [maintained build](#). It was released 18 days ago, on 2021-10-13.

Click here to [download manually](#), if your download hasn't started.

Other Git for Windows downloads
[Git for Windows Setup](#)
[32-bit Git for Windows Setup](#).
[64-bit Git for Windows Setup](#).
[Git for Windows Portable \("thimbdive edition"\)](#)
[32-bit Git for Windows Portable](#).
[64-bit Git for Windows Portable](#).

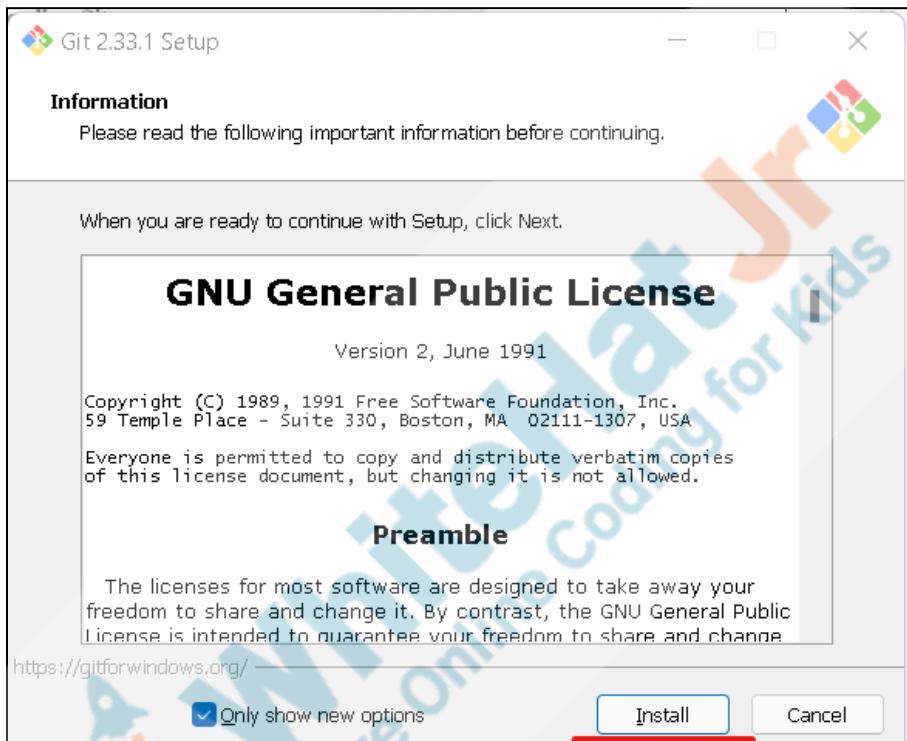
The current source code release is version 2.33.1. If you want the newer version, you can build it from the [source code](#).

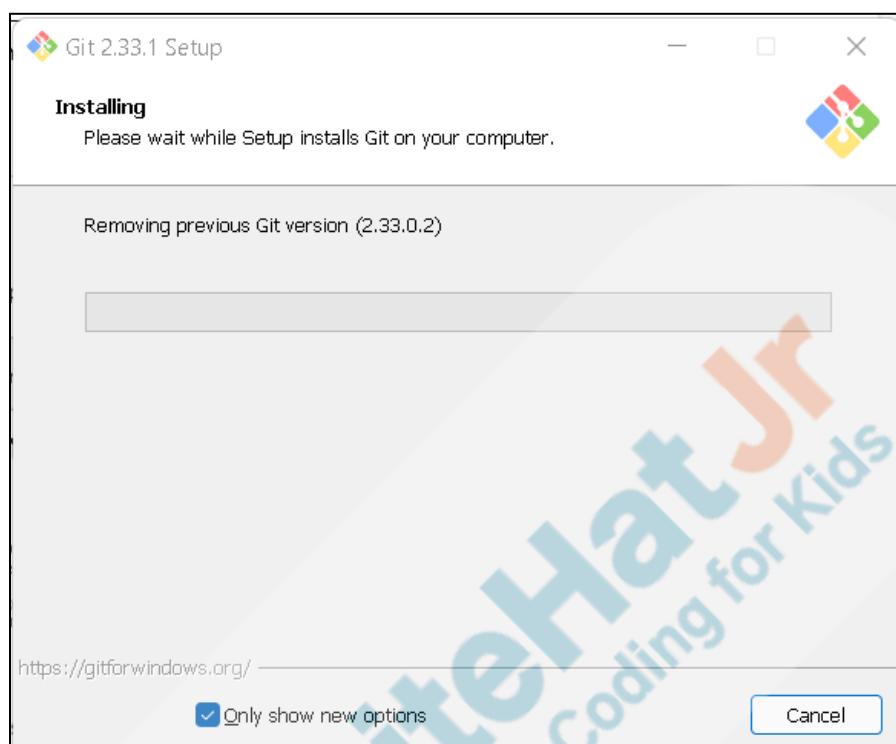
Now What?
Now that you have downloaded Git, it's time to start using it.

3. Now, double-click on the downloaded Git application and follow the below

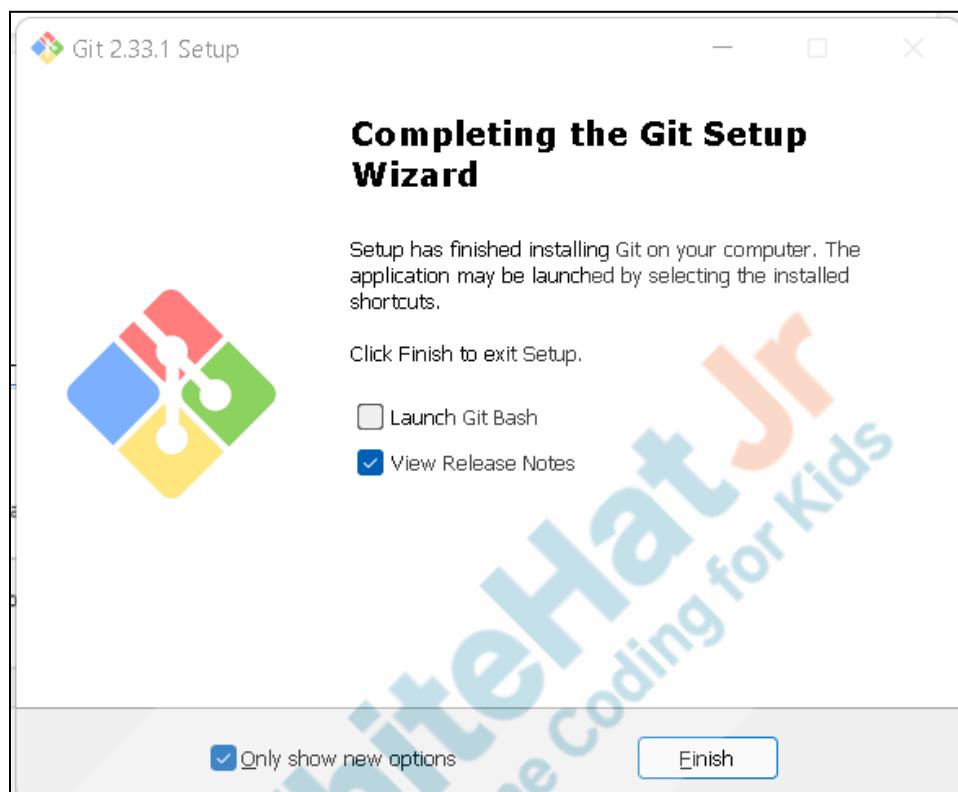
mentioned steps:

- Click on “Install.”





- b. Click on **Finish**, now Git is successfully installed.

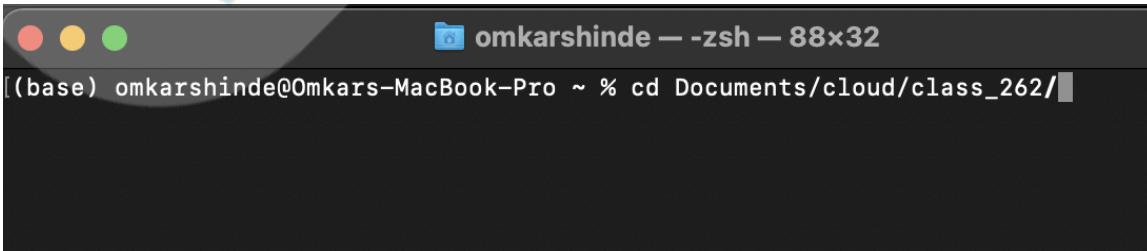


NOTE - After installing git you can continue following the steps for MAC and WINDOWS users for uploading files on GitHub.

23. Let's upload our web app on GitHub. To do that, open the terminal and go to the **class_262** folder in the terminal.

Use the following command to direct your terminal to **class_262** folder

cd Documents/cloud/class_262/



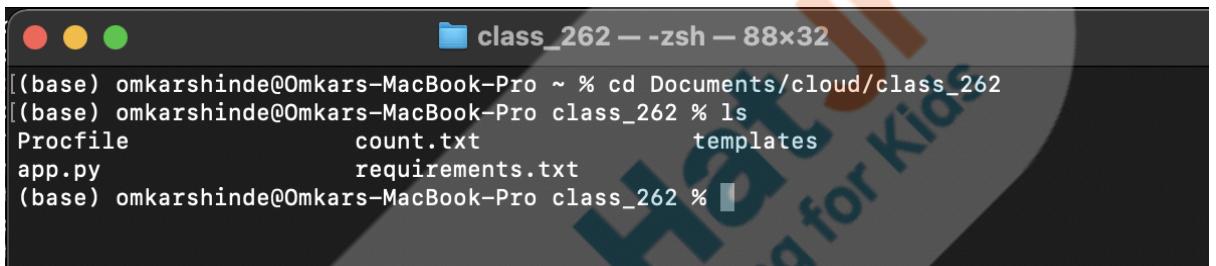
```
omkarshinde ~ % cd Documents/cloud/class_262/
```

24. Now, use the ‘ls’ command to see how many files are there in the **class_262** folder.



```
class_262 — zsh — 88x32
[(base) omkarshinde@Omkars-MacBook-Pro ~ % cd Documents/cloud/class_262
[(base) omkarshinde@Omkars-MacBook-Pro class_262 % ls
```

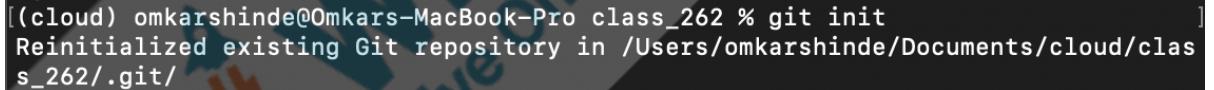
Output:



```
class_262 — zsh — 88x32
[(base) omkarshinde@Omkars-MacBook-Pro ~ % cd Documents/cloud/class_262
[(base) omkarshinde@Omkars-MacBook-Pro class_262 % ls
Procfile           count.txt          templates
app.py             requirements.txt
(base) omkarshinde@Omkars-MacBook-Pro class_262 %
```

25. It is time to initialize git in the **class_262** folder. This will initialise git in the project folder to upload folder files on GitHub.

git init



```
[(cloud) omkarshinde@Omkars-MacBook-Pro class_262 % git init
Reinitialized existing Git repository in /Users/omkarshinde/Documents/cloud/class_262/.git/
```

26. Now, let's add **class_262** folder files in GitHub. By using the following command,
git add .



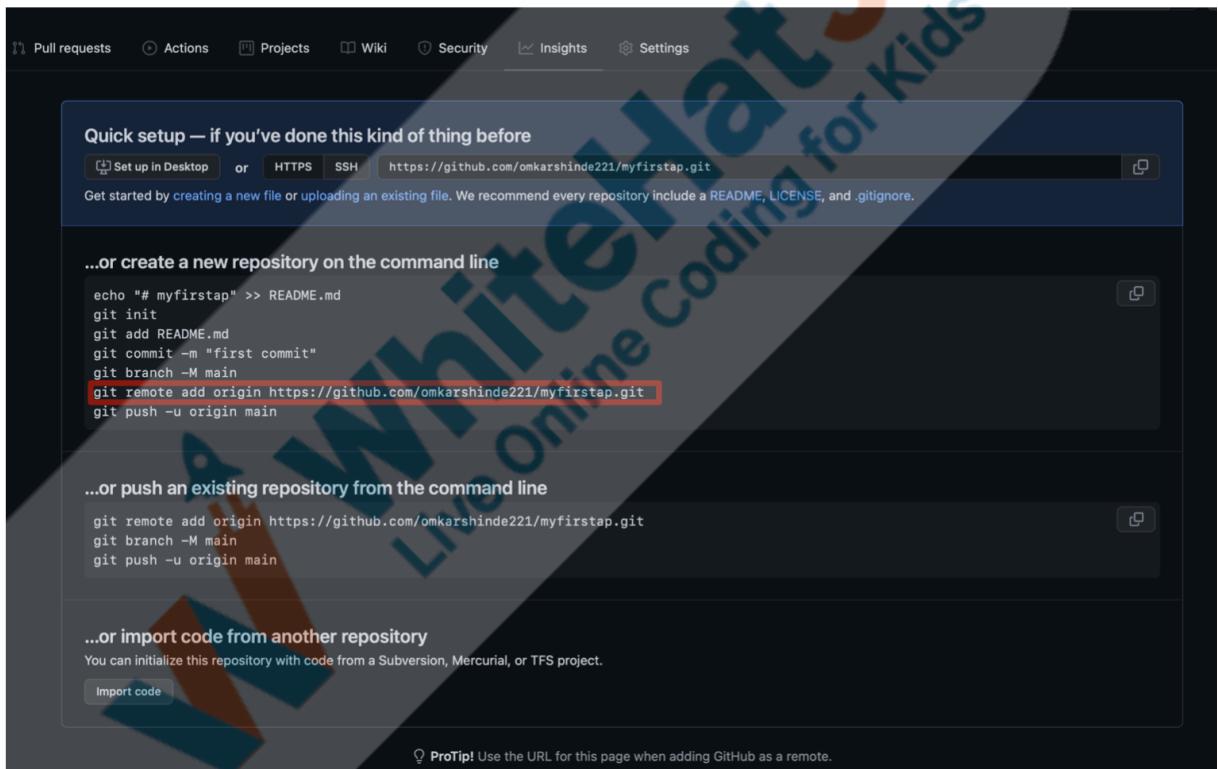
```
[(cloud) omkarshinde@Omkars-MacBook-Pro class_262 % git add .
```

27. Now, use the **commit** command to commit our first uploading files to GitHub.

git commit -m “first commit”

```
[(cloud) omkarshinde@Omkars-MacBook-Pro class_262 % git commit -m 'first commit'
[master (root-commit) 989dcc7] first commit
 5 files changed, 85 insertions(+)
 create mode 100644 .DS_Store
 create mode 100644 app.py
 create mode 100644 class_263.py
 create mode 100644 count.txt
 create mode 100644 templates/index.html
```

28. Now, to get the next command open a GitHub account in the browser. And copy the 6th command.



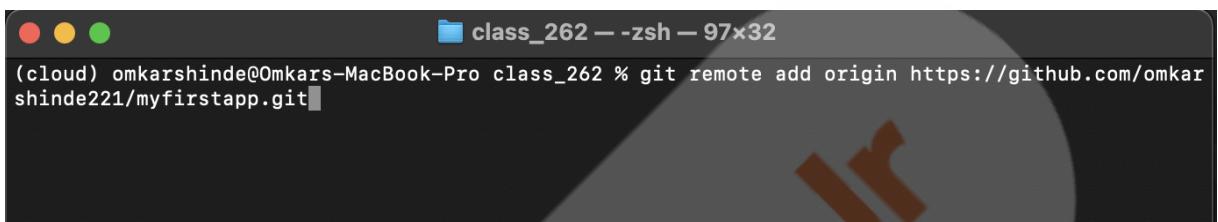
29. Put the above copied URL and the below command.

git remote add origin <URL>

NOTE: This command is different for everyone. So make sure you are copying the command correctly else it will give you an error.

```
git remote add origin https://github.com/omkarshinde221/myfirstapp.git
```

30. Just like this, paste the command in the terminal and hit enter.



```
(cloud) omkarshinde@Omkars-MacBook-Pro class_262 % git remote add origin https://github.com/omkarshinde221/myfirstapp.git
```

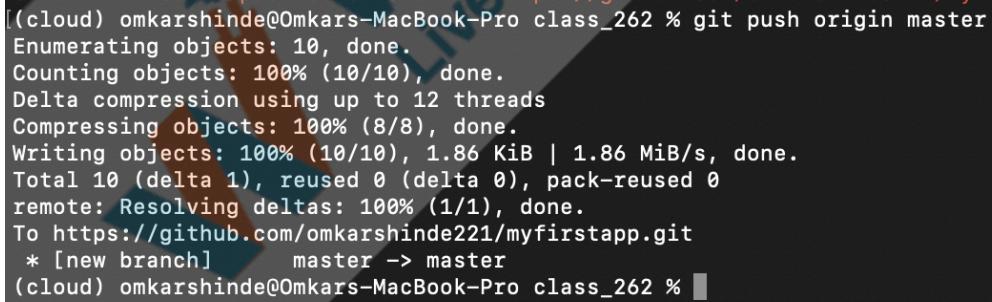
31. Once you hit enter. It will show nothing, it means the command is successful.



```
[(cloud) omkarshinde@Omkars-MacBook-Pro class_262 % git remote add origin https://github.com/omkarshinde221/myfirstapp.git  
(cloud) omkarshinde@Omkars-MacBook-Pro class_262 %
```

32. Now, it's time to push the files on GitHub. For that follow the below command and hit enter.

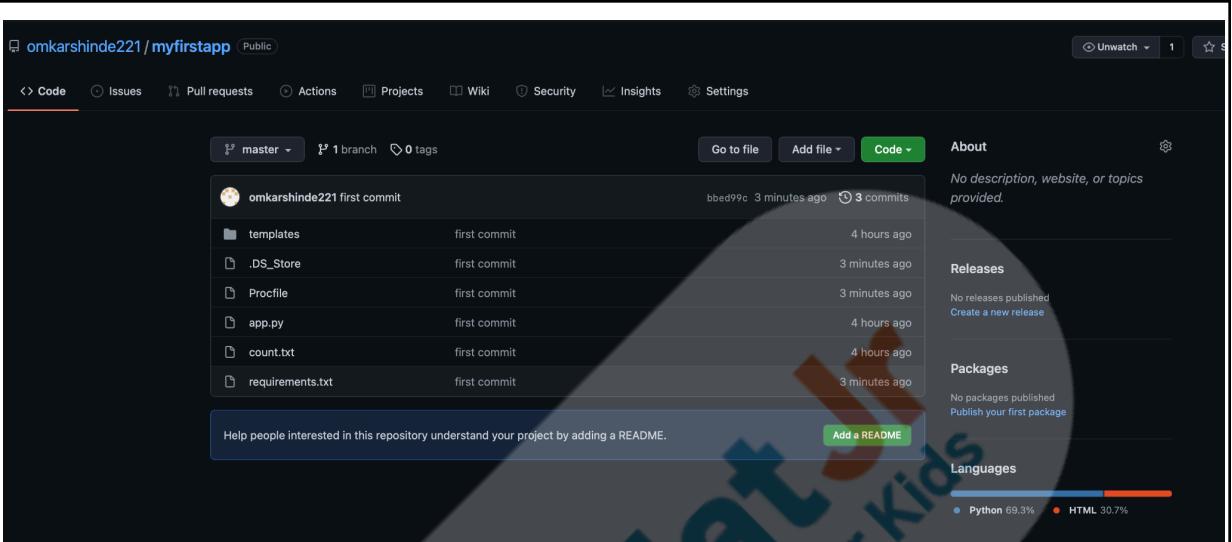
git push origin master



```
[(cloud) omkarshinde@Omkars-MacBook-Pro class_262 % git push origin master  
Enumerating objects: 10, done.  
Counting objects: 100% (10/10), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (8/8), done.  
Writing objects: 100% (10/10), 1.86 KiB | 1.86 MiB/s, done.  
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0  
remote: Resolving deltas: 100% (1/1), done.  
To https://github.com/omkarshinde221/myfirstapp.git  
 * [new branch] master -> master  
(cloud) omkarshinde@Omkars-MacBook-Pro class_262 %
```

NOTE: It may ask you to sign in. Then, do sign in and Authorize access to GitHub. It will upload your code file and assets on GitHub.

33. Now, go to the repository you created and refresh the page you can see that our files have been uploaded on GitHub.



Perfect, till now we have completed 50% of the deployment process by uploading all code files and assets on GitHub.

Now, let's move to complete the rest of the 50% process of deployment by working on Render.

Now, you will ask why we upload code files and all assets on GitHub.

Well, it's simple, GitHub works as a cloud that stores our web app files. And Render on the other hand will be our cloud platform that will execute our web app on the internet.

render

Now upload your webapp i.e weatherstats on cloud so that anyone can access it from anywhere anytime.

1. Now to Upload and host your website kindly follow the steps mentioned **Teacher Reference Activity-3**.
2. You can see your app is running on the internet using Render.

Weather Stats!

Enter Location Coordinates

🕒	
Place	Bhakkar
Region	Punjab
Country	Pakistan
Condition	Clear
Temperature	15.2

You have 60 Visitors.

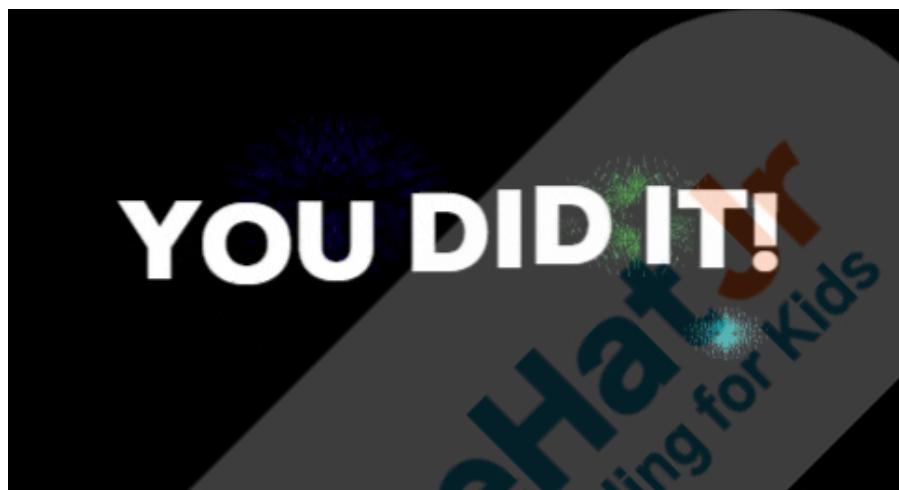
Weather Stats!

Enter Location Coordinates

🕒	
Place	Bhakkar
Region	Punjab
Country	Pakistan
Condition	Clear
Temperature	15.2

You have 60 Visitors.

Congratulations, You have done it. You have successfully created a web app and deployed it on the internet. That's great.



Submit your hosted web app link in the student panel.

Teacher Guides Student to Stop Screen Share

**Step 4:
Wrap-Up
(5 mins)**

You did great today as well.
Great! You have two hats off.

(Give at least 2 hats off)
Press the **Hats Off** Icon for
Creatively Solving Activities.



Q What is **Git**?

A Git is nothing but software that helps you to upload your project on GitHub.

Press the **Hats Off** Icon for a **Great Question**.



Press the **Hats Off** Icon for

	<p>Q What is Render?</p> <p>A Render gives web applications a place to run over the Internet.</p>	<p>Strong Concentration</p>  <p>If you don't have time to perform additional activities, ask the student to perform all the additional activities after the class. Additional activities are VERY important for kids so that they are ready for the next module. And some challenging concepts are coming ahead.</p> <p>Also, remind the student to refer to the Student Reference activity for increasing his/her knowledge. This should also be done.</p>
Teacher Initiates Screen Share		
Step 5: Project Pointers and Cues	DEPLOYING MATH WEBAPP Goals of the Project:	Open the Project Solution link and showcase the output.

(5 mins)

In today's class we learned what is GitHub, what is Render and how to deploy our apps on both of these platforms:

DEPLOYING MATH WEBAPP:

This project is the continuation of the previous project C262, in this, you will be adding the style to the web page and deploy the app on Render and GitHub.

Story:

Global Maths Academy, the institute for which you have developed an application for generating mathematical expression given using the maths API, for which you had added the GUI using the FLASK. In this project, you have to deploy this application that will make it alive for the users to use it.

Good Luck!

For the solution of all the Additional Activities, open Teacher Activity 7 and navigate to class number C263.

Additional Activity 1 -

Run **Student-Activity-7** from the panel.

The **TASK** and **HINTS** are mentioned on the website itself.

Additional Activity 2 -

Run **Student-Activity-8** from the panel.

The **TASK** and **HINTS** are mentioned on the website itself.

Additional Activity 3 -

Run **Student-Activity-9** from the panel.

The **TASK** and **HINTS** are mentioned on the website itself.

Additional Activity 4 -

Run **Student-Activity-10** from the panel.

The **TASK** and **HINTS** are mentioned on the website itself.

Additional Activity 5 -

Run **Student-Activity-11** from the panel.

The **TASK** and **HINTS** are mentioned on the website itself.

x End Class

Teacher Clicks

Activity	Activity Name	Links
Teacher Activity 1	PREDEFINED FILES	https://drive.google.com/drive/folders/1O9yMT9iVjYaCB-Z5Eeubaphu1N_dpdGk?usp=sharing
Teacher Activity 2	GITHUB	http://github.com/
Teacher Activity 3	GIT DOWNLOAD	https://git-scm.com/downloads
Teacher Activity 4	RENDER	https://render.com/
Teacher Activity 5	RENDER DOCUMENTATION	https://render.com/docs
Teacher Activity 6	CODE DIAGRAM	https://docs.google.com/document/d/e/2PACX-1vQa1jf1zBixVwlRnohGXEflKhq1z78nKwNDWZchxny6nprBKXi8-CztKa1rVTp-PrbRnFR-7_ysb1K5/pub
Teacher Activity 7	ADDITIONAL ACTIVITY	https://mahdihat791.github.io/v2/additional_activities_solution.html
Teacher Reference	RECORD SCREEN	https://curriculum.whitehatjr.com/ADV+Ass

Activity 1	REFERENCE	et/C145+-Guide+to+loom+recording.pdf
Teacher Reference Activity 2	Cloud Troubleshoot	https://docs.google.com/document/d/e/2PA_CX-1vSFIIWI8CTjvcKlqxnUbQv8c26Bo5yZn16UDzH2cE9paOyL9hNpZgq2DwerNh9Q78lpq4s5cqSGf9Hq/pub
Teacher Reference Activity 3	Deploy flask on Render Documentation	https://docs.google.com/document/d/e/2PA_CX-1vQXPZI9cP1t3uGLUT1opFdjjHDCLzKKct7rRndrluBQqyAQo-k2iRxNNavy5OKZVs-QDW3xR5Ak89Hq/pub
Teacher Reference Activity 4	Deployed Link	https://c263-covid.onrender.com/
Student Activity 1	PREDEFINED FILES	https://drive.google.com/drive/folders/1O9yMT9iVjYaCB-Z5Eubaphu1N_dpdGk?usp=sharing
Student Activity 2	GITHUB	http://github.com/
Student Activity 3	GIT DOWNLOAD	https://git-scm.com/downloads
Student Activity 4	RENDER	https://render.com/
Student Activity 5	RENDER DOCUMENTATION	https://render.com/docs
Student Activity 6	CODE DIAGRAM	https://docs.google.com/document/d/e/2PA_CX-1vQa1jfzBixVwlRnohGXEflKhg1z78nKwNDWZchxny6nprBKXi8-CztKa1rVTp-PrbRnFR-7_ysb1K5/pub
Student Reference Activity 1	Deploy flask on Render Documentation	https://docs.google.com/document/d/e/2PA_CX-1vQXPZI9cP1t3uGLUT1opFdjjHDCLzKKct7rRndrluBQqyAQo-k2iRxNNavy5OKZVs-QDW3xR5Ak89Hq/pub
Project solution	DEPLOYING MATH WEBAPP	https://docs.google.com/document/d/e/2PA_CX-1vSikAYHwoIN0s-fuq32IHVyTHYkkE4

		<u>ueBmQN9NZkpQ27om1qW6wG7cKqn82b pi2b0Mj2Kznezr-BJyK/pub</u>
--	--	---

