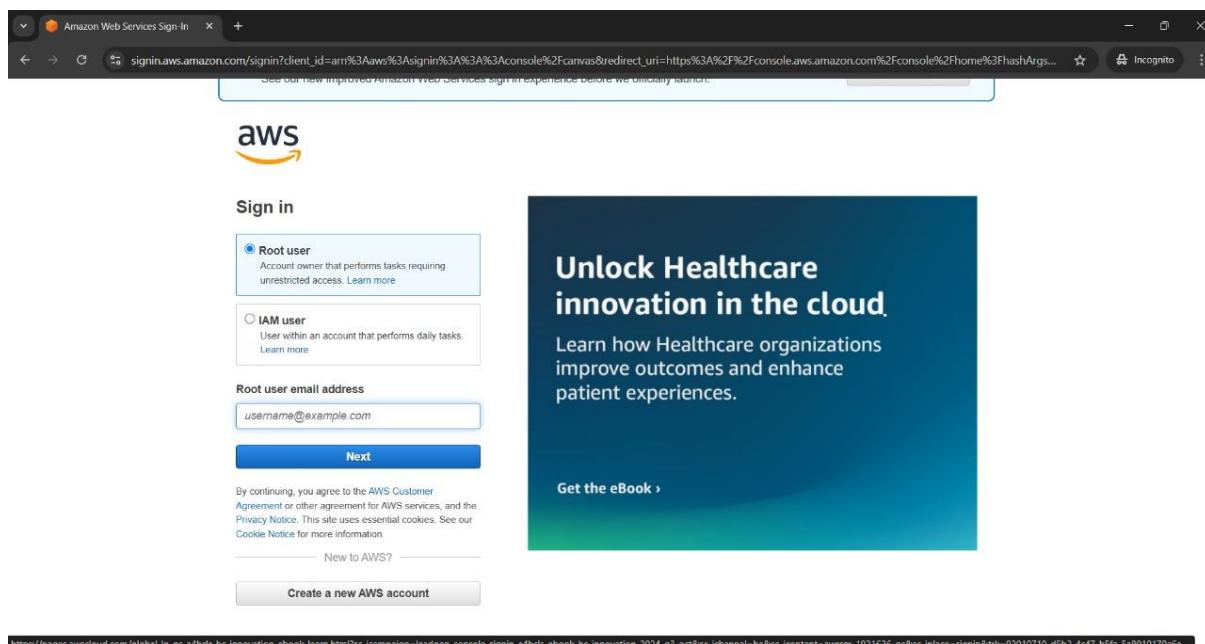
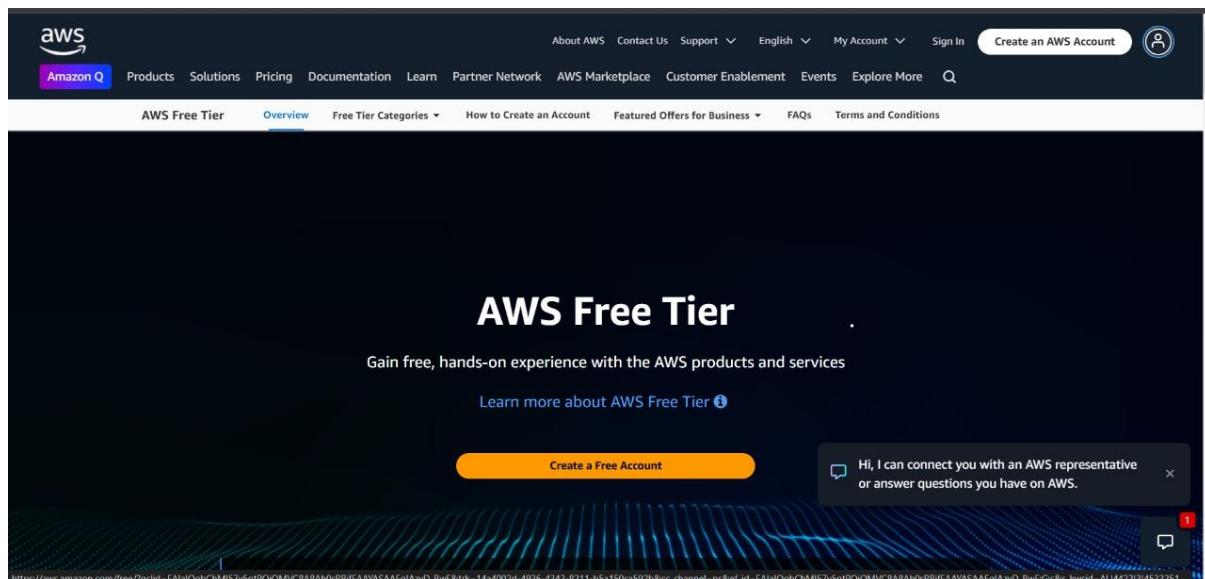
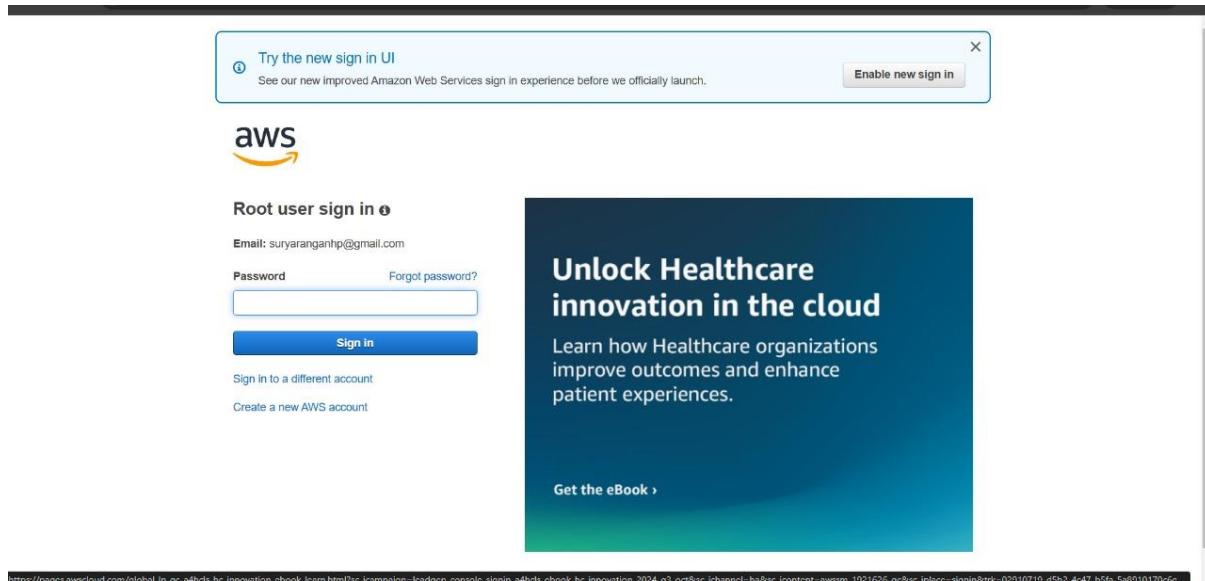


DOCUMENTATION ON HOSTING THE WEBSITE ON AWS (AMAZON WEB SERVICES) USING GITHUB REPOSITORY

1. Login to your AWS Account (If not create a new account by following the procedures given by the AWS).





2. After login, you will see the dashboard.

A screenshot of the AWS EC2 dashboard. The search bar at the top left contains "ec2". The left sidebar has a "Services" section with various options like Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, and Network & Security. Under "Instances", "Instances" is selected. The main content area shows search results for "ec2": "EC2" (Virtual Servers in the Cloud), "EC2 Image Builder" (A managed service to automate build, customize and deploy OS images), and "Recycle Bin" (Protect resources from accidental deletion). On the right, there's a panel for "EC2 Free Tier Info" which says "Offers for all AWS Regions." and "0 EC2 free tier offers in use". It also shows "End of month forecast" and "Exceeds free tier" information. Below this is a "View all AWS Free Tier offers" button. Further down is an "Account attributes" section with "Default VPC" (vpc-0465e2b4e7f88ed0) and "Settings" for Data protection and security, Zones, EC2 Serial Console, Default credit specification, and EC2 console preferences.

3. In the search bar, search for EC2 and click.

4. In the left bar, select instances.

Surya Rangan H P ▾

Dashboard

- EC2 Global View
- Events
- Instances
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity
 - Reservations **New**
- Images
 - AMIs
 - AMI Catalog
- Elastic Block Store
 - Volumes
 - Snapshots
 - Lifecycle Manager
- Network & Security
 - Security Groups
 - Elastic IPs

Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	1	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	1
Key pairs	9	Load balancers	0	Placement groups	0
Security groups	9	Snapshots	0	Volumes	1

Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Service health

AWS Health Dashboard

Note: Your instances will launch in the Asia Pacific (Mumbai) Region

Instance alarms

Zones

Zone name	Zone ID
ap-south-1a	aps1-az1

EC2 Free Tier Info
Offers for all AWS Regions.

0 EC2 free tier offers in use

End of month forecast:
⚠️ 0 offers forecasted to exceed free tier limit.

Exceeds free tier:
⚠️ 0 offers exceeded and is now pay-as-you-go pricing.

[View Global EC2 resources](#)

[View all AWS Free Tier offers](#)

Account attributes

Default VPC **vpc-0465e22b4e7f88ed0**

Settings

- Data protection and security
- Zones
- EC2 Serial Console
- Default credit specification
- EC2 console preferences

https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

5. Select launch instance.

Surya Rangan H P ▾

Dashboard

- EC2 Global View
- Events
- Instances
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity
 - Reservations **New**
- Images
 - AMIs
 - AMI Catalog
- Elastic Block Store
 - Volumes
 - Snapshots
 - Lifecycle Manager
- Network & Security
 - Security Groups
 - Elastic IPs

Instances (1) Info

Last updated: less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
Website	i-049d2822e84e1dd35	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-233-253-197.ap...	13.23

Select an instance

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

6. In the “Name and Tags” field, select and type the name as you wish.

The screenshot shows the "Launch an instance" wizard. On the left, under "Name and tags", a text input field contains "e.g. My Web Server". On the right, the "Summary" section shows "Number of instances: 1", "Virtual server type (instance type): t2.micro", and "Storage (volumes): 1 volume(s) - 8 GiB". A tooltip for "Free tier" is displayed, stating: "Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet." At the bottom are "Cancel", "Launch instance", and "Preview code" buttons.

7. Scroll down and in the Key Pair (Login), click on “Create new key pair” and type the name as you want select “create key pair” and a .pem file will be downloaded.

The screenshot shows the "Key pair (login)" configuration. It includes a note: "You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance." Below is a dropdown menu labeled "Select" and a button labeled "C Create new key pair".

Create key pair

X

Key pair name

Key pairs allow you to connect to your instance securely.

hello

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA

RSA encrypted private and public key pair

ED25519

ED25519 encrypted private and public key pair

Private key file format

.pem

For use with OpenSSH

.ppk

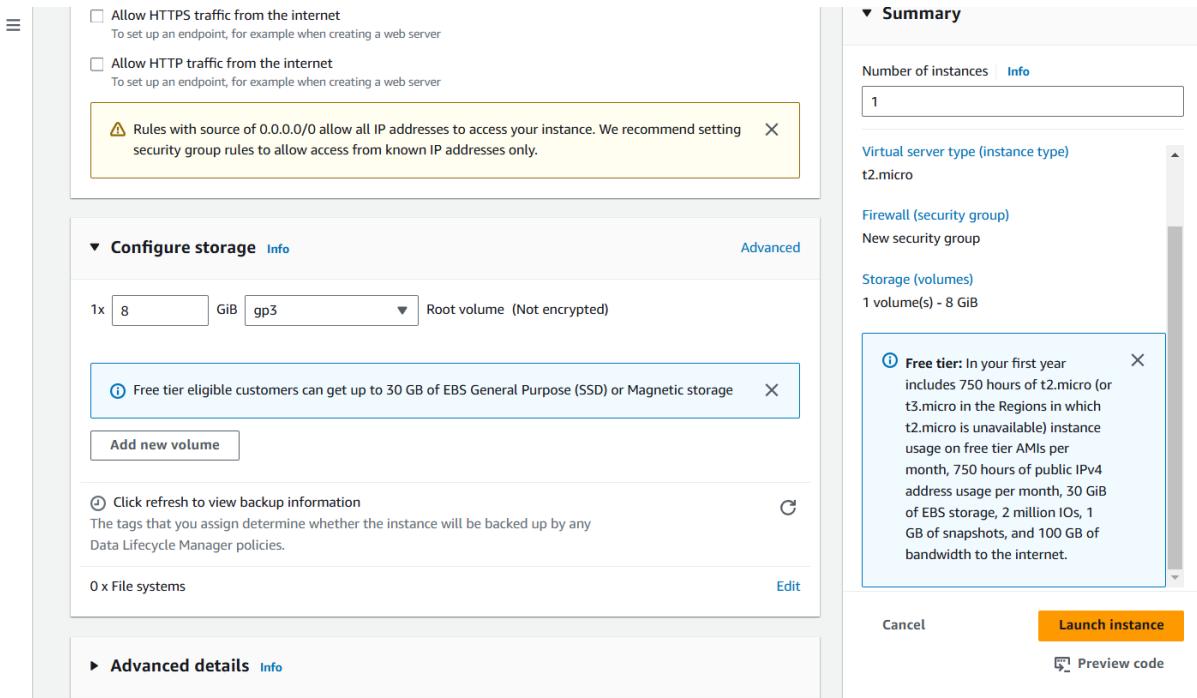
For use with PuTTY

 When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#) 

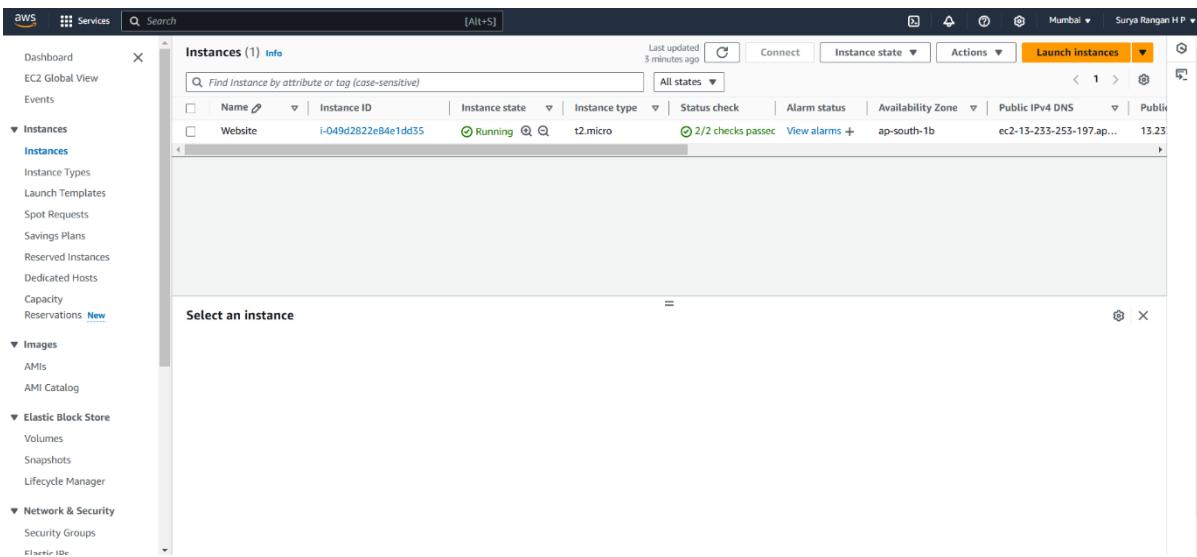
Cancel

Create key pair

8. Scroll down and select launch instance



9. After that EC2 -> instances again, to see the instance you created. Make sure that your instance is running. If not, wait for some time until it runs.



10. In the instances table, select the instance you created, go down and copy the Auto assigned Ip address and open a new tab in browser and paste it and come back to AWS page.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, and CloudShell. The main area displays an instance summary for 'i-049d2822e84e1dd35'. Key details include:

- Instance ID: i-049d2822e84e1dd35
- IPv4 address: 13.233.253.197 [Public IP]
- Instance state: Running
- Private IP DNS name (IPv4 only): ip-172-31-7-112.ap-south-1.compute.internal
- Instance type: t2.micro
- VPC ID: vpc-0463e22b4c7f88ed0
- Subnet ID: subnet-090ae2584fb72c233
- Instance ARN: arnawssec2:ap-south-1:050752626750:instance/i-049d2822e84e1dd35

At the top right, there are buttons for 'Connect' and 'Actions'. Below the main table, there are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. A tooltip at the top right says 'Public IPv4 address copied'.

11. Select Connect at the top. You will go to the new page and again select connect there.

The screenshot shows the 'Connect to Instance' dialog box. At the top, a warning message states: "Port 22 (SSH) is open to all IPv4 addresses. Port 22 (SSH) is currently open to all IPv4 addresses, indicated by **0.0.0.0/0** in the inbound rule in your [security group](#). For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 13.233.177.0/29. [Learn more](#)".

The dialog form includes the following fields:

- Instance ID: i-049d2822e84e1dd35 (Website)
- Connection Type:
 - Connect using EC2 Instance Connect: Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.
 - Connect using EC2 Instance Connect Endpoint: Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.
- Public IPv4 address: 13.233.253.197
- IPv6 address: —
- Username: ec2-user
- Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

At the bottom right, there are 'Cancel' and 'Connect' buttons.

12. You will get an Amazon CLI.

13. Type the following commands:

1. sudo su –
2. yum update -y
3. yum install -y httpd
4. systemctl status httpd
5. mkdir aws_anyfilename
6. cd aws_anyfilename

```
VS Services Search [Alt+S] N. Virginia vclabs/user1899835=shreyas.22010443@viit.ac.in @ 2345-9171-879
stalling : mod_http2-1.15.19-1.amzn2.0.1.x86_64
stalling : httpd-2.4.54-1.amzn2.x86_64
rifying : apr-util-1.6.1-5.amzn2.0.2.x86_64
rifying : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64
rifying : httpd-tools-2.4.54-1.amzn2.x86_64
rifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64
rifying : httpd-2.4.54-1.amzn2.x86_64
rifying : mailcap-2.1.41-2.amzn2.noarch
rifying : generic-logos-httpd-18.0.0-4.amzn2.noarch
rifying : httpd-filesystem-2.4.54-1.amzn2.noarch
rifying : apr-1.7.0-9.amzn2.x86_64
alled:
tpd.x86_64 0:2.4.54-1.amzn2

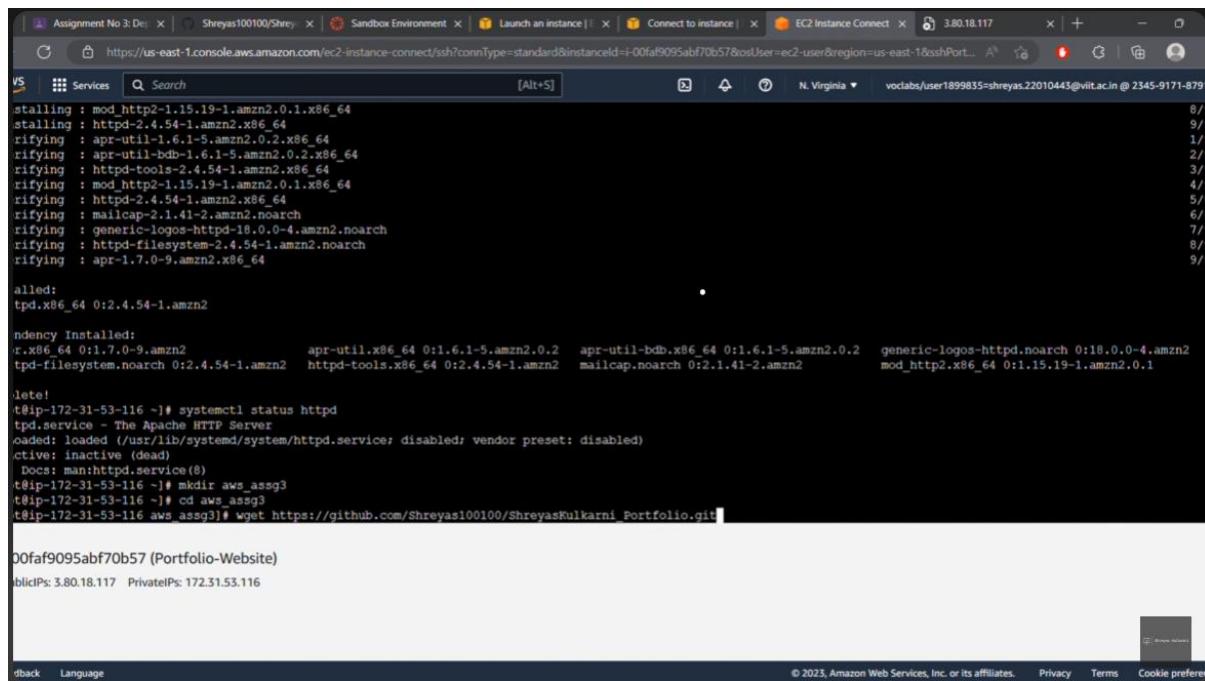
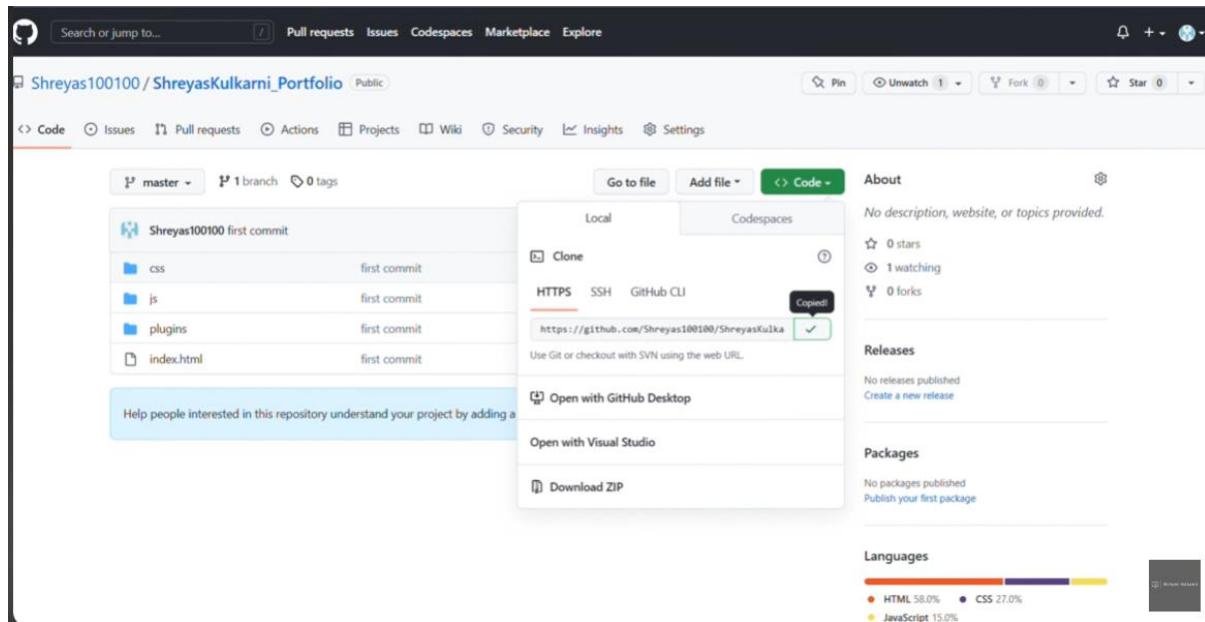
ndency Installed:
r.x86_64 0:1.7.0-9.amzn2      apr-util.x86_64 0:1.6.1-5.amzn2.0.2  apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2  generic-logos-httpd.noarch 0:18.0.0-4.amzn2
tpd-filesystem.noarch 0:2.4.54-1.amzn2  httpd-tools.x86_64 0:2.4.54-1.amzn2  mailcap.noarch 0:2.1.41-2.amzn2  mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

late!
t@ip-172-31-53-116 ~]$ systemctl status httpd
httpd.service - The Apache HTTP Server
loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
active: inactive (dead)
  Docs: man:httpd.service(8)
t@ip-172-31-53-116 ~]$ mkdir aws_assg3
t@ip-172-31-53-116 ~]$ cd aws_assg3
t@ip-172-31-53-116 aws_assg3]$ 
```

00faf9095abf70b57 (Portfolio-Website)
Public IPs: 3.80.18.117 Private IPs: 172.31.53.116

Logout Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

14. Open GitHub where you have the GitHub repository of your project. Select the repository and go to code and copy https link and type the following command:
wget “linkyouthavegot”



15. Now continue with the commands

1. ls -lrt

```
Assignment No 3: Deploying a static website | Shreyas100100/ShreyasKulkarni_Portfolio | Sandbox Environment | Launch an instance | Connect to instance | EC2 Instance Connect | 3.80.18.117

https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-00faf9095abf70b57&osUser=ec2-user&region=us-east-1&sshPort=22010

[ Services Search [Alt+S] ] [ ] [ ] [ ] [ ] [ ] N. Virginia [ ] vclabs/user1899835=shreyas.22010

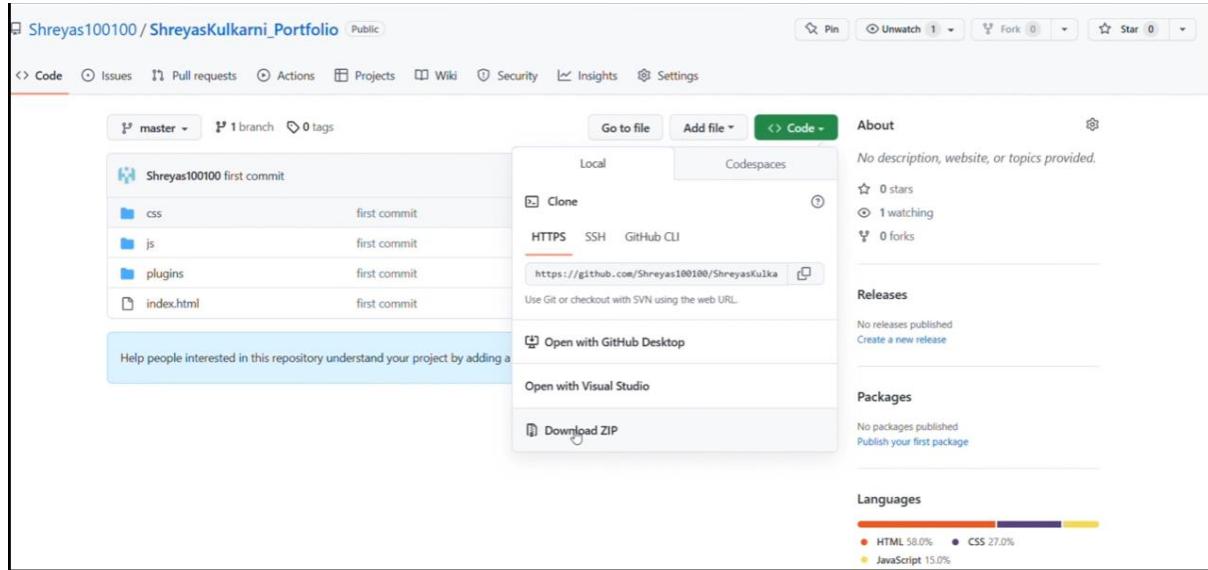
ls
[...]
t@ip-172-31-53-116 ~]# systemctl status httpd
httpd.service - The Apache HTTP Server
   loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   active: inactive (dead)
     Docs: man:httpd.service(8)
t@ip-172-31-53-116 ~]# mkdir aws_assg3
t@ip-172-31-53-116 ~]# cd aws_assg3
t@ip-172-31-53-116 aws_assg3]# wget https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio.git
23-02-04 14:44:20-- https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio.git
l:ving github.com (github.com)... 140.82.114.4
ecting to github.com (github.com)... 140.82.114.4:443... connected.
request sent, awaiting response... 301 Moved Permanently
location: https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio [following]
23-02-04 14:44:20-- https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio
ing existing connection to github.com:443.
request sent, awaiting response... 200 OK
Content-Type: unspecified [text/html]
ng to: 'ShreyasKulkarni_Portfolio.git'

[ <=>
-02-04 14:44:20 (15.2 MB/s) - 'ShreyasKulkarni_Portfolio.git' saved [169529]

t@ip-172-31-53-116 aws_assg3]# ls -lrt
l 168
r--r-- 1 root root 169529 Feb 4 14:44 ShreyasKulkarni_Portfolio.git
t@ip-172-31-53-116 aws_assg3]# [ 169,529

00faf9095abf70b57 (Portfolio-Website)
Public IPs: 3.80.18.117 Private IPs: 172.31.53.116
```

16. Go to GitHub, select repository ->Code, right-click on Download ZIP and select “Copy link address.”



17. Type this

wget “paste_the_zip_link”

```
Assignment No 3: Deployment | Shreyas100100/ShreyasKulkarni_Portfolio | Sandbox Environment | Launch an instance | Connect to instance | EC2 Instance Connect | 3.80.18.117 | - | Services | Search | [Alt+S] | N. Virginia | vclabs/user1899835=shreyas.22010443@viit.ac.in @ 2345-9171-8791

t@ip-172-31-53-116 ~]$ systemctl status httpd
httpd.service - The Apache HTTP Server
   loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   active: inactive (dead)
     Docs: man:httpd.service(8)

t@ip-172-31-53-116 ~$ mkdir aws_assg3
t@ip-172-31-53-116 ~$ cd aws_assg3
t@ip-172-31-53-116 aws_assg3]$ wget https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio.git
23-02-04 14:44:20 -- https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio.git
lving github.com (github.com)... 140.82.114.4
ecting to github.com (github.com)|140.82.114.4|:443... connected.
request sent, awaiting response... 301 Moved Permanently
location: https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio (following)
23-02-04 14:44:20 -- https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio
ing existing connection to github.com:443.
request sent, awaiting response... 200 OK
Content-Type: application/x-gzip
Content-Length: 169529
Last-Modified: Mon, 04 Feb 2024 14:44:20 UTC
Content-Encoding: gzip
Content-Type: text/html
Content-Length: 168
Content-Encoding: gzip
Content-Type: application/x-gzip
Content-Length: 619972
Content-Encoding: gzip

[ <> ] 169,529    --.-K/s   in 0.01s

-02-04 14:44:20 (15.2 MB/s) - 'ShreyasKulkarni_Portfolio.git' saved [169529]

t@ip-172-31-53-116 aws_assg3]$ ls -lrt
l 168
r--r-- 1 root root 169529 Feb  4 14:44 ShreyasKulkarni_Portfolio.git
t@ip-172-31-53-116 aws_assg3]$ wget https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio/archive/refs/heads/master.zip
23-02-04 14:44:49 -- https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio/archive/refs/heads/master.zip
lving github.com (github.com)... 140.82.114.4
ecting to github.com (github.com)|140.82.114.4|:443... connected.
request sent, awaiting response... 302 Found
location: https://code.load.github.com/Shreyas100100/ShreyasKulkarni_Portfolio/zip/refs/heads/master [following]
23-02-04 14:44:49 -- https://code.load.github.com/Shreyas100100/ShreyasKulkarni_Portfolio/zip/refs/heads/master
lving code.load.github.com (code.load.github.com)|140.82.113.10|:443... connected.
request sent, awaiting response... 200 OK
Content-Type: application/zip
Content-Length: 619972
Content-Encoding: gzip
Content-Type: text/html
Content-Length: 176
Content-Encoding: gzip
Content-Type: application/zip
Content-Length: 619972
Content-Encoding: gzip

[ <> ] 619,972    --.-K/s   in 0.1s

-02-04 14:44:49 (5.57 MB/s) - 'master.zip' saved [619972]

t@ip-172-31-53-116 aws_assg3]$ ls -lrt
l 776
r--r-- 1 root root 169529 Feb  4 14:44 ShreyasKulkarni_Portfolio.git
r--r-- 1 root root 619972 Feb  4 14:44 master.zip
t@ip-172-31-53-116 aws_assg3]$ unzip master.zip

00faf9095abf70b57 (Portfolio-Website)
Public IPs: 3.80.18.117 Private IPs: 172.31.53.116
```

18. Type the commands:

1. ls -lrt

2. You will get the zip file name displayed in cli. Unzip using this command: unzip “zipfilename.zip.”

```
Assignment No 3: Deployment | Shreyas100100/ShreyasKulkarni_Portfolio | Sandbox Environment | Launch an instance | Connect to instance | EC2 Instance Connect | 3.80.18.117 | - | Services | Search | [Alt+S] | N. Virginia | vclabs/user1899835=shreyas.22010443@viit.ac.in @ 2345-9171-8791

-02-04 14:44:20 (15.2 MB/s) - 'ShreyasKulkarni_Portfolio.git' saved [169529]

t@ip-172-31-53-116 aws_assg3]$ ls -lrt
l 168
r--r-- 1 root root 169529 Feb  4 14:44 ShreyasKulkarni_Portfolio.git
t@ip-172-31-53-116 aws_assg3]$ wget https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio/archive/refs/heads/master.zip
23-02-04 14:44:49 -- https://github.com/Shreyas100100/ShreyasKulkarni_Portfolio/archive/refs/heads/master.zip
lving github.com (github.com)... 140.82.114.4
ecting to github.com (github.com)|140.82.114.4|:443... connected.
request sent, awaiting response... 302 Found
location: https://code.load.github.com/Shreyas100100/ShreyasKulkarni_Portfolio/zip/refs/heads/master [following]
23-02-04 14:44:49 -- https://code.load.github.com/Shreyas100100/ShreyasKulkarni_Portfolio/zip/refs/heads/master
lving code.load.github.com (code.load.github.com)|140.82.113.10|:443... connected.
request sent, awaiting response... 200 OK
Content-Type: application/zip
Content-Length: 619972
Content-Encoding: gzip
Content-Type: text/html
Content-Length: 176
Content-Encoding: gzip
Content-Type: application/zip
Content-Length: 619972
Content-Encoding: gzip

[ <> ] 619,972    --.-K/s   in 0.1s

-02-04 14:44:49 (5.57 MB/s) - 'master.zip' saved [619972]

t@ip-172-31-53-116 aws_assg3]$ ls -lrt
l 776
r--r-- 1 root root 169529 Feb  4 14:44 ShreyasKulkarni_Portfolio.git
r--r-- 1 root root 619972 Feb  4 14:44 master.zip
t@ip-172-31-53-116 aws_assg3]$ unzip master.zip

00faf9095abf70b57 (Portfolio-Website)
Public IPs: 3.80.18.117 Private IPs: 172.31.53.116
```

3. Ls -lrt

4. You will get the folder name in blue color. Now type this: cd “folder_name”

5. Ls -lrt

6. You will get some files. Type this:

Mv * /var/www/html/

7. Cd /var/www/html

8. Ls -lrt

```
t@ip-172-31-53-116 aws_assg3]# cd ShreyasKulkarni_Portfolio-master
t@ip-172-31-53-116 ShreyasKulkarni_Portfolio-master]# ls -lrt
l 12
r-xr-x 6 root root 154 Feb 4 14:33 plugins
r-xr-x 2 root root 23 Feb 4 14:33 js
r--r-- 1 root root 10607 Feb 4 14:33 index.html
r-xr-x 3 root root 94 Feb 4 14:33 css
t@ip-172-31-53-116 ShreyasKulkarni_Portfolio-master]# mv * /var/www/html/
t@ip-172-31-53-116 ShreyasKulkarni_Portfolio-master]# cd /var/www/html
t@ip-172-31-53-116 html]# ls -lrt
l 12
r-xr-x 6 root root 154 Feb 4 14:33 plugins
r-xr-x 2 root root 23 Feb 4 14:33 js
r--r-- 1 root root 10607 Feb 4 14:33 index.html
r-xr-x 3 root root 94 Feb 4 14:33 css
t@ip-172-31-53-116 html]# █
```

19. Go to the AWS Instances.

20. Select the instance ID of the instance you created, then go to security, inbound rules, and select launch wizard.

Instances (1/2) **Info**

Name	Instance ID	Instance state	Instance type	Status check	Availability Zone	Public IPv4 DNS
Portfolio-Website	i-00faf9095abf70b57	Running	t2.micro	2/2 checks passed	us-east-1e	ec2-3-80-18-117.comp...
Bastion Host	i-02654f03f44b0e840	Running	t2.micro	2/2 checks passed	us-east-1a	ec2-44-202-211-234.co...

Instance: i-00faf9095abf70b57 (Portfolio-Website)

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups
-	sgr-038d117f333cb1cf7	22	TCP	0.0.0.0/0	launch-wizard-1

Outbound rules

Filter rules
-

21. In that select inbound rules tab, select Edit inbound rules.

Security Groups (1/1) **Info**

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-014028b89c22543a9	launch-wizard-1	vpc-0d8ded0e051b1191b ...	launch-wizard-1 create...	234591718791

sg-014028b89c22543a9 - launch-wizard-1

Details | **Inbound rules** | Outbound rules | Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Inbound rules (1/1)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-038d117f333cb1cf7	IPv4	SSH	TCP	22

22. Select Add rule and perform these

1. HTTP, anywhere ipv4, 0:0:0:0/0, select add rule
2. https, anywhere ipv4, 0:0:0:0/0

EC2 > Security Groups > sg-014028b89c22543a9 - launch-wizard-1 > Edit inbound rules

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-038d117f333cb1cf7	SSH	TCP	22	Custom	Q. 0.0.0.0/0
-	HTTP	TCP	80	Anywhere	0.0.0.0/0
Add rule					

Cancel [Preview changes](#) [Save rules](#)

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

EC2 > Security Groups > sg-014028b89c22543a9 - launch-wizard-1 > Edit inbound rules

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-038d117f333cb1cf7	SSH	TCP	22	Custom	Q. 0.0.0.0/0
-	HTTP	TCP	80	Anywhere	Web Port 0.0.0.0/0
-	HTTPS	TCP	443	Anywhere	Web Port 0.0.0.0/0
Add rule					

Cancel [Preview changes](#) [Save rules](#)

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

23. Go to Aws cli and type this

Systemctl status httpd

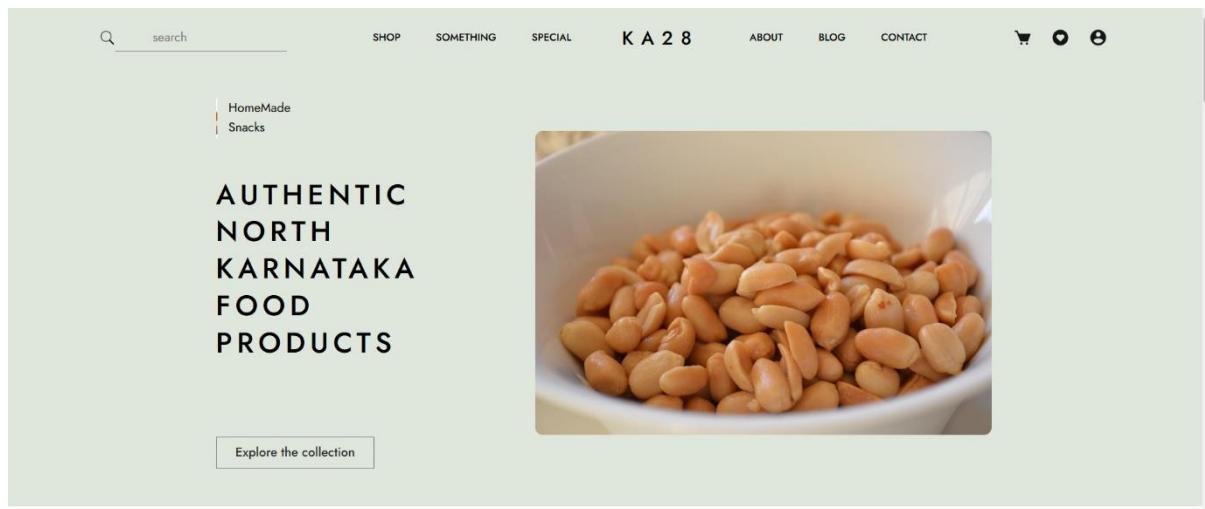
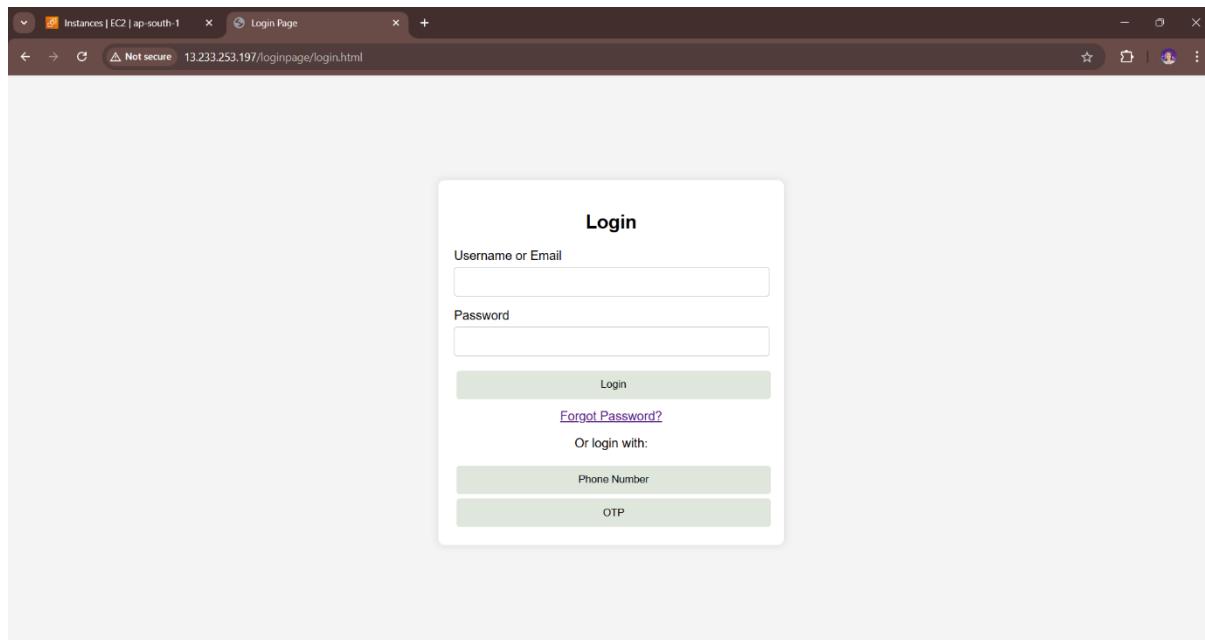
Systemctl enable httpd.

Systemctl start httpd

```
VS Services Search [Alt+S] N. Virginia vocabs/user1899835=shreyas.22010445@vliit.ac.in @ 2345-9171-8791
t@ip-172-31-53-116 aws_assg3]# ls -lrt
1 776
r--x--x 5 root root 60 Feb 4 14:33 ShreyasKulkarni_Portfolio-master
r--r-- 1 root root 169529 Feb 4 14:44 ShreyasKulkarni_Portfolio.git
r--r-- 1 root root 619972 Feb 4 14:44 master.zip
t@ip-172-31-53-116 aws_assg3]# cd ShreyasKulkarni_Portfolio-master
t@ip-172-31-53-116 ShreyasKulkarni_Portfolio-master]# ls -lrt
1 12
r--x--x 6 root root 154 Feb 4 14:33 plugins
r--x--x 2 root root 23 Feb 4 14:33 js
r--r-- 1 root root 10607 Feb 4 14:33 index.html
r--x--x 3 root root 94 Feb 4 14:33 css
t@ip-172-31-53-116 ShreyasKulkarni_Portfolio-master]# mv * /var/www/html/
t@ip-172-31-53-116 ShreyasKulkarni_Portfolio-master]# cd /var/www/html
t@ip-172-31-53-116 html]# ls -lrt
1 2
r--x--x 6 root root 154 Feb 4 14:33 plugins
r--x--x 2 root root 23 Feb 4 14:33 js
r--r-- 1 root root 10607 Feb 4 14:33 index.html
r--x--x 3 root root 94 Feb 4 14:33 css
t@ip-172-31-53-116 html]# systemctl status httpd
httpd.service - The Apache HTTP Server
   loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   active: inactive (dead)
     Docs: man:httpd.service(8)
t@ip-172-31-53-116 html]# systemctl enable httpd
  Failed to set property 'Wants' on target '/usr/lib/systemd/system/httpd.service': Invalid argument
t@ip-172-31-53-116 html]# systemctl start httpd
[00faf9095abf70b57 (Portfolio-Website)
PublicIPs: 3.80.18.117 PrivateIPs: 172.31.53.116

dback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

24. Now reload the tab where you put your Ip address and see the results of your website.



How to deploy with Microsoft Azure

Step 1: - Create your Microsoft Azure account and make a new storage account and name it



Step 2: - Here I named it as geminipush, now select the geminipush storage account

A screenshot of the Microsoft Azure Storage accounts page. The top navigation bar shows "Storage accounts" and the user's email "YASH.DIVYA@studenta... STUDENT AMBASSADORS (STDN...)" with a profile icon. Below the navigation, there are buttons for "Create", "Restore", "Manage view", "Refresh", "Export to CSV", "Open query", "Assign tags", and "Delete". There are also filter options: "Subscription equals all", "Resource group equals all", "Location equals all", and "Add filter". The main table lists two storage accounts: "csg100320031f315a5a" and "geminipush". The columns include Name, Type, Kind, Resource group, Location, and Subscription. Both accounts are listed under "cloud-shell-storage-central..." with "Central India" as the location and "Visual Studio Enterprise S..." as the subscription.

Step 3: - Go to Containers and select the web container

The screenshot shows the Microsoft Azure Storage accounts interface. In the left sidebar, under 'Data storage', 'Containers' is selected. The main area displays two containers: '\$logs' and '\$web'. Both containers were created on 8/11/2024 at 12:27:38 AM and are set to 'Private' with 'Available' lease state.

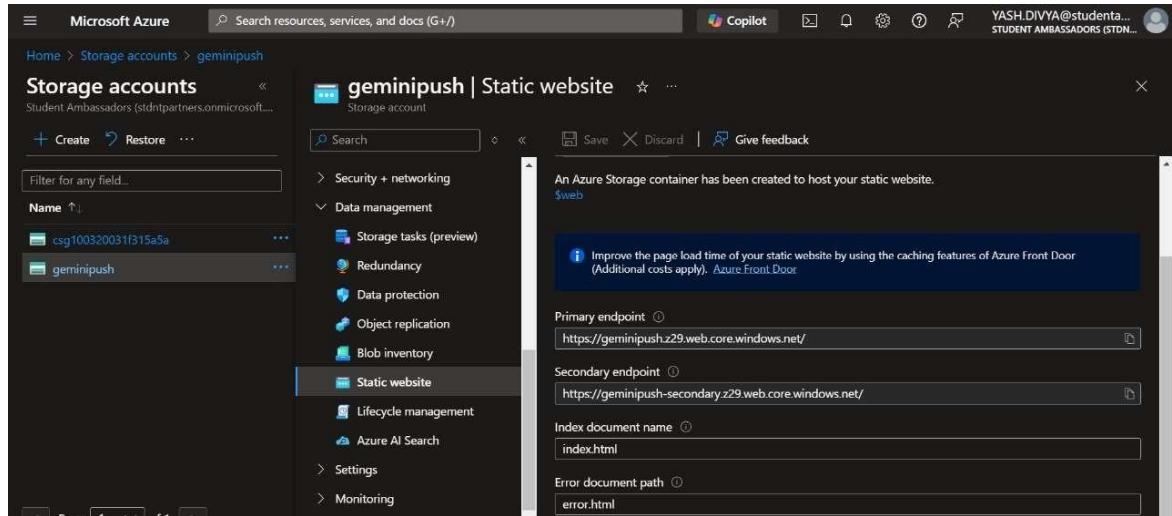
Name	Last modified	Anonymous access	Lease state
\$logs	8/11/2024, 12:27:38 ...	Private	Available
\$web	8/11/2024, 12:32:39 ...	Private	Available

Step 4: - Upload all your files inside the web container (make sure about giving the path and proper formatting)

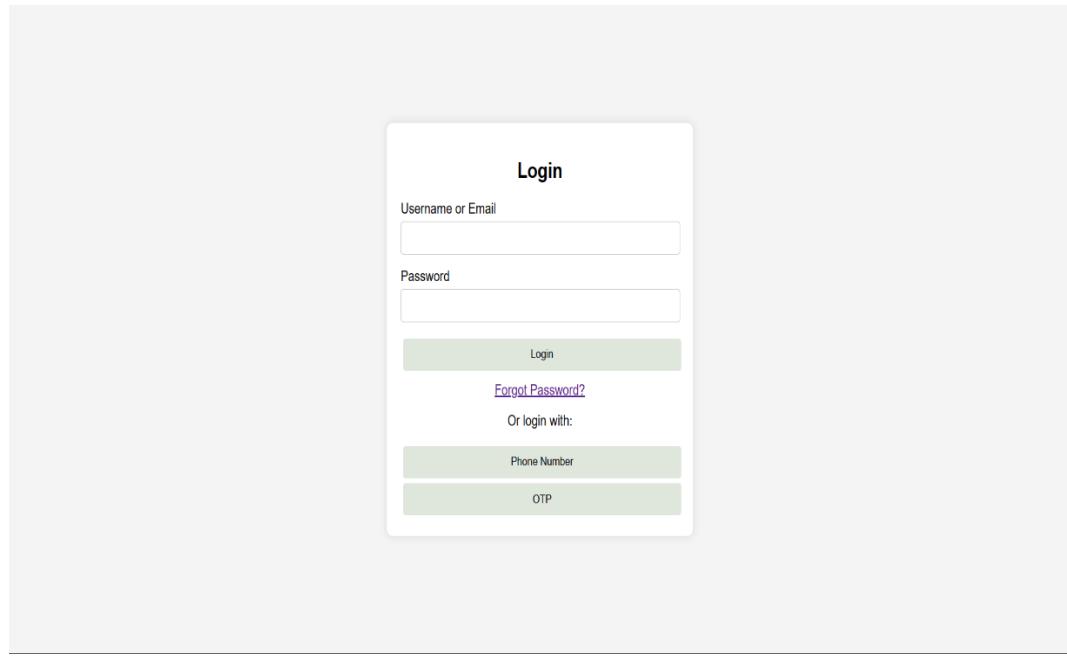
The screenshot shows the '\$web' container overview page. The left sidebar includes 'Overview', 'Diagnose and solve problems', 'Access Control (IAM)', and 'Settings'. The main area displays a table of blobs:

Name	Modified	Access tier	Archive status	Blob type	Size
app.py	8/11/2024, 12:35:52 ...	Hot (Inferred)		Block blob	1.09
demo.gif	8/11/2024, 12:35:31 ...	Hot (Inferred)		Block blob	785
Dockerfile	8/11/2024, 12:36:21 ...	Hot (Inferred)		Block blob	572
iba_logo.png	8/11/2024, 12:35:25 ...	Hot (Inferred)		Block blob	33.9
index.html	8/11/2024, 12:34:27 ...	Hot (Inferred)		Block blob	5.71

Step 5: - Go to static website and give the name of the index and the error document.



Step 6: Now you can use the generated index document link to access your website



 search

SHOP SOMETHING SPECIAL KA 28 ABOUT BLOG CONTACT



HomeMade
Snacks

AUTHENTIC
NORTH
KARNATAKA
FOOD
PRODUCTS

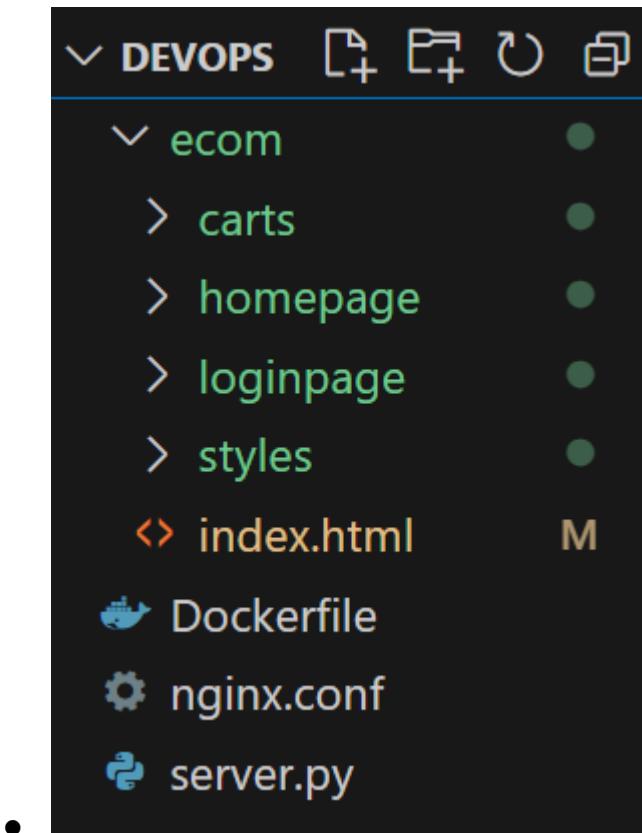


Explore the collection

CREATING THE PROJECT AND CONTAINERIZATION USING DOCKER

1. Creating the actual project – Webpage

- Here we are creating a webpage which basically falls under the category of E-commerce.
- We will be dividing this task into 3 separate folders under the actual main folder
- One for Login page, one for Homepage and the other for the Cart page
- Here is the file structure



2. Creating the Python HTTP Service

- Here, for the purpose of serving the webpage, we are making the use of Python HTTP service, other alternatives like Nginx can also be preferred.
- Create a file named index.py to start a simple HTTP server and serve the static webpage:

```
import http.server
import socketserver
import os

PORT = 8080
DIRECTORY = "ecom"

class Handler(http.server.SimpleHTTPRequestHandler):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, directory=DIRECTORY, **kwargs)

    def do_GET(self):
        if self.path == '/homepage' or self.path=='/':
            self.path = '/homepage/index.html'
        elif self.path == '/login' :
            self.path = '/loginpage/login.html'
        elif self.path == '/cart':
            self.path = '/carts/cart.html'
        return http.server.SimpleHTTPRequestHandler.do_GET(self)

    with socketserver.TCPServer(("", PORT), Handler) as httpd:
        print("Serving at port", PORT)
        httpd.serve_forever()
```

- `http.server.SimpleHTTPRequestHandler` automatically serves files in the current directory.
- The server will be available at <http://localhost:8080>.

3. Dockerize the application

- To run the static webpage in a container, we need to package it along with the Python HTTP service. This is done using Docker.

(1) Create a Dockerfile

- In the root of the project directory (ecom) as shown in the above picture, create a file named Dockerfile with the following content:

```
FROM python:3.9-slim

WORKDIR /app

# Copy the necessary files
COPY ./ecom /app/ecom
COPY ./server.py /app/server.py

# Expose port 8080
EXPOSE 8080

# Run the Python HTTP server
CMD ["python", "server.py"]
```

- Explanation of the Dockerfile:
- FROM python:3.9-slim: Uses a minimal Python 3.9 image.
- WORKDIR /app: Sets the working directory inside the container to /app.

- COPY ./ecom /app/ecom: Copies all files from the ecom directory into the container's /app/ecom directory.
- COPY ./server.py /app/server.py: Copies all content from the server file into the container's /app/server.py file.
- EXPOSE 8080: Exposes port 8080 for the Python HTTP server.
- CMD ["python", "app.py"]: Runs the Python application to start the HTTP server.

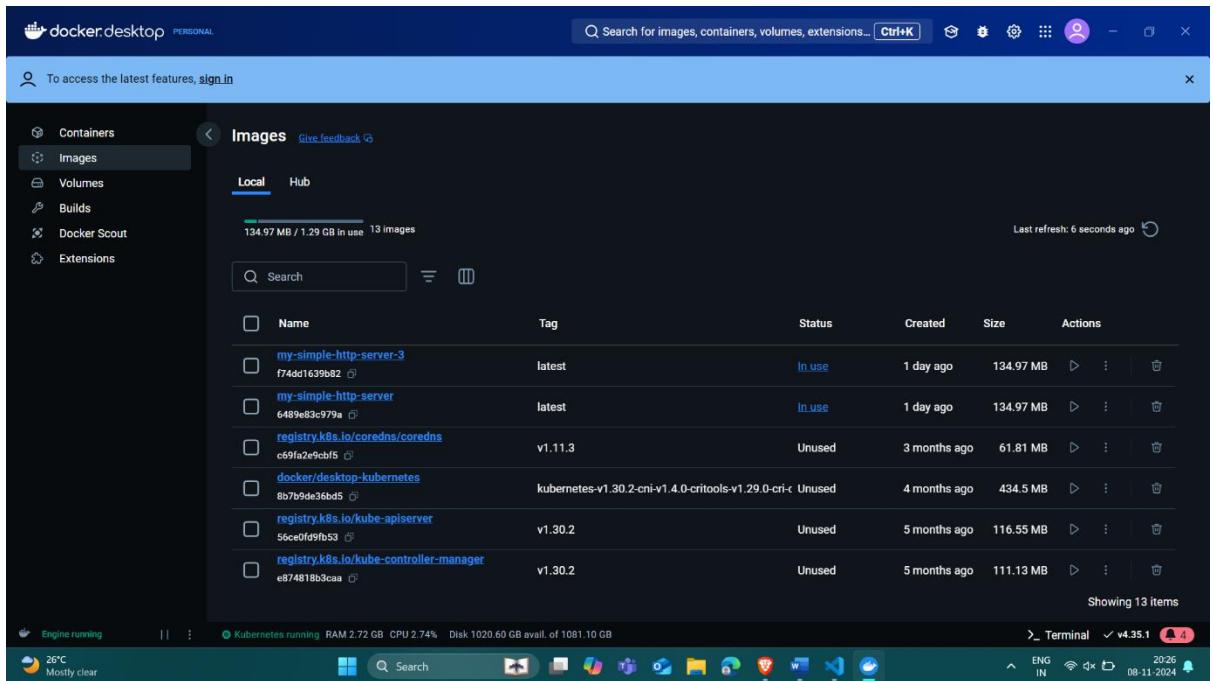
(2). Add the index.html and app.py to the Docker container:

Make sure both the index.html file and app.py are in the same directory as the Dockerfile.

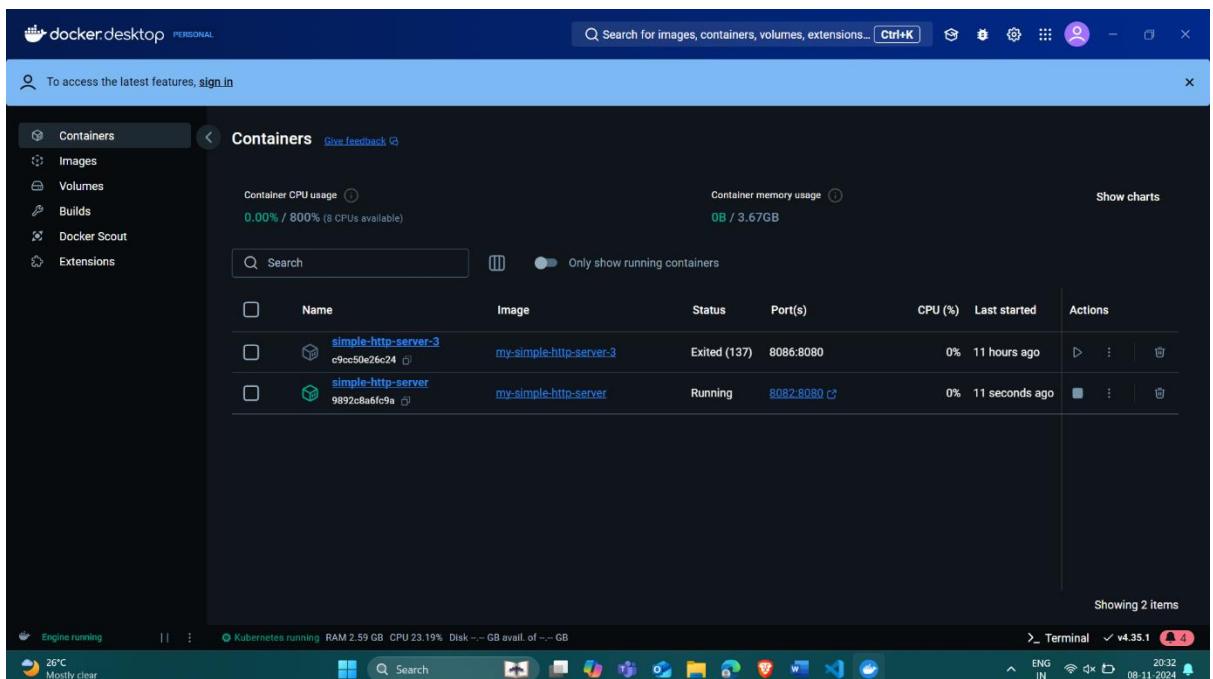
4. Test the docker container locally

- Open a terminal in the project directory ecom and run:
- docker build -t my-simple-http-server .
- This command builds the Docker image and tags it as my-simple-http-server
- Run the docker container using the code:
- docker run -p 8082:8080 simple-http-server
- Here, there is port mapping done, even though the server starts at the port 8080 originally, for us, the application will run on the port 8082.

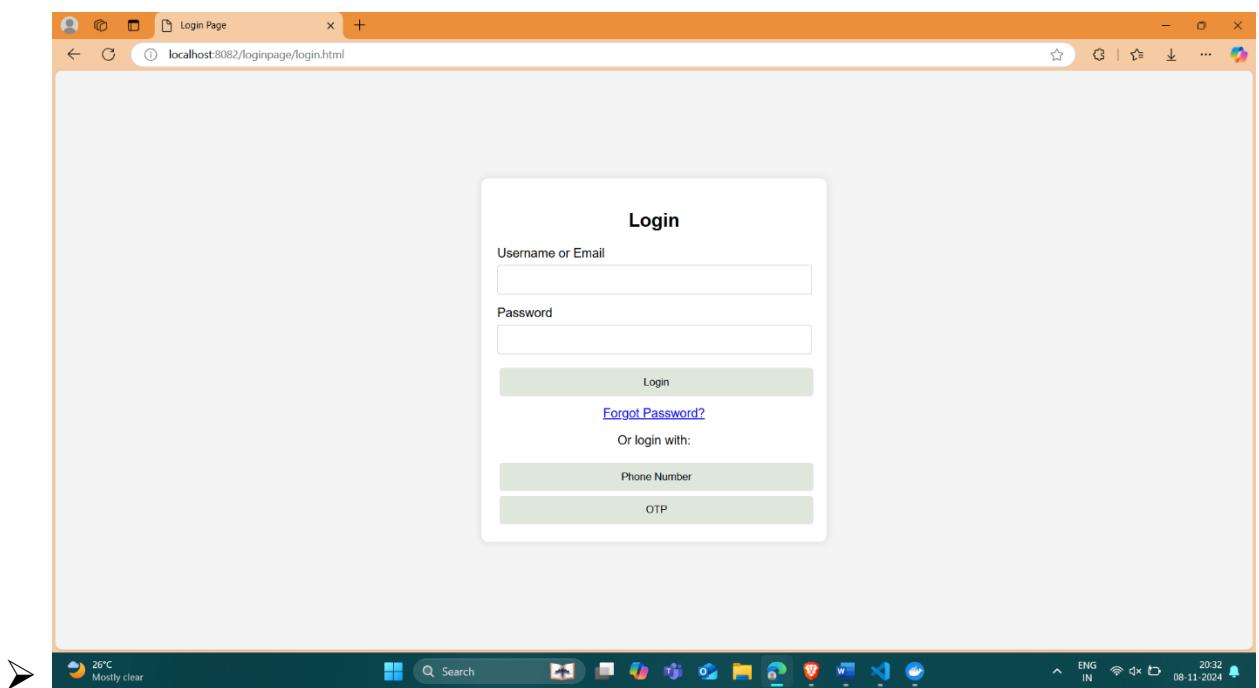
- Test the webpage:
- Open your web browser and go to <http://localhost:8082>. We can see the webpage served by our Python HTTP server.



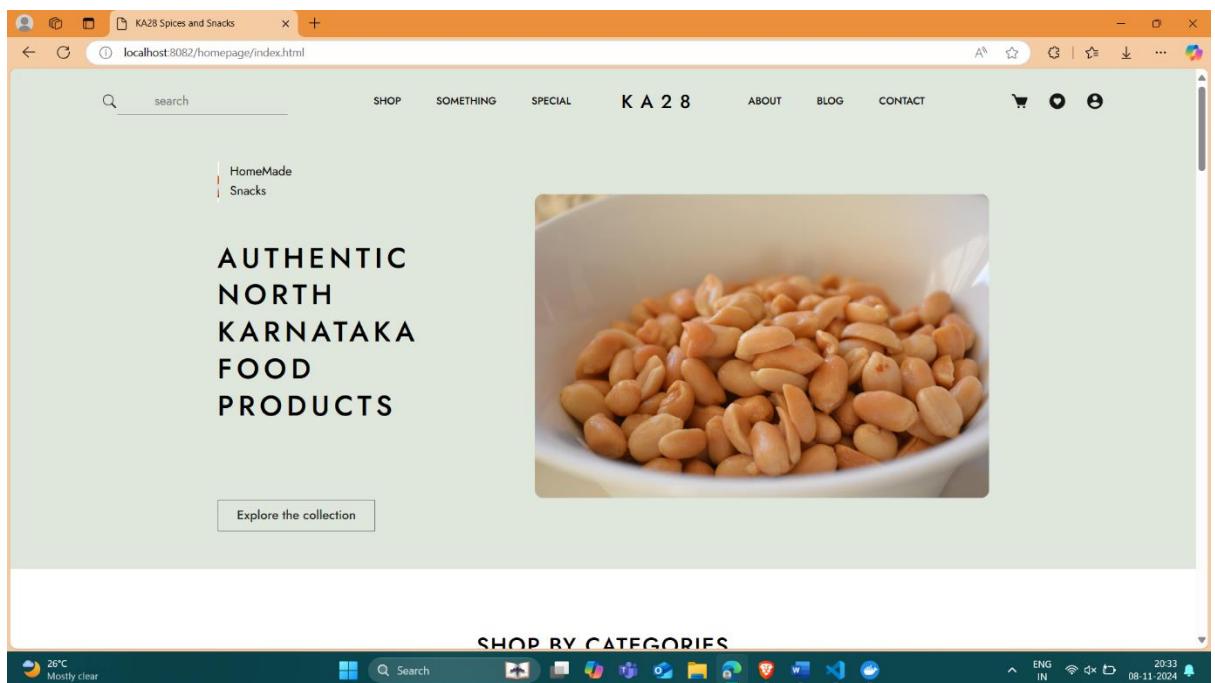
- Here, we can see that out of all these images, we have created an image called as “my-simple-http-server”



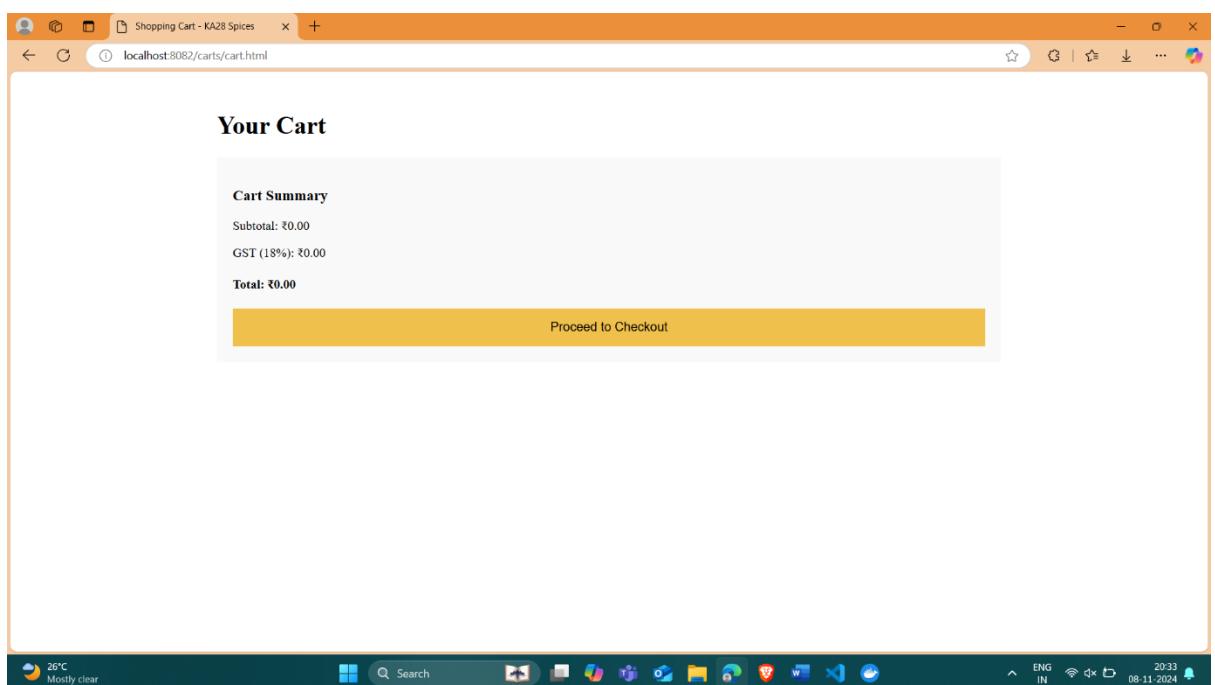
- In the above image you can see that the container has been created named as “simple-http-server” which runs with the image of “my-simple-http-server” which is being mapped on port 8082. This is the actual container which we have just created a while back.
- After clicking on the port 8082:8080 (or) going to the localhost:8082, we will get the following results
- Login Page:



❖ Homepage:



❖ Cart Page:

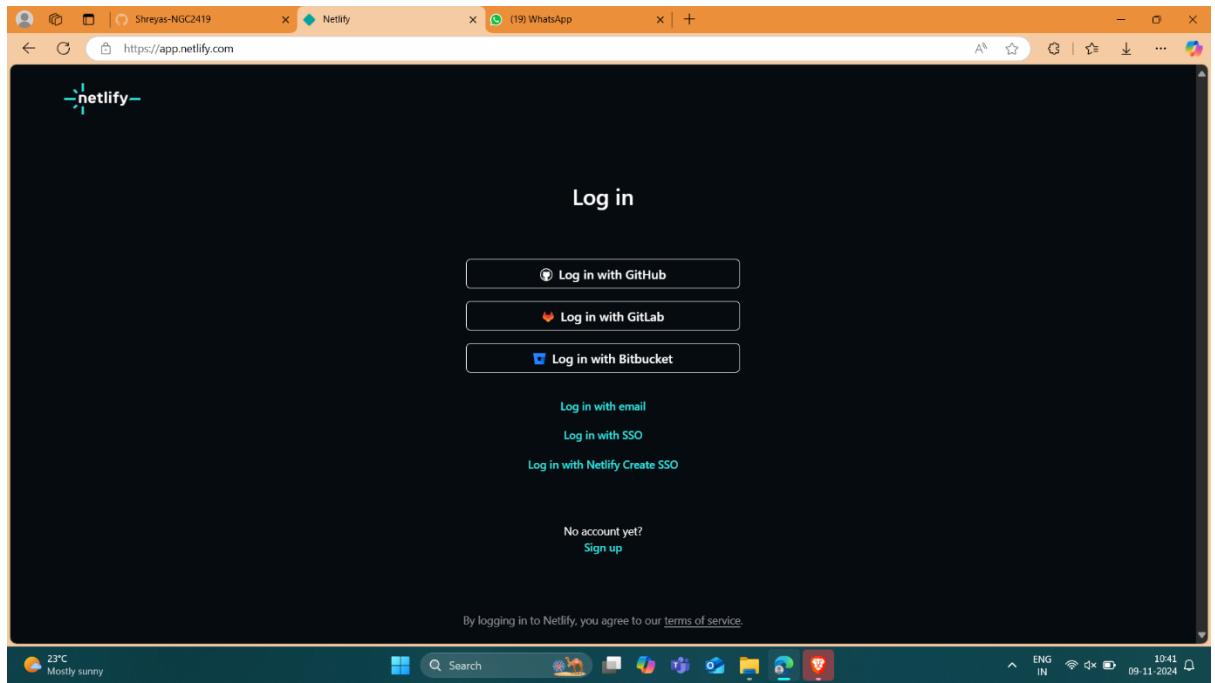


And hence, our webpage application is containerized and is running on the local system successfully.

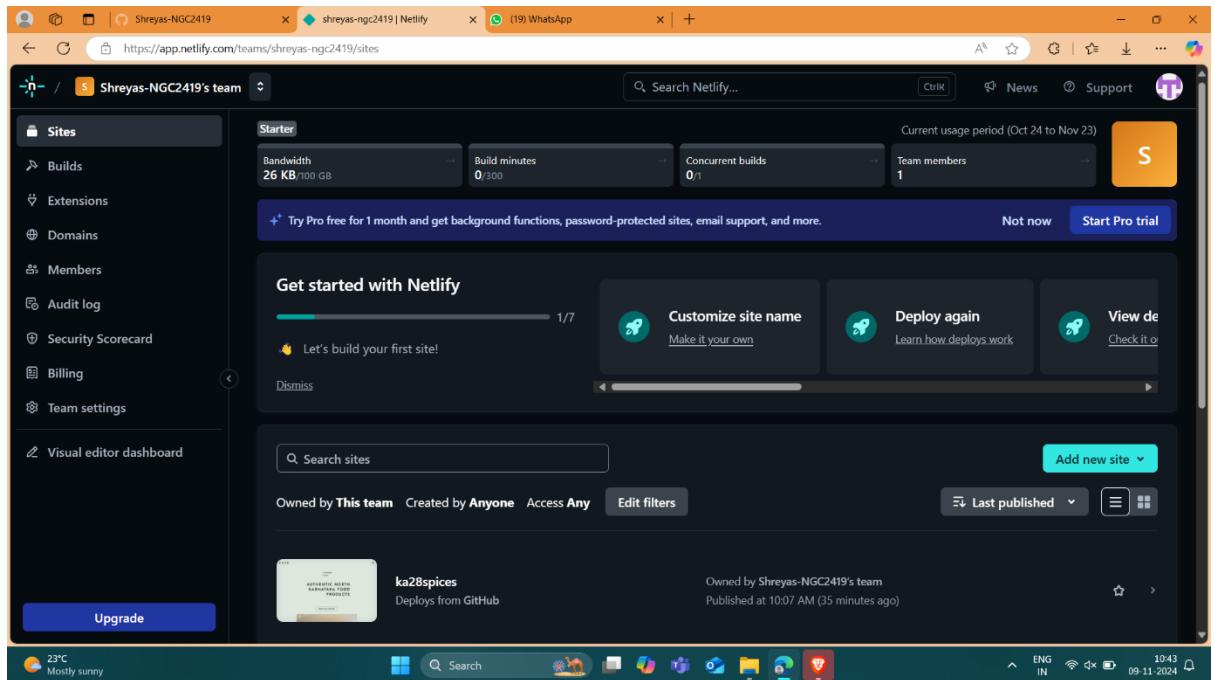
This can also be pushed into Docker Hub if we want to make it accessible to the outside world.

UPLOADING OF THE FILES AND DEPLOYMENT USING NETLIFY

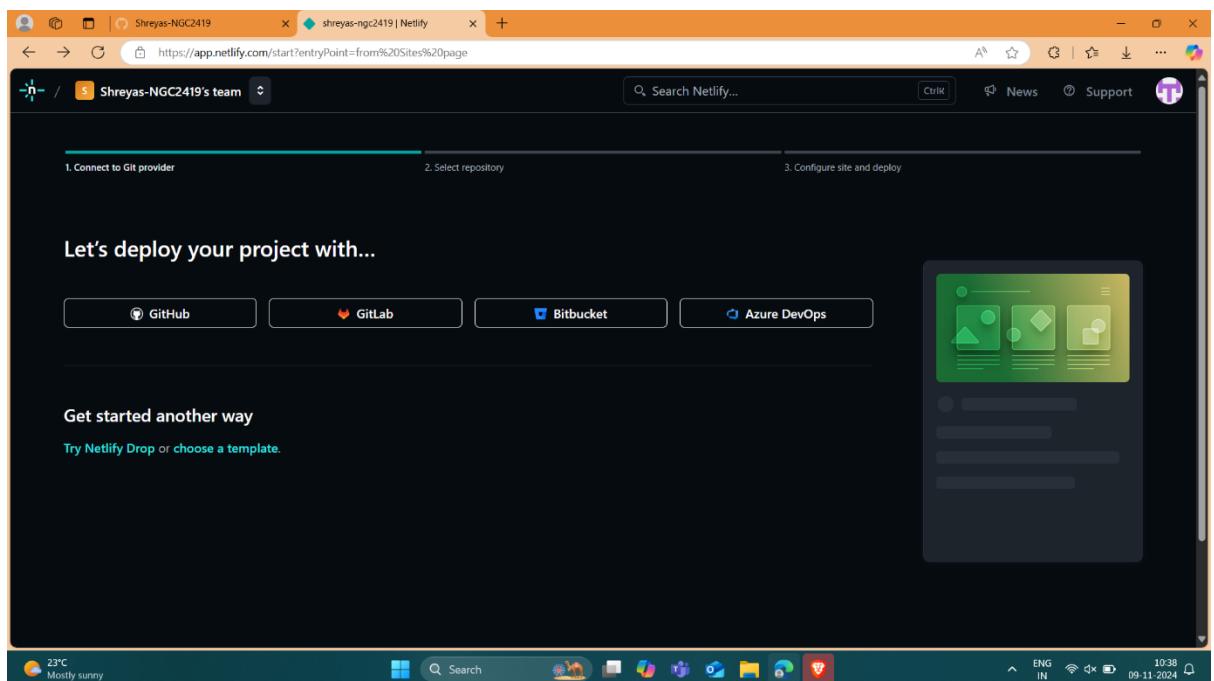
1. Login to your Netlify account (preferably using GitHub option).



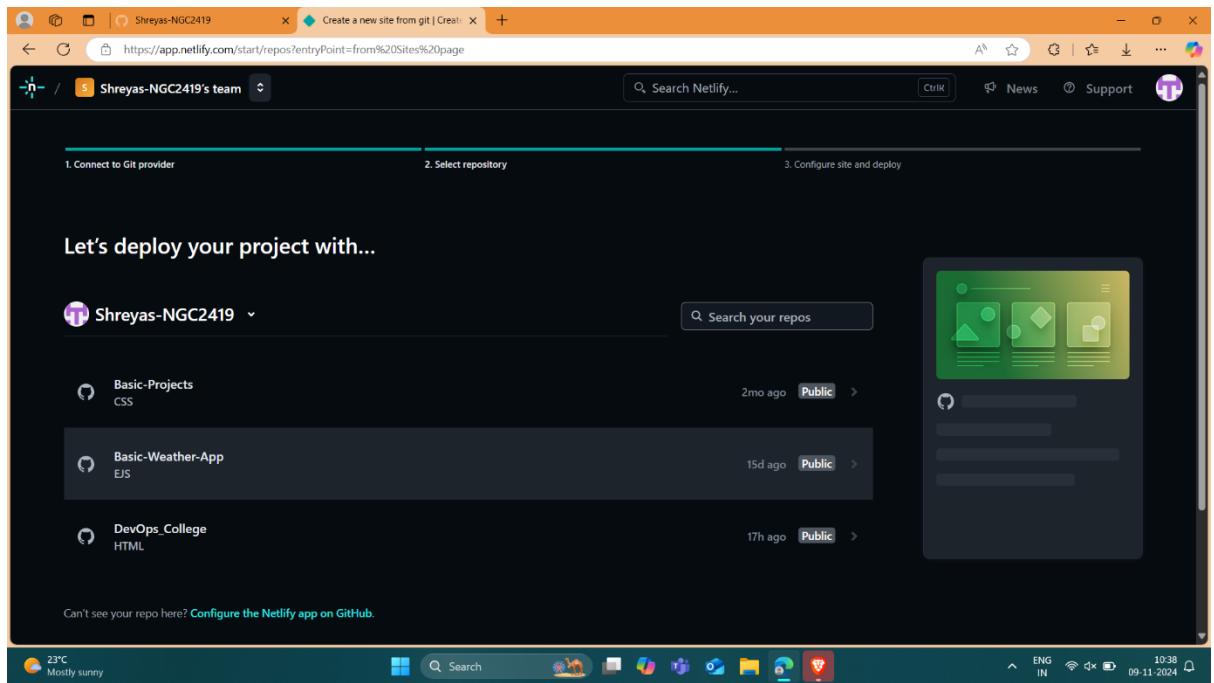
2. Click on the “Deploy Project” option / “Add New Site” (if we have existing projects already)



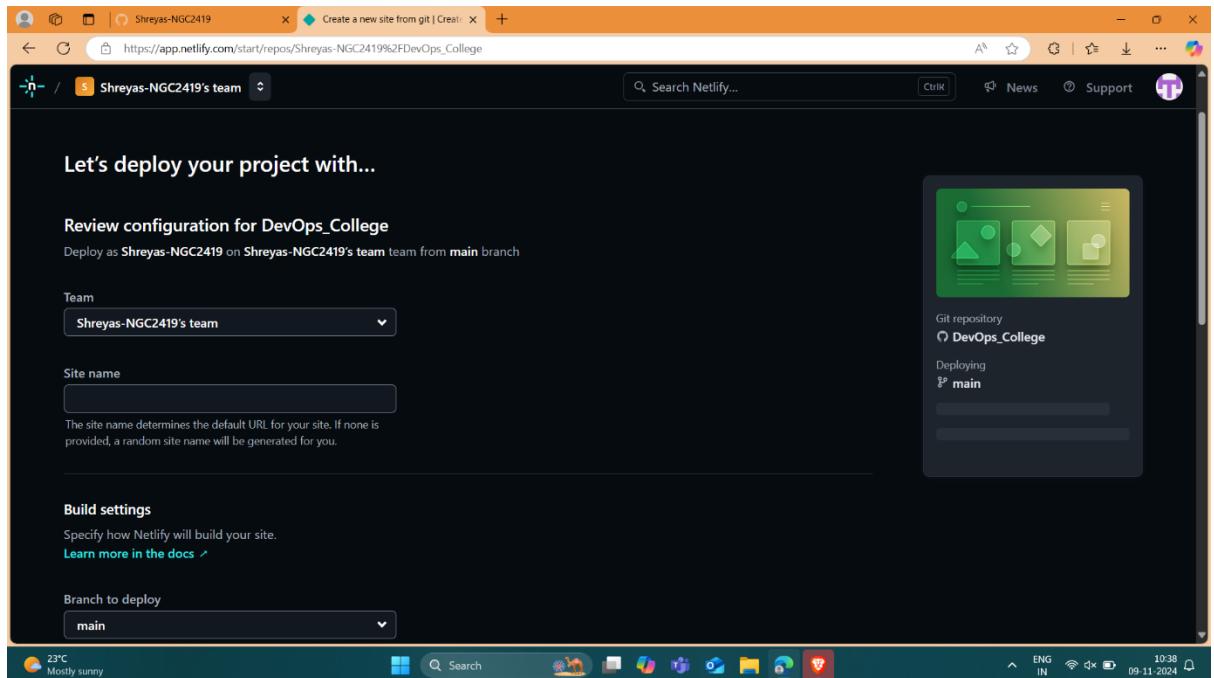
3. Choose any way to deploy the project by selecting from any of the ways provided there.



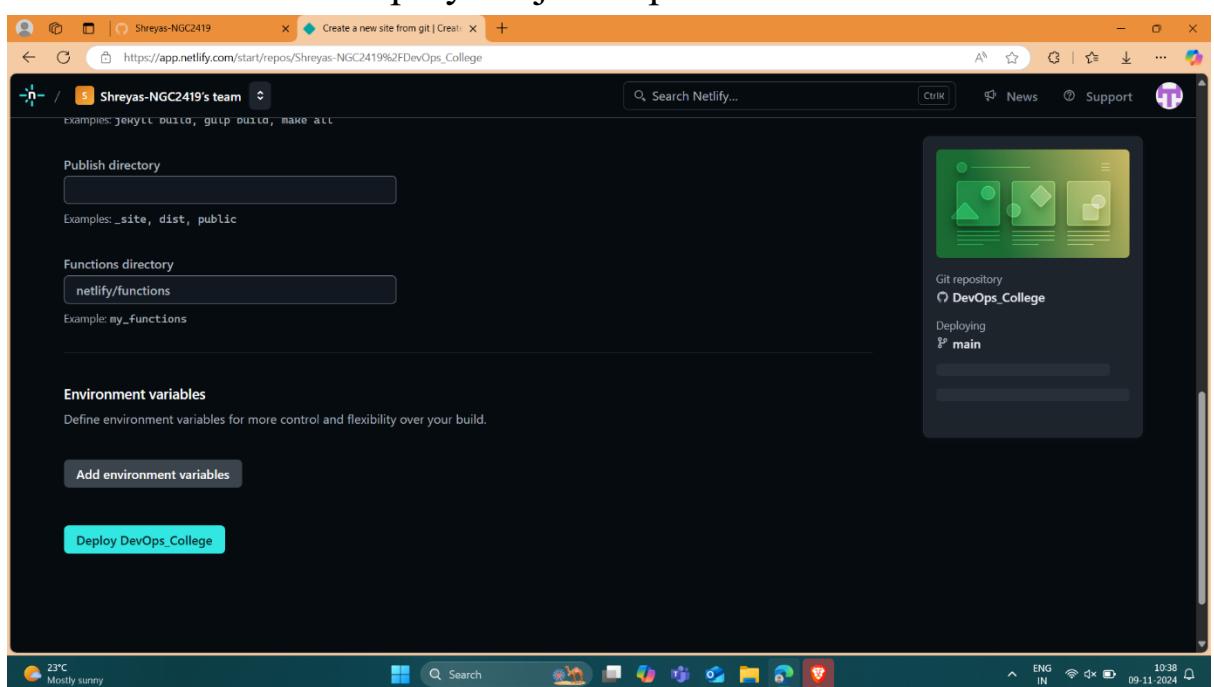
4. Here, we have selected our option as “GitHub.” After this, we must authorize our GitHub account by logging in. Select the repository from the options provided below where the actual project that is to be deployed lies in.



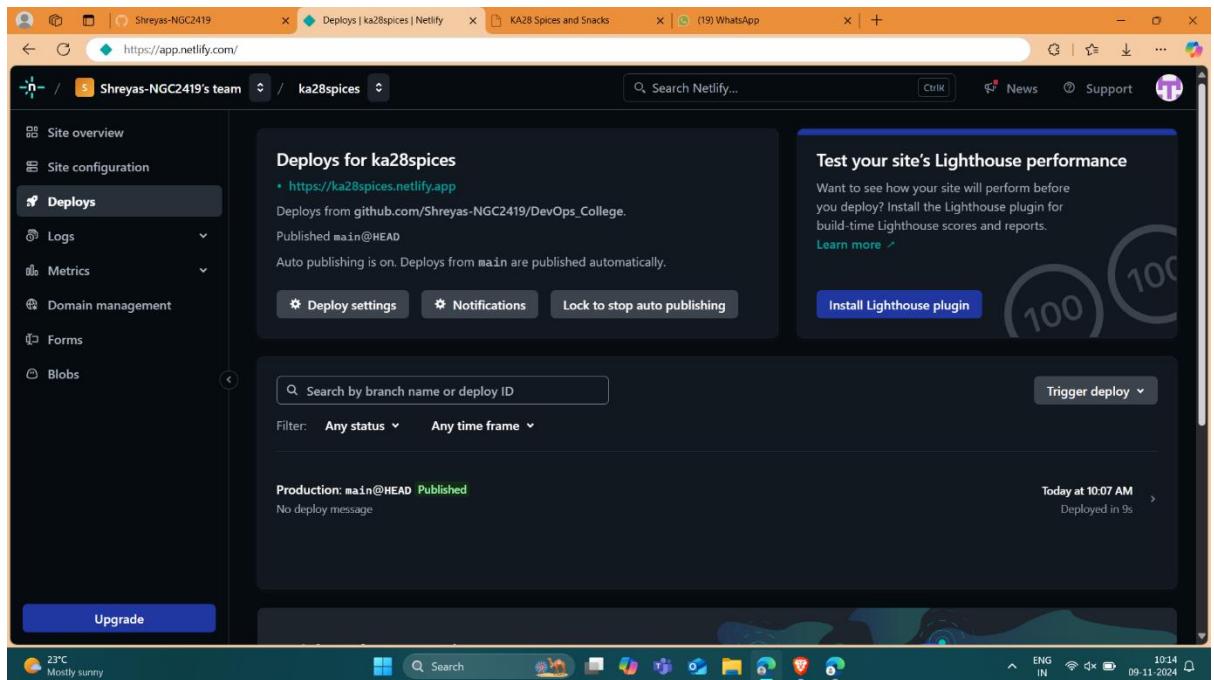
5. If we want to deploy our project with custom URLs, we can do so by providing it in the given field below. Else, an automatic link is generated by the system.



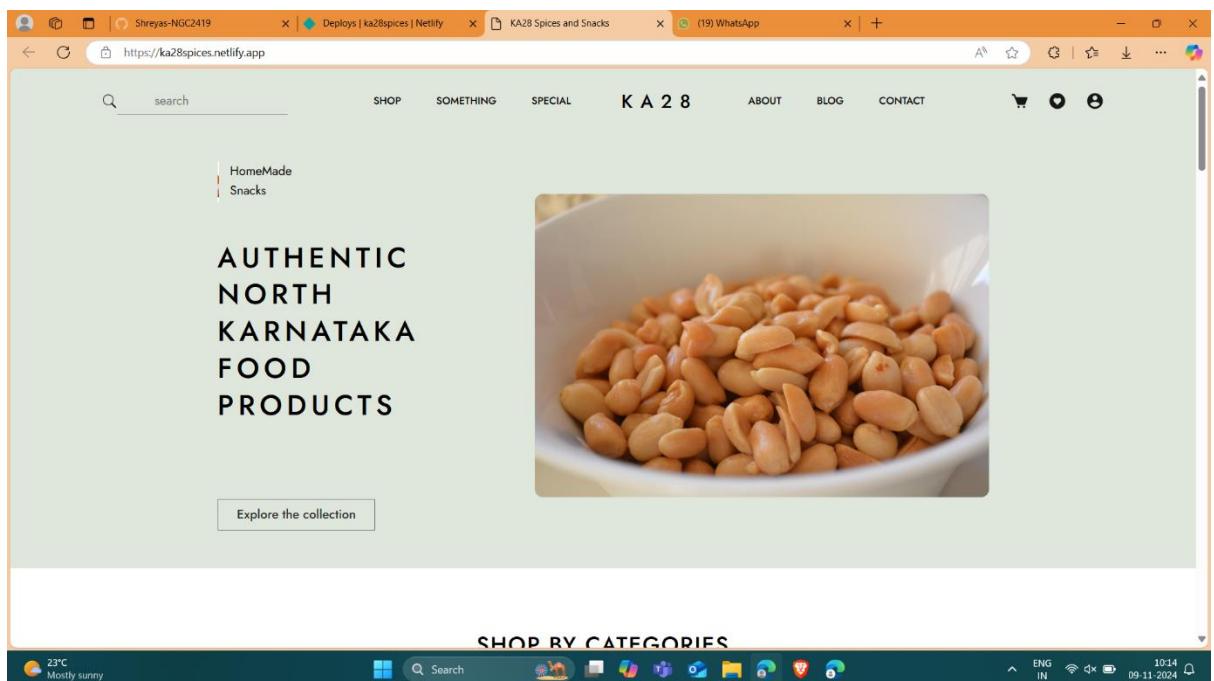
6. If the project is ready for deployment, then scroll down till the end and select the “Deploy Project” option.



- Now, after sometime, the project will be deployed and we will get the active URL which on clicking leads us to our deployed project page.



- The main page of the project is shown below, which is deployed successfully using Netlify. (Observe the presence of “netlify.app” in the URL)



UPLOADING OF THE FILES AND DEPLOYMENT USING GITHUB

1. Login to your GitHub Account and Create a new repository for your project or upload in the repository you already have (which is empty).
2. In that new repository, upload the project files.

The screenshot shows a GitHub repository page for a user named 'Shree-23012004' with a repository titled 'hellocoms'. The repository is public and contains several files uploaded by the user:

- Shree-23012004 Update login.js (uploaded 9 hours ago)
- carts Add files via upload (uploaded 10 hours ago)
- homepage Update index.html (uploaded 9 hours ago)
- loginpage Update login.html (uploaded 9 hours ago)
- styles Add files via upload (uploaded 10 hours ago)
- index.html Update index.html (uploaded 9 hours ago)

On the right side of the repository page, there is an 'About' section with the following details:

- this is a ecom web done by using dockers , jenkins and kubernetes,
- Activity (0)
- 0 stars
- 1 watching
- 0 forks

Below the 'About' section is a 'Releases' section which states "No releases published" and "Create a new release".

Under the 'Releases' section is a 'Packages' section which states "No packages published" and "Publish your first package".

Further down is a 'Deployments' section showing a green checkmark next to "github-pages 9 hours ago" and "+ 20 deployments".

At the bottom of the page is a 'Languages' section showing a horizontal bar with the following data:

- HTML 54.1%
- CSS 44.7%
- JavaScript 1.2%

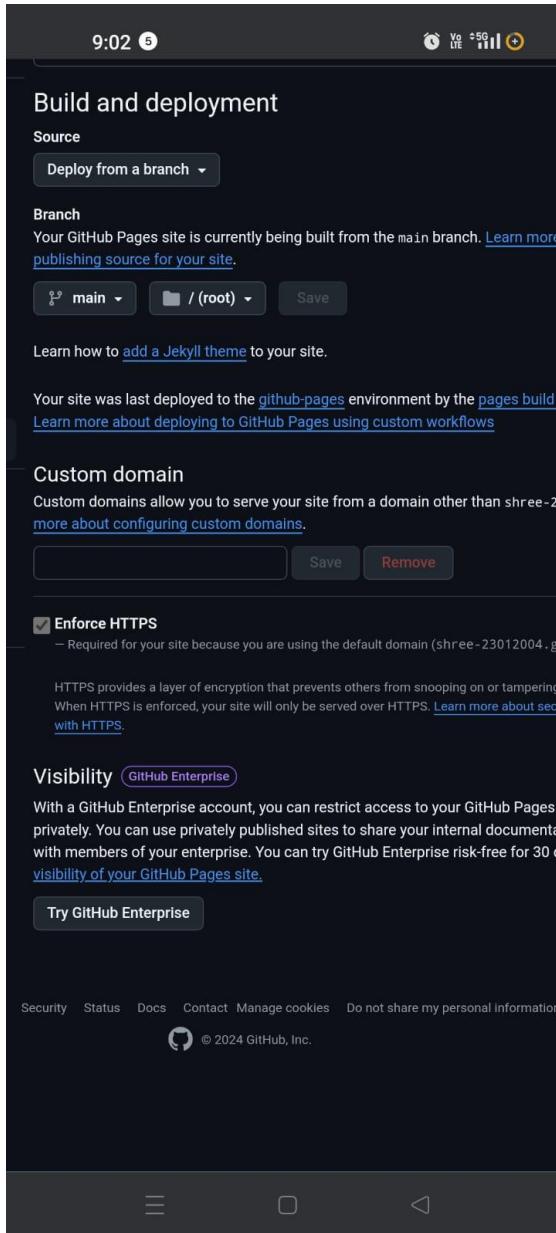
At the very bottom of the page, there are links for Terms, Privacy, Security, Status, Docs, Contact, Manage cookies, and Do not share my personal information, along with the GitHub logo and the text "© 2024 GitHub, Inc."

3. Select the settings and select pages in that.

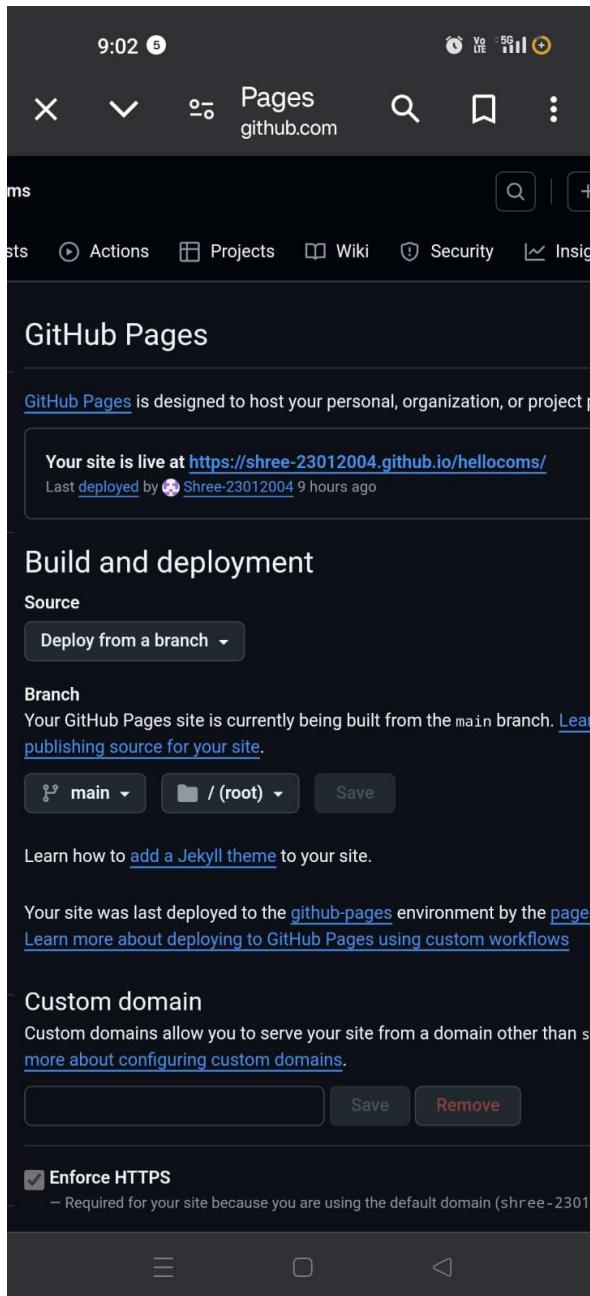
The screenshot shows the 'Moderation options' section of a GitHub repository's settings. On the left, a sidebar lists various moderation tools: Code and automation (Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces), Pages, Security (Code security, Deploy keys, Secrets and variables), and Integrations (GitHub Apps, Email notifications). The main area displays several configuration sections:

- Template repository**: A section about template repositories, with a link to 'Template repositories let users generate template repositories'.
- Require contributors to sign off on code changes**: A checkbox that, when enabled, requires contributors to affirm that the code commits are automatically merged.
- Default branch**: A section where the 'main' branch is selected.
- Social preview**: A section for customizing the repository's social preview image, with a placeholder 'Upload an image to customize your repository's social preview' and a 'Download template' button.
- Features**: A section containing two checkboxes:
 - Wikis**: Wikis host documentation for your repository.
 - Restrict editing to collaborators**: A feature that limits editing to repository collaborators.

4. In Source: Select Deploy from the branch. And in branch select main and root. And click on save.

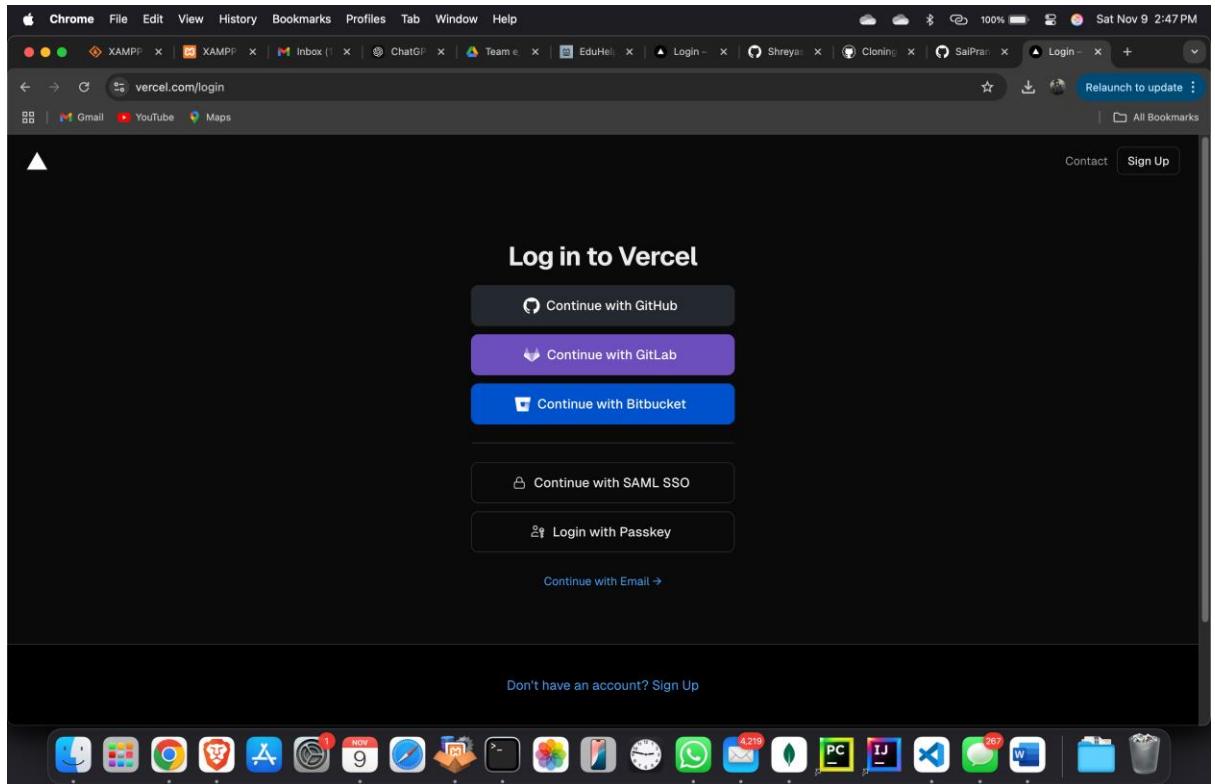


5. Wait for the link to be generated. After some time, you will get the link where your project is deployed.

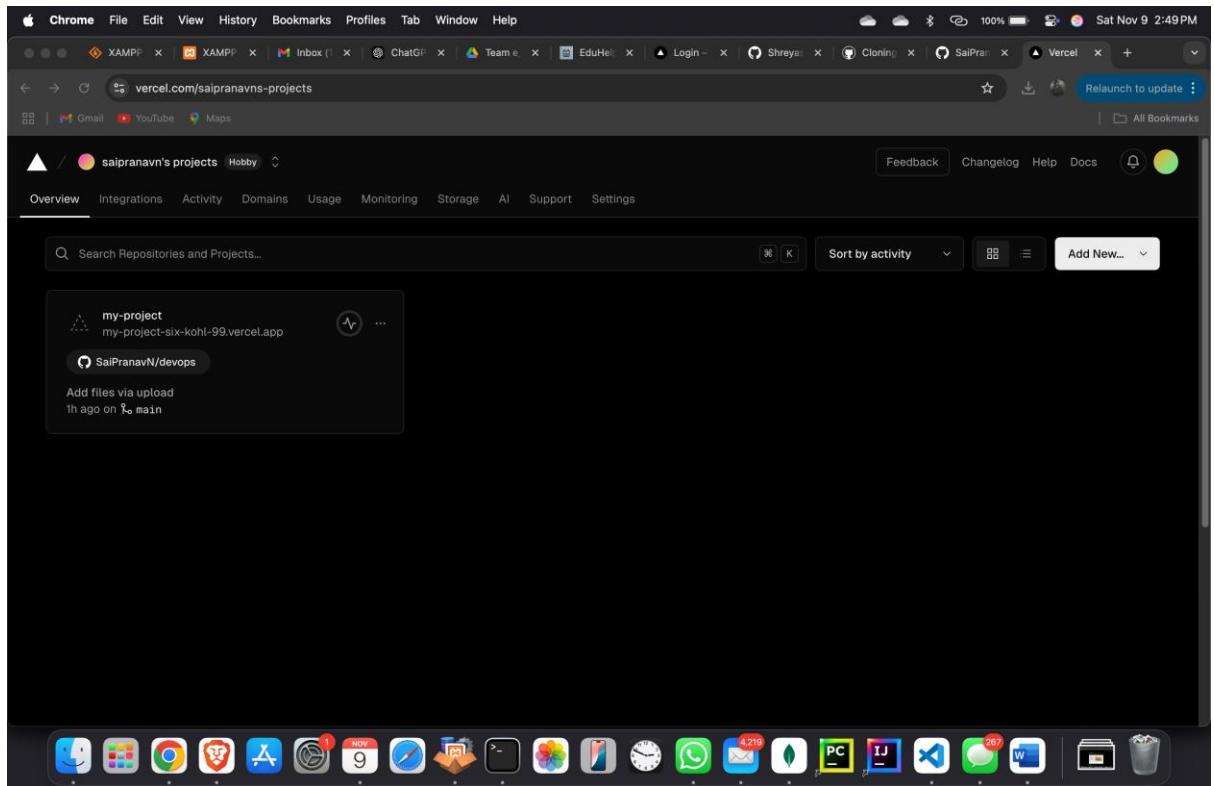


UPLOADING OF THE FILES AND DEPLOYMENT USING VERCEL

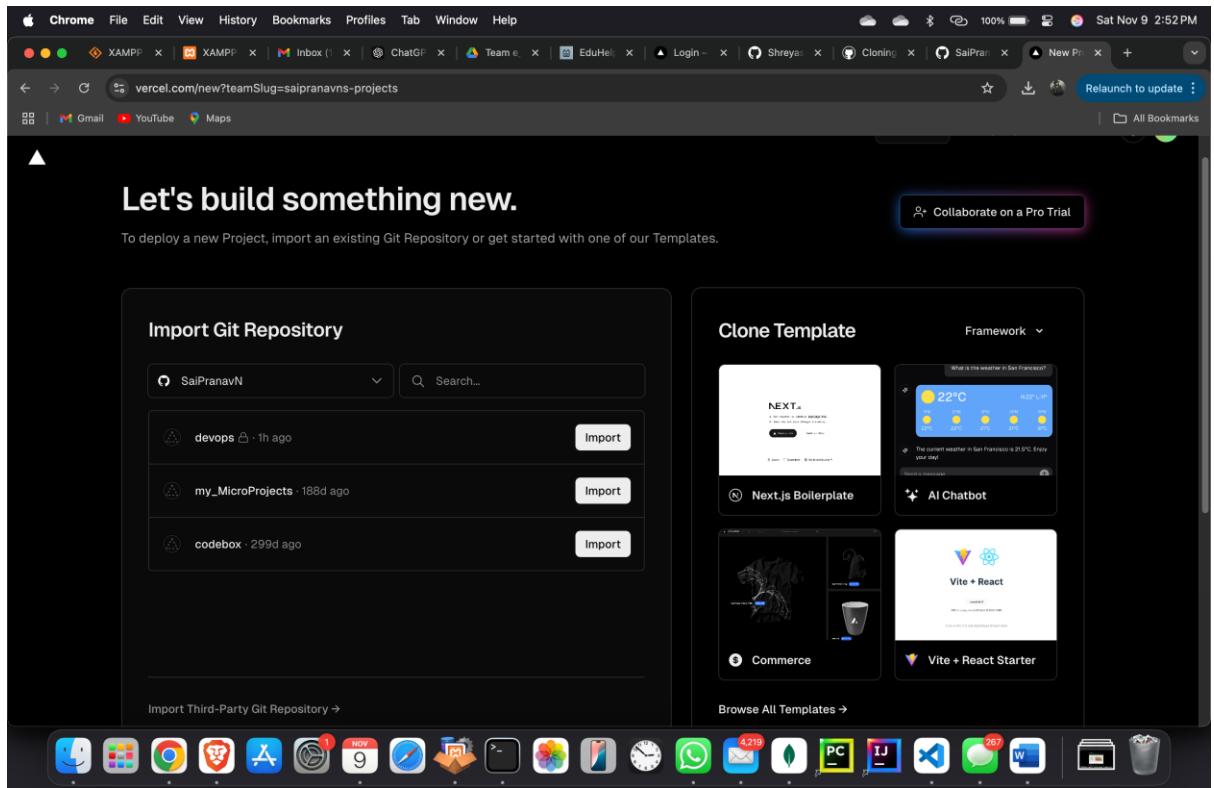
1. Login to your Vercel account (preferably using GitHub option).



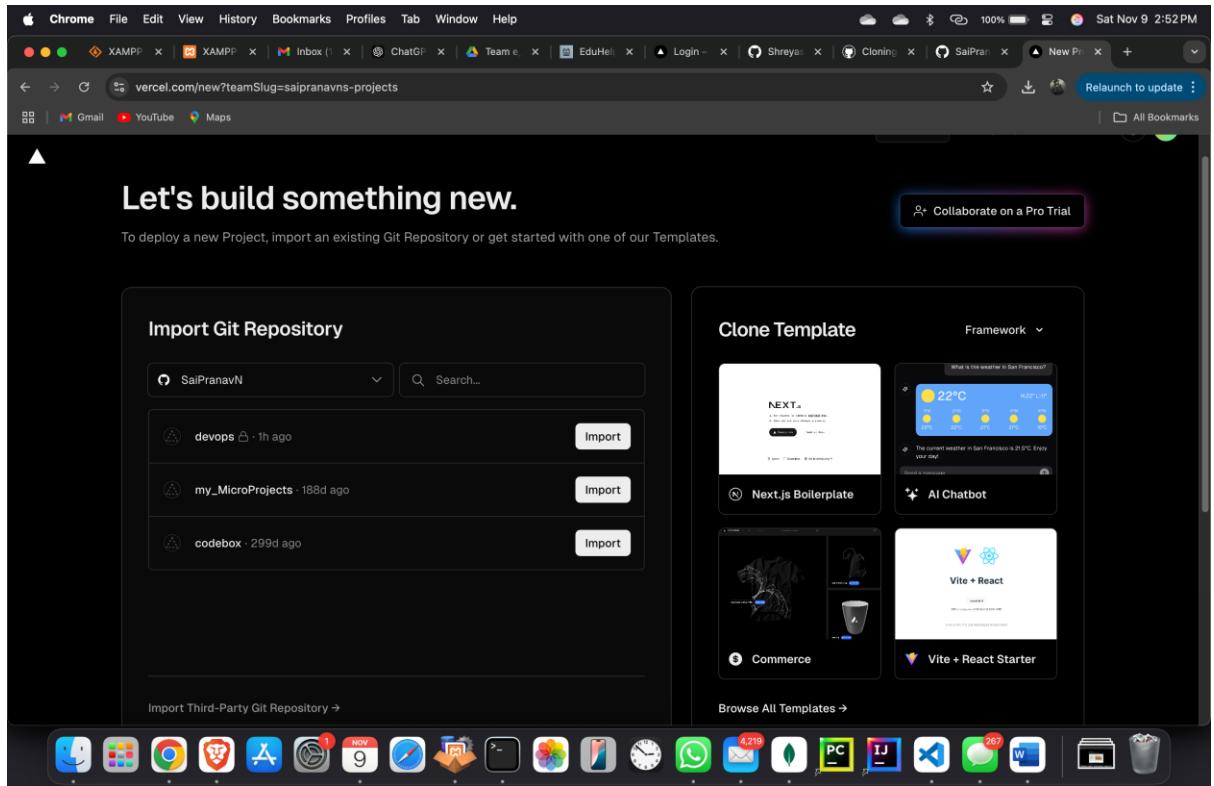
2. Click on the “Deploy Project” option / “Add New Site” (if we have existing projects already)



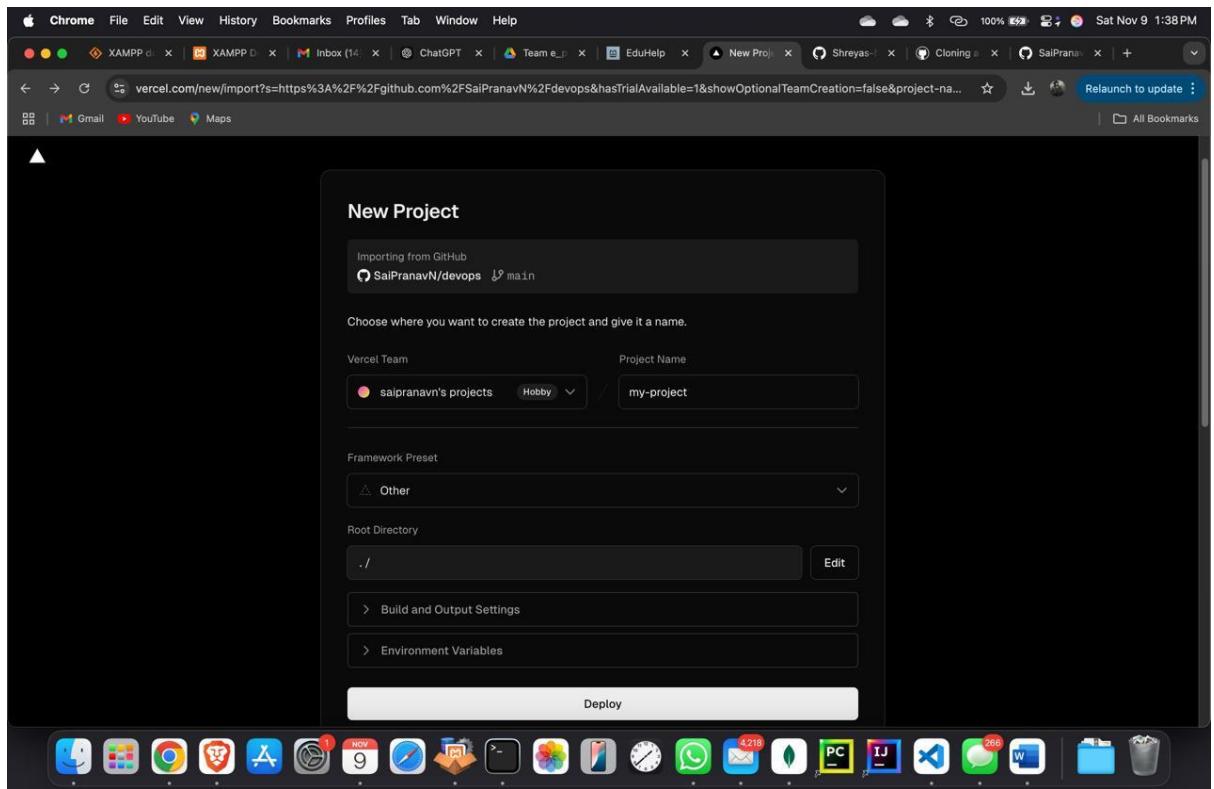
3. Choose any way to deploy the project by selecting from any of the ways provided there.



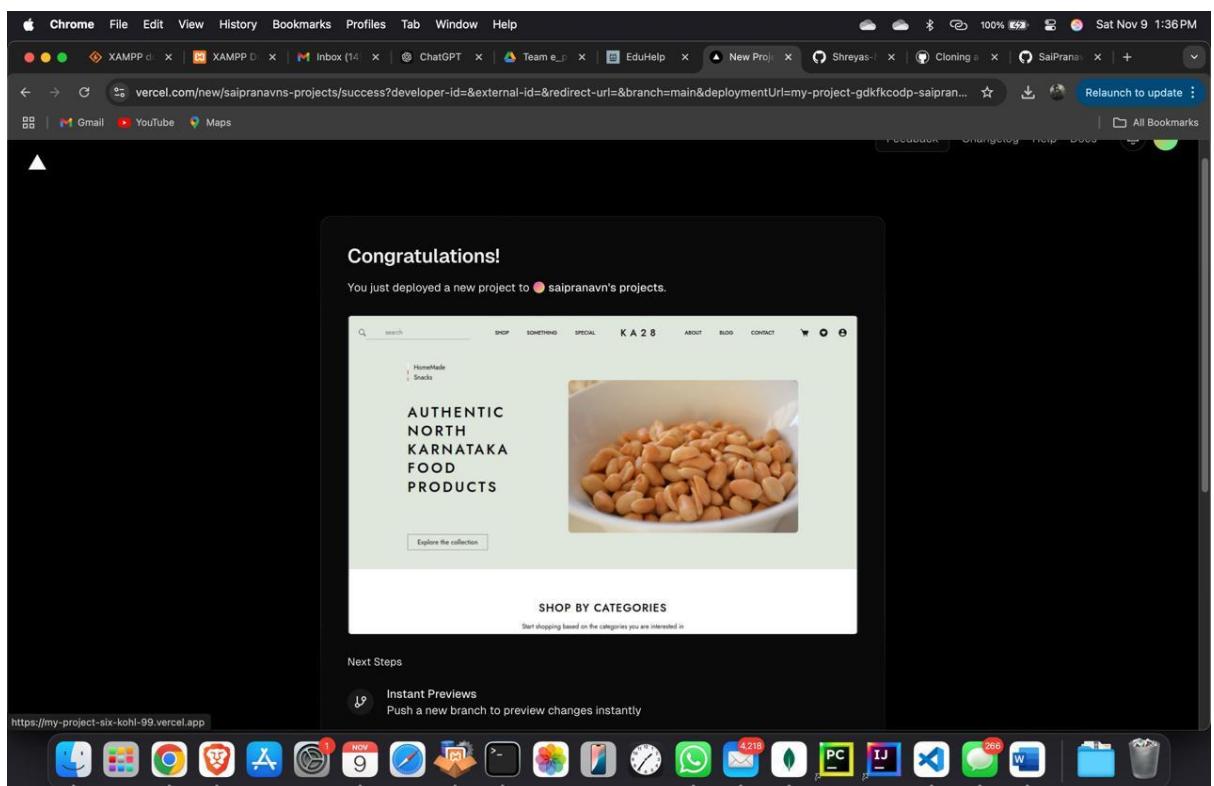
4. Here, we have selected our option as “GitHub.” After this, we must authorize our GitHub account by logging in. Select the repository from the options provided below where the actual project that is to be deployed lies in.



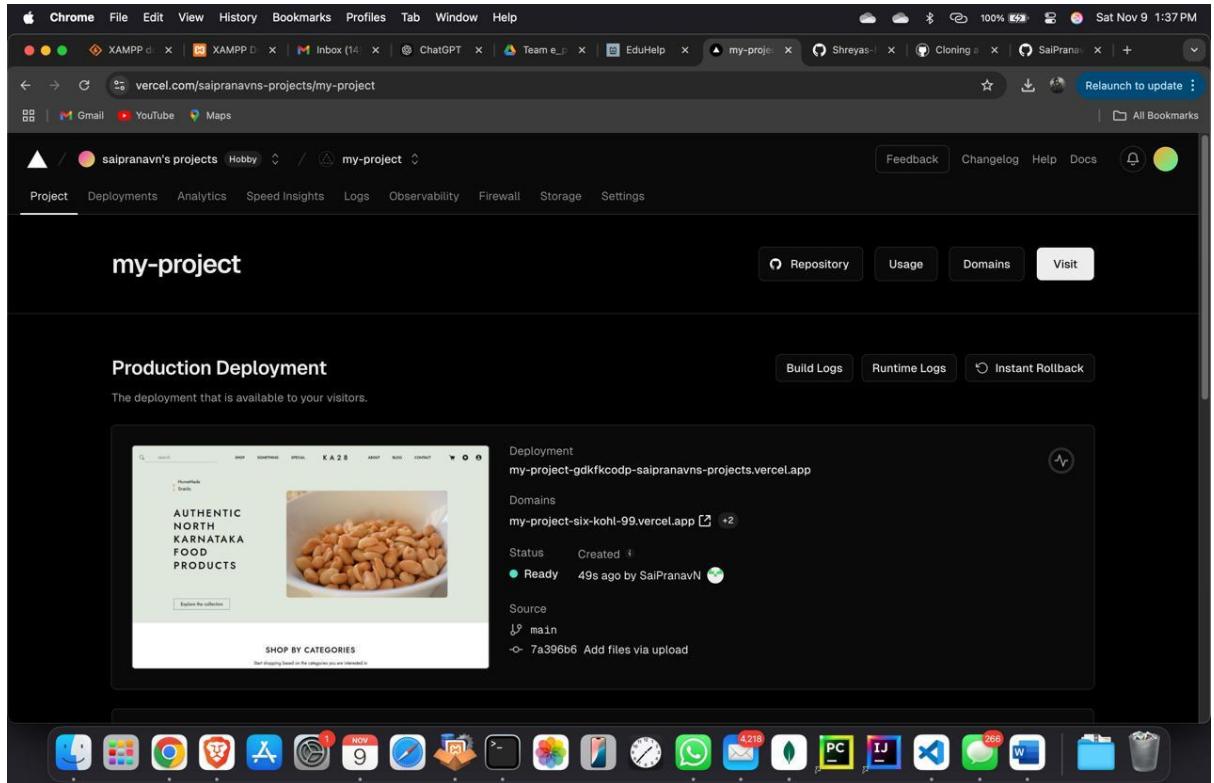
5. Set the name of the project and click on deploy



6. Upon clicking on deploy, the project will be deployed. Click on dashboard to get the link



- Now, the project will be deployed and we will get the active URL which on clicking leads us to our deployed project page.



8. The main page of the project is shown below, which is deployed successfully using Vercel.

