

Internship Report

On

“Web & App Development

@

The Sparks Foundation ~ Singapore”

CONTENTS

1. Certificate Of Selection
2. Certificate Of Internship Completion
3. Recommendation Letter from Managing Director
4. About The Sparks Foundation & Tasks Assigned
5. Works Carried Out in Task-1
6. Works Carried Out in Task-2
7. Works Carried Out in Task-3

Certificate Of Selection



THE SPARKS FOUNDATION



THIS IS PRESENTED TO

RISHAV PANDEY

PRANAV DUBEY

DIRECTOR

26-APR-2021

DATE

for successful selection as an intern at The Sparks
Foundation for function Mobile App Development.



CODE : D888S9M5Y5

Verify at:

<https://truecertificates.com/verification>

Certificate Of Internship Completion



THE SPARKS FOUNDATION

INSPIRE, INNOVATE, INTEGRATE

CERTIFICATE OF COMPLETION

This Certificate is presented to

Rishav Pandey

for an outstanding contribution during the session (May 2021 - Jun 2021) of
Graduate Rotational Internship Program at The Sparks Foundation on 01-Jun-2021.



Certificate Number: GFST6RJMPN

Verification at:
<https://truecertificates.com/verification/>

Pranav Dubey
PRANAV DUBEY

MANAGING DIRECTOR

**Recommendation Letter from the Managing Director of TSF,
Singapore for outstanding performance in GRIP.**



THE SPARKS FOUNDATION

INSPIRE, INNOVATE, INTEGRATE

TO WHOM IT MAY CONCERN

This is to certify that Rishav Pandey has successfully completed the Graduate Rotational Internship Program at The Sparks Foundation for a duration of one month from May 2021 to Jun 2021 as a Tech Intern.

Rishav Pandey did an excellent job in this position and was an asset to our organization during his internship tenure. He has excellent analytical skills and he was able to learn new skills and adapt to new technologies in a very short span of time, thus being productive quickly.

Rishav Pandey was always willing to offer his assistance and had excellent rapport with his team members. He would be an asset to any employer and I recommended him for any endeavor he chooses to pursue.

REGARDS,

PRANAV DUBEY

MANAGING DIRECTOR

SINGAPORE:
THESPARKSFUNDATION.SG

INDIA:
THESPARKSFUNDATION.IN

GLOBAL:
THESPARKSFUNDATION.INFO



Certificate Number: A3LDKKEWMN

Verification at:

<https://truecertificates.com/verification/>

The Sparks Foundation

Sparks Foundation is a Singapore based IT company. It runs a Graduate Rotational Internship Programme every year. In this internship programme they hire STEM students for technical tasks.

In the month of May'21, I got selected to work as a **Web & App Development Intern** for one month. Tasks which I completed during my entire tenure are summarized below:

Task: 1

Payment Gateway Integration

- Create a simple website where payment gateway is integrated.
- There will be a simple donate button on homepage. On clicking the donate button, the user will land on the payment page where user can select the amount to be paid and the payment type, e.g. credit card, Paypal, etc.
- Once the payment is done and invoice will be generated and email will be sent to the user for the payment received. The invoice will contain the amount.
- On any page / email, only basic information is needed.
- Create your own temporary / sandbox / testing accounts with 3rd party for integrations.
- Host the website at 000webhost, github.io, heroku app or any other free hosting provider. Check in code in gitlab.

Task: 2

Social Media Integration

- Create a mobile app, where user can login through at least two social media from such as Facebook and Google.
- After login, display all the details (e.g. Name, profile photo, email, etc.) on the second page.
- Take help of online tutorials and You tube videos.
- No backend / server-side programming required.
- Very good-looking UI and responsive UI, which should work for mobiles as well as tablets.
- Clean code is a must.
- Upload video demo of your application on you tube and submit the url.

Task: 3

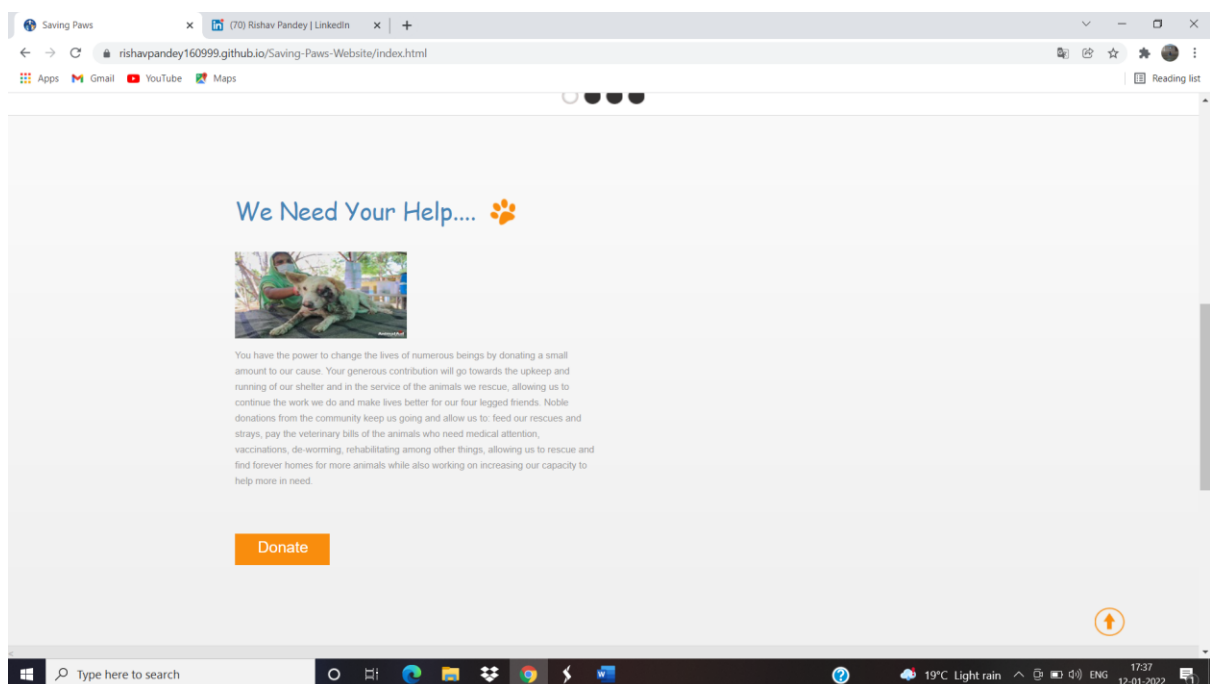
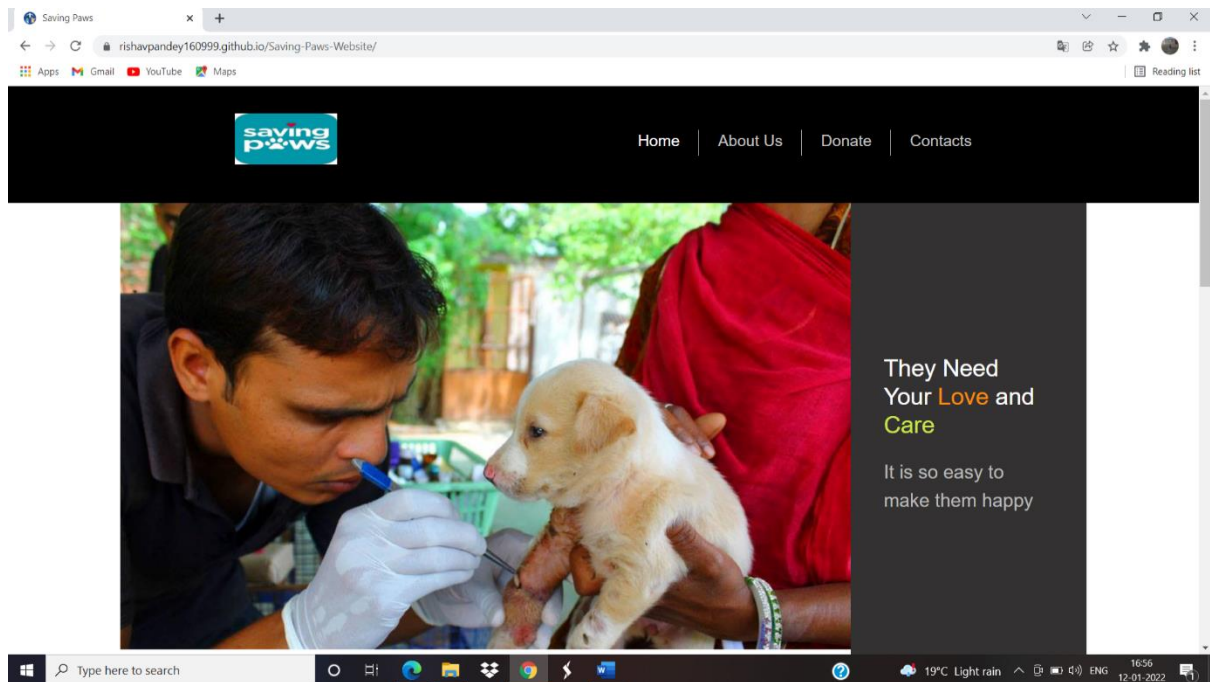
CI/CD: Cloud Computing

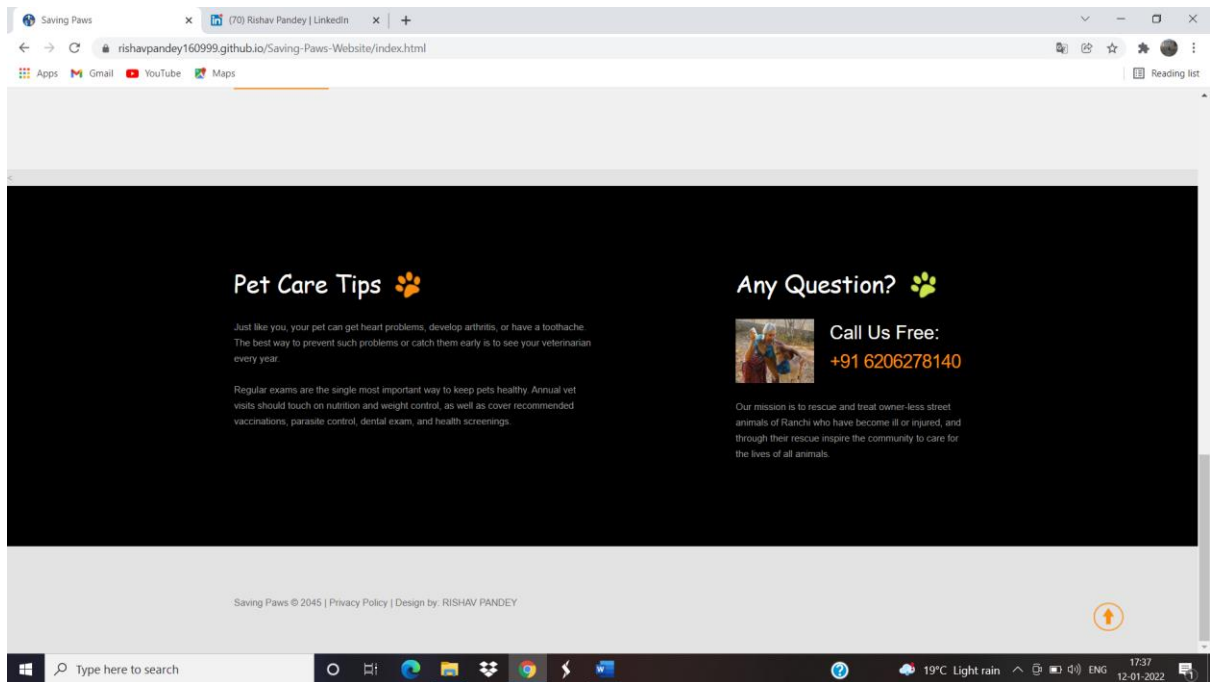
- Read up about AWS or Azure.
- Write up about the steps of setup and essentials of AWS EC2 or Azure VM (one page step by step).
- Create an EC2 or azure VM instance and access it through ssh from your pc over internet.
- In the EC2, deploy and run any application (a website with tomcat/spring boot) or python-based project.
- Use at least one service apart from EC2 or VM, i.e. Database service, or MQ, ML, Mobile or any other services provided by AWS or Azure.
- Submit the URL of the application which is running on EC2.
- Your video should show that you are able to run applications on cloud.

Works Carried Out In Task: 1

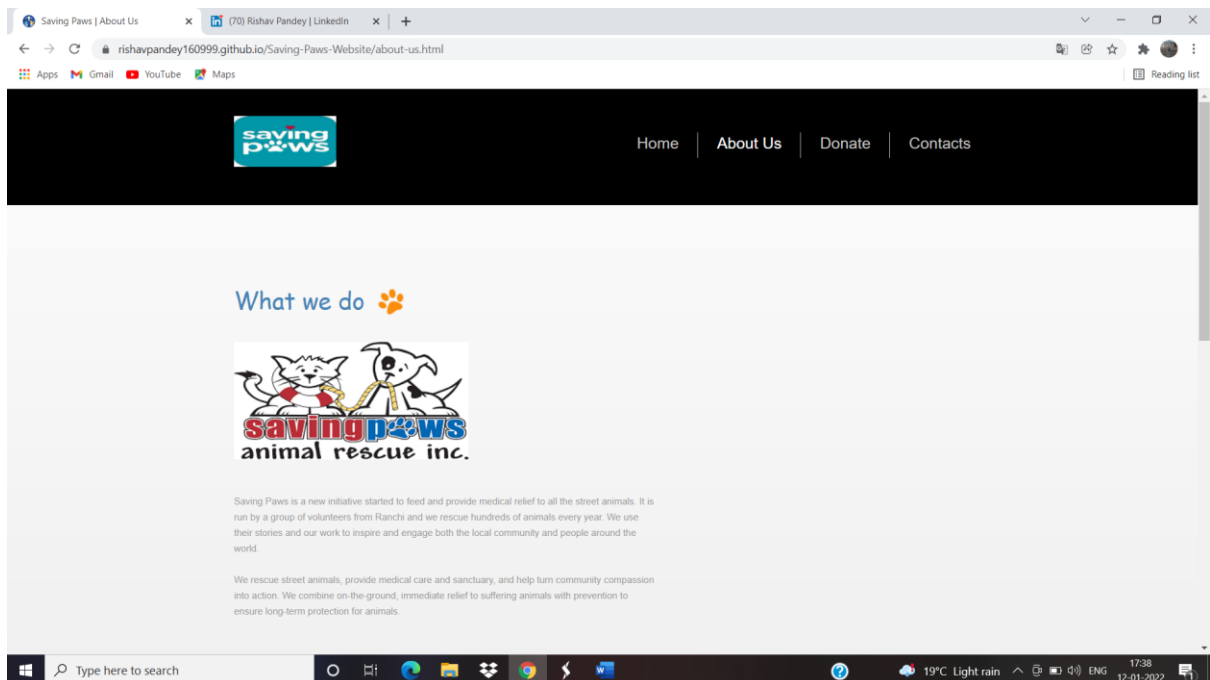
Designed an animal donation website named 'Saving Paws' and then integrated a payment gateway page in it. There is a donate button on home page, just by tapping the same user is directed to the payment page. Once the payment is done, an invoice is generated and email is sent to the user for the payment received. The invoice contains the amount.

Home Page:

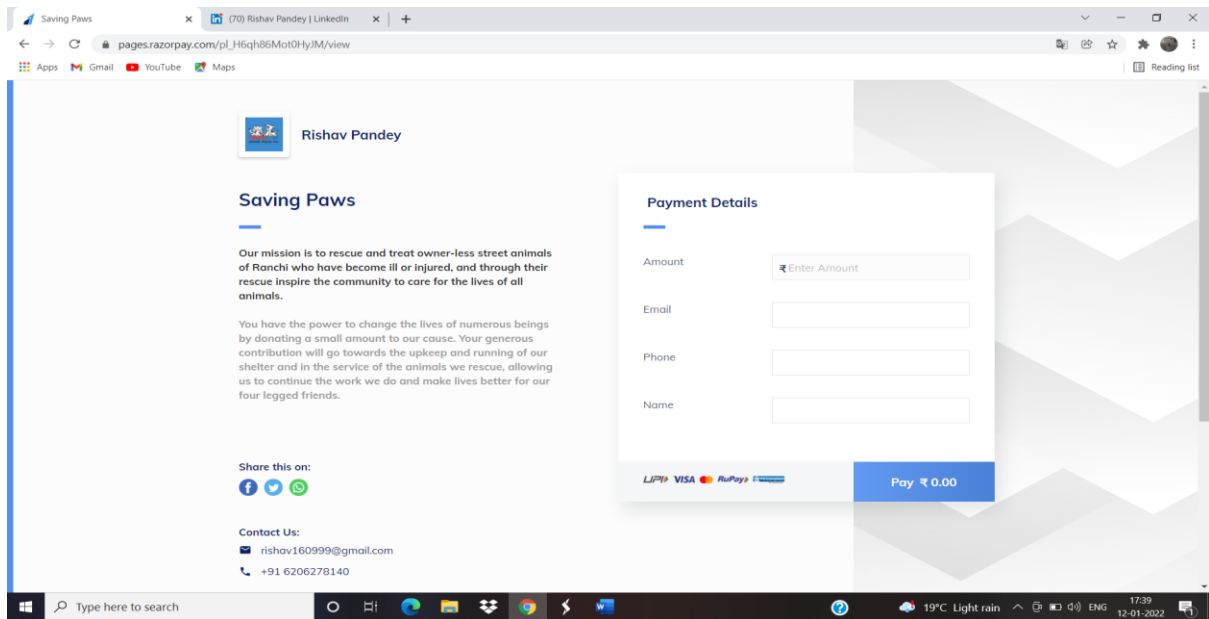




About Us Page:



Donate Page:



The screenshot shows a web browser window with the URL `pages.razorpay.com/pl_H6qh86Mot0HyIM/View`. The page features a profile for Rishav Pandey with a mission statement: "Our mission is to rescue and treat owner-less street animals of Ranchi who have become ill or injured, and through their rescue inspire the community to care for the lives of all animals." It also includes a "Payment Details" modal form with fields for Amount, Email, Phone, and Name, and a "Pay ₹ 0.00" button. Social media sharing icons and contact information are also present.

Saving Paws

Our mission is to rescue and treat owner-less street animals of Ranchi who have become ill or injured, and through their rescue inspire the community to care for the lives of all animals.

You have the power to change the lives of numerous beings by donating a small amount to our cause. Your generous contribution will go towards the upkeep and running of our shelter and in the service of the animals we rescue, allowing us to continue the work we do and make lives better for our four legged friends.

Share this on:

Contact Us:

Payment Details

Amount: ₹ Enter Amount

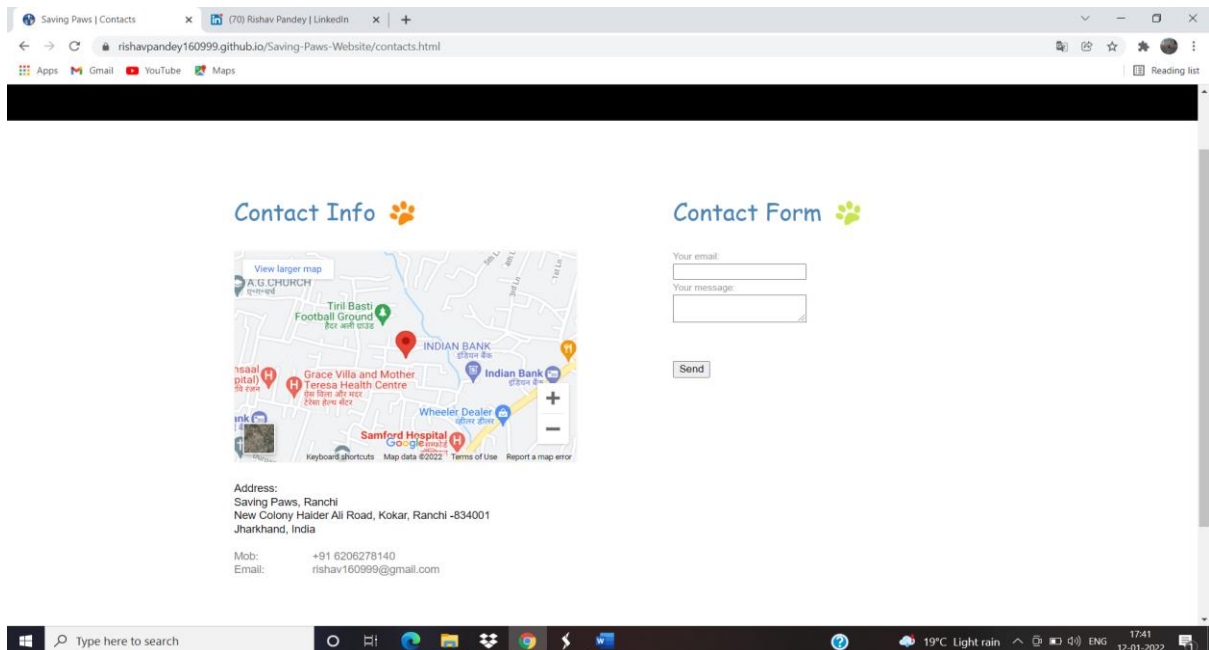
Email:

Phone:

Name:

Pay ₹ 0.00

Contact Us Page:



The screenshot shows a web browser window with the URL `rishavpandey160999.github.io/Saving-Paws-Website/contacts.html`. The page is divided into two main sections: "Contact Info" and "Contact Form". The "Contact Info" section includes a map of Ranchi, India, and the following address: "Saving Paws, Ranchi, New Colony Haider Ali Road, Kokar, Ranchi -834001, Jharkhand, India". The "Contact Form" section has fields for "Your email" and "Your message", and a "Send" button.

Contact Info

Contact Form

View larger map

A.G. CHURCH

Tini Basti Football Ground

INDIAN BANK

Grace Villa and Mother Teresa Health Centre

Wheeler Dealer

Samford Hospital

Address:

Saving Paws, Ranchi
New Colony Haider Ali Road, Kokar, Ranchi -834001
Jharkhand, India

Mob: +91 6206278140
Email: rishav160999@gmail.com

Your email:

Your message:

Send

[Link of the website](#)

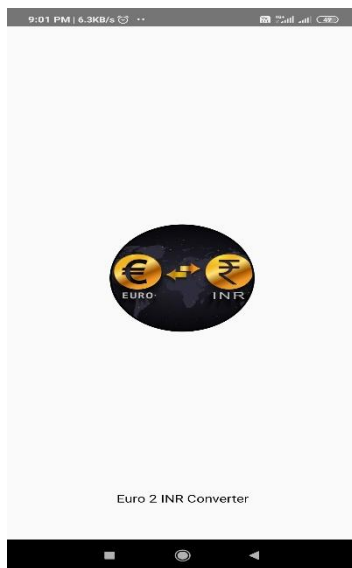
[GitHub Repo](#)

[Presentation](#)

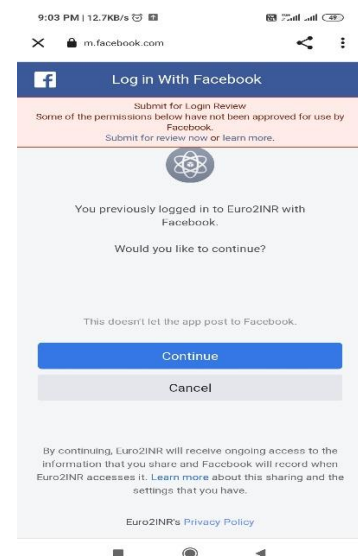
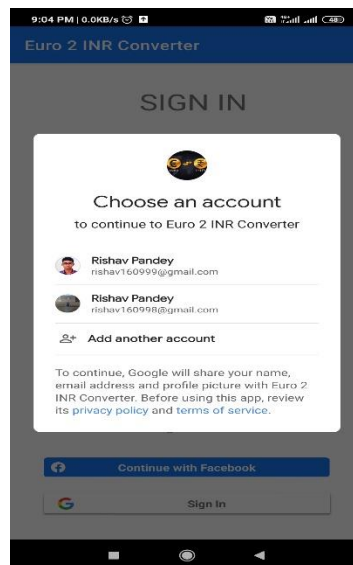
Works Carried Out In Task: 2

Designed an android app named "Euro 2 INR Converter" which converts a given amount in Euro into INR. For creating the same I used JAVA and tool like Android Studio. I've integrated Google and Facebook Sign-in in it, so that a user can sign in by any of the two ways. Once the user is signed in, their name, email, and profile picture is displayed on the next screen. User can further continue to the next interface i.e. "Euro 2 INR Converter" page.

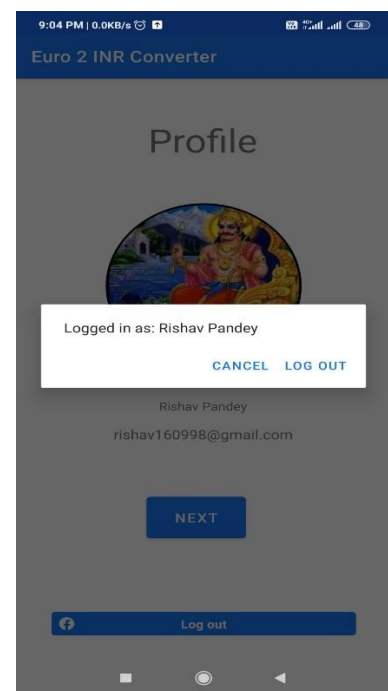
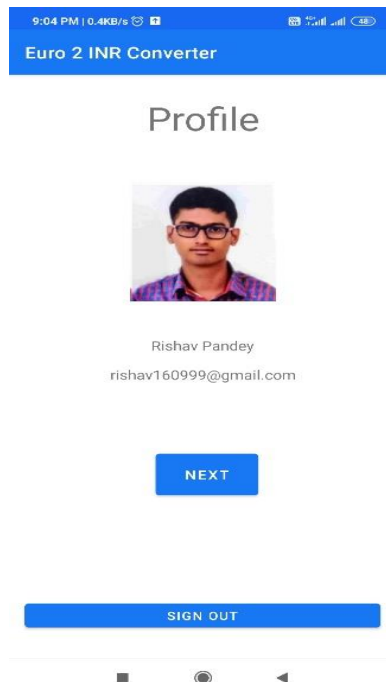
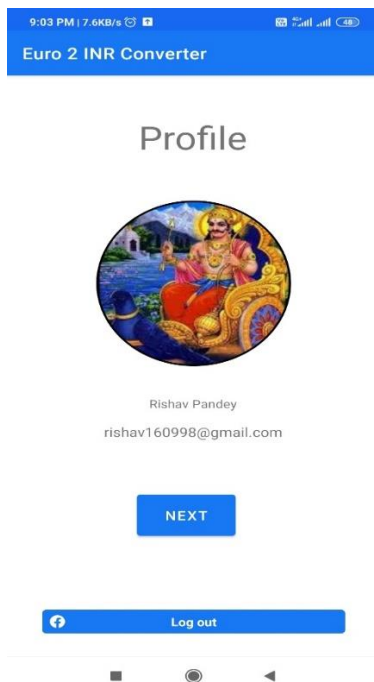
Splash Screen:



Sign-in Page:



Profile Page:



Euro-2-INR Converter Page:



[GitHub Repo](#)

[Presentation](#)

Works Carried Out In Task: 3

Learnt about all the services offered by an extremely secure cloud service provider Amazon Web Services (AWS) and implemented practically by launching an AWS EC2 Instance. I even used other services of AWS like storage services (AWS S3) and security services (IAM). After launching the instance, I secure shelled into the same and further hosted my portfolio website on that virtual server.

Steps to host a website on AWS EC2 Instance:

Step 1: Choose an Amazon Machine Image (AMI).

The screenshot shows the AWS Management Console's 'Launch Instance wizard' for EC2. The first step, 'Choose an Amazon Machine Image (AMI)', is active. It displays a search bar and a list of AMIs. The 'Quick Start' section on the left includes 'My AMIs', 'AWS Marketplace', 'Community AMIs', and a 'Free tier only' filter. The main area shows two AMIs: 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type' and 'Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type'. The first AMI is selected, and its details are shown: 'Free tier eligible', 'Amazon Linux', 'Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.', 'Root device type: ebs', 'Virtualization type: hvm', and 'ENA Enabled: Yes'. The 'Select' button is highlighted.

Step 2: Choose an Instance Type.

The screenshot shows the AWS Management Console's 'Launch Instance wizard' for EC2. The second step, 'Choose an Instance Type', is active. It displays a table of instance types. The 'Filter by' section at the top shows 'All instance families' and 'Current generation'. The 'Currently selected' section shows 't2.micro (- ECUs, 1 vCPUs, 2.5 GHz, ~, 1 GiB memory, EBS only)'. The table lists various instance types, with 't2.micro' selected. The table columns are: Family, Type, vCPUs, Memory (GiB), Instance Storage (GB), EBS-Optimized Available, Network Performance, and IPv6 Support. The 't2.micro' row is highlighted in blue. The 'Review and Launch' button is highlighted.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

Step 3: Configure Instance Details.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network [Create new VPC](#)

Subnet [Create new subnet](#)

Auto-assign Public IP

Hostname type

DNS Hostname ☐ Enable IP name IPv4 (A record) DNS requests
☒ Enable resource-based IPv4 (A record) DNS requests
☐ Enable resource-based IPv6 (AAAA record) DNS requests

Placement group ☐ Add instance to placement group

Capacity Reservation

Domain join directory

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

Step 4: Add Storage.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0655b9215a5010620	<input type="text" value="8"/>	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

▼ Shared file systems

You currently don't have any file systems on this instance. Select "Add file system" button below to add a file system.

[Add file system](#)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

Step 5: Add Tags.

Launch instance wizard | EC2 Ma

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Search for services, features, blogs, docs, and more [Alt+S]

EC2

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum) Value (256 characters maximum) Instances Volumes Network Interfaces

This resource currently has no tags

Choose the Add tag button or [click to add a Name tag](#).
Make sure your [IAM policy](#) includes permissions to create tags.

Add Tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Type here to search

Step 6: Configure Security Group.

Launch instance wizard | EC2 Ma

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Search for services, features, blogs, docs, and more [Alt+S]

EC2

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name: launch-wizard-17

Description: launch-wizard-17 created 2022-01-12T19:38:03.709+05:30

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
SSH	TCP	22	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review and Launch

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Type here to search

Step 7: Review Instance Launch.

The screenshot shows the AWS Management Console interface for the 'Launch Instance Wizard' in the 'ap-south-1' region. The wizard is at Step 7: Review Instance Launch. The console displays the following details:

- AMI Details:** Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0af25d0df86db00c1. It includes a 'Free tier eligible' badge and a description of the AMI's support and features.
- Instance Type:** t2.micro. A table shows the instance specifications: 1 vCPU, 1 GiB Memory, EBS only storage, and Low to Moderate network performance.
- Security Groups:** launch-wizard-17. A table shows the security group rules, including HTTP, TCP, and SSH access.

At the bottom right, there are buttons for 'Cancel', 'Previous', and 'Launch'.

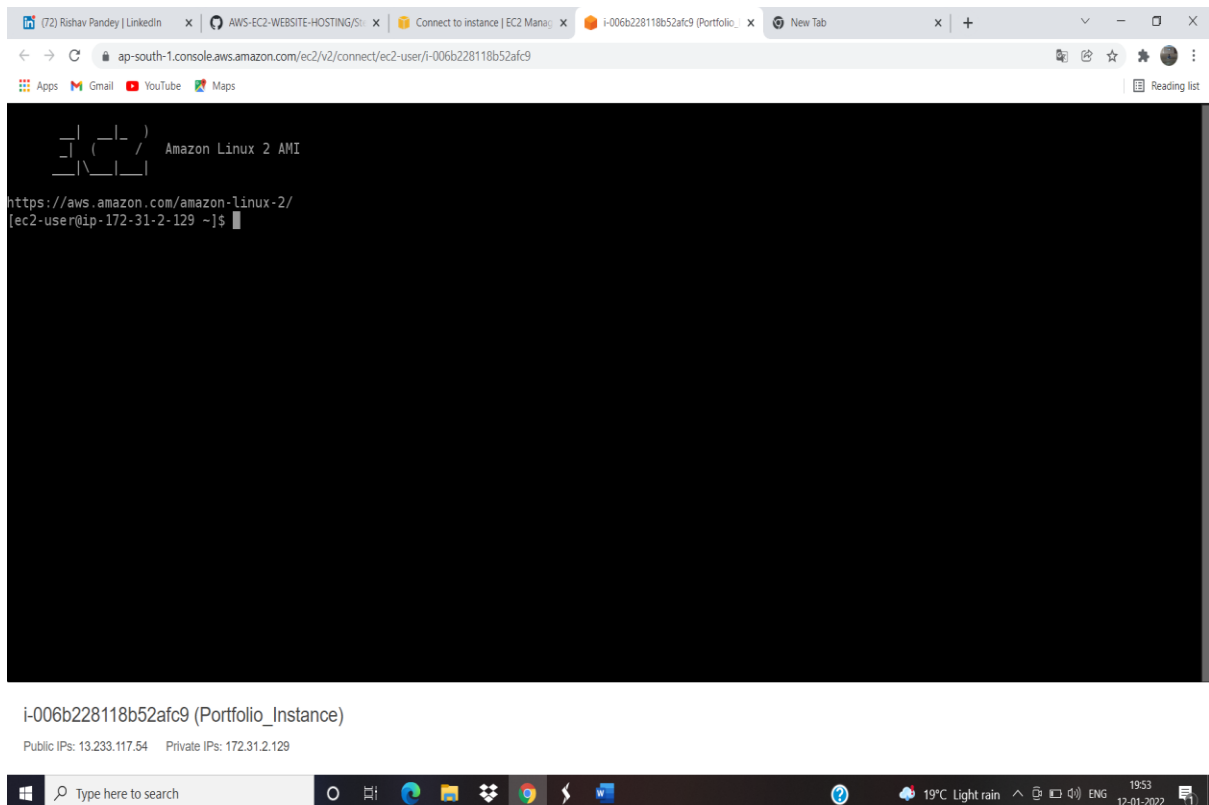
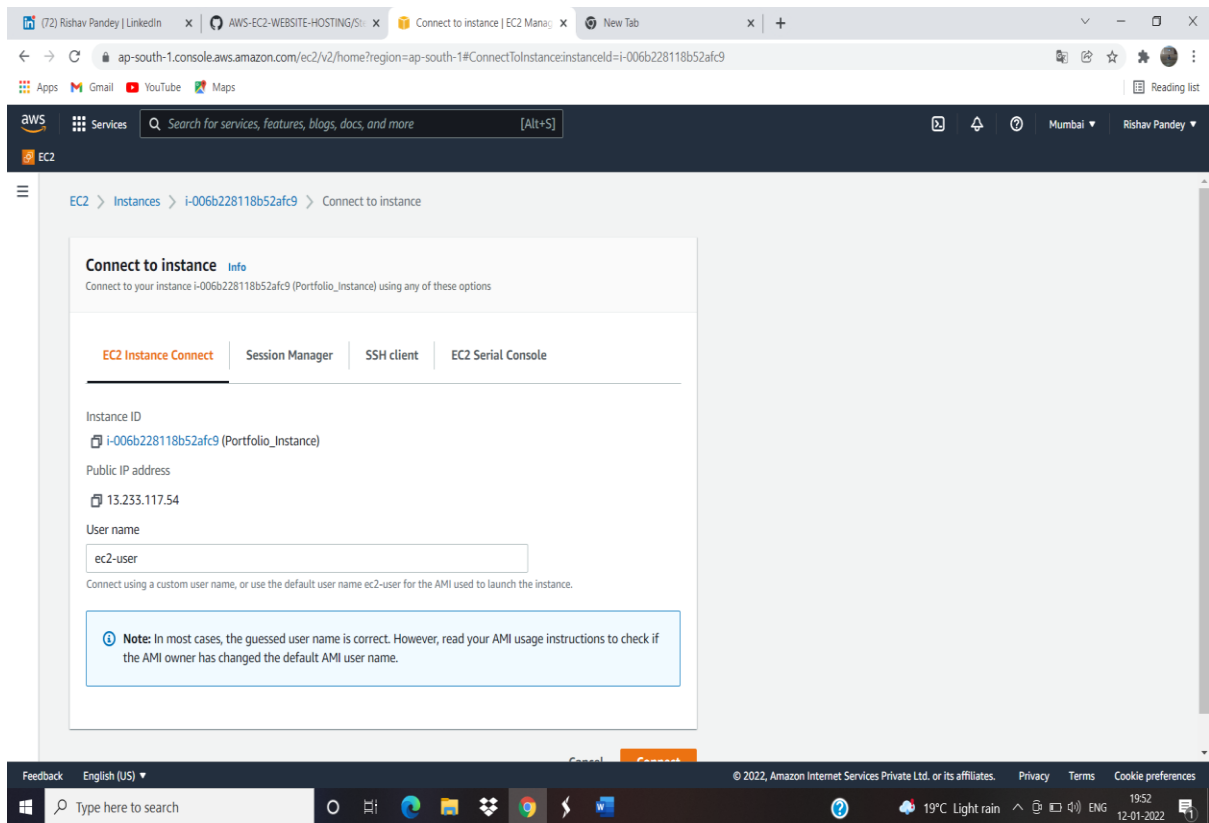
Step 8: Select an existing keypair or create a new key pair.

The screenshot shows the same AWS Management Console interface as in Step 7, but with a modal dialog open for selecting a key pair. The dialog contains the following information:

- Title:** Select an existing key pair or create a new key pair.
- Description:** A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.
- Note:** The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.
- Options:** A dropdown menu to 'Choose an existing key pair' and a 'Select a key pair' dropdown menu showing 'KEYPAIR2 | RSA'.
- Acknowledgment:** A checkbox labeled 'I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.' which is checked.
- Buttons:** 'Cancel' and 'Launch Instances'.

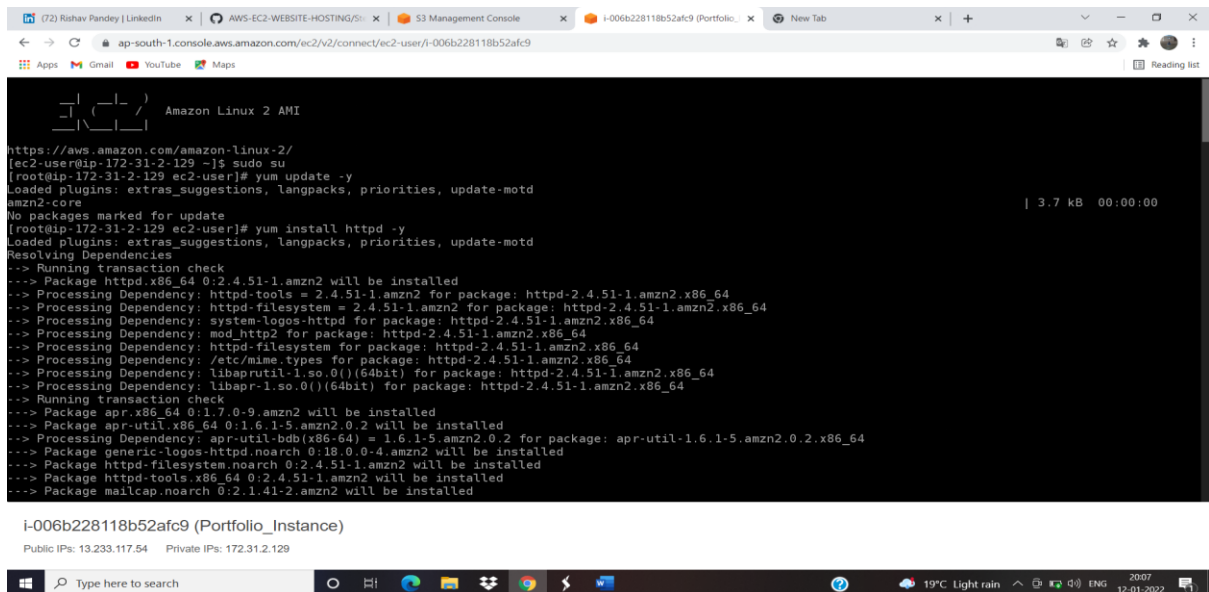
The background shows the 'Storage' section of the wizard, including volume type, device, and snapshot details.

Step 9: Connect to instance.



Step 10: Run the below mentioned SSH commands one by one.

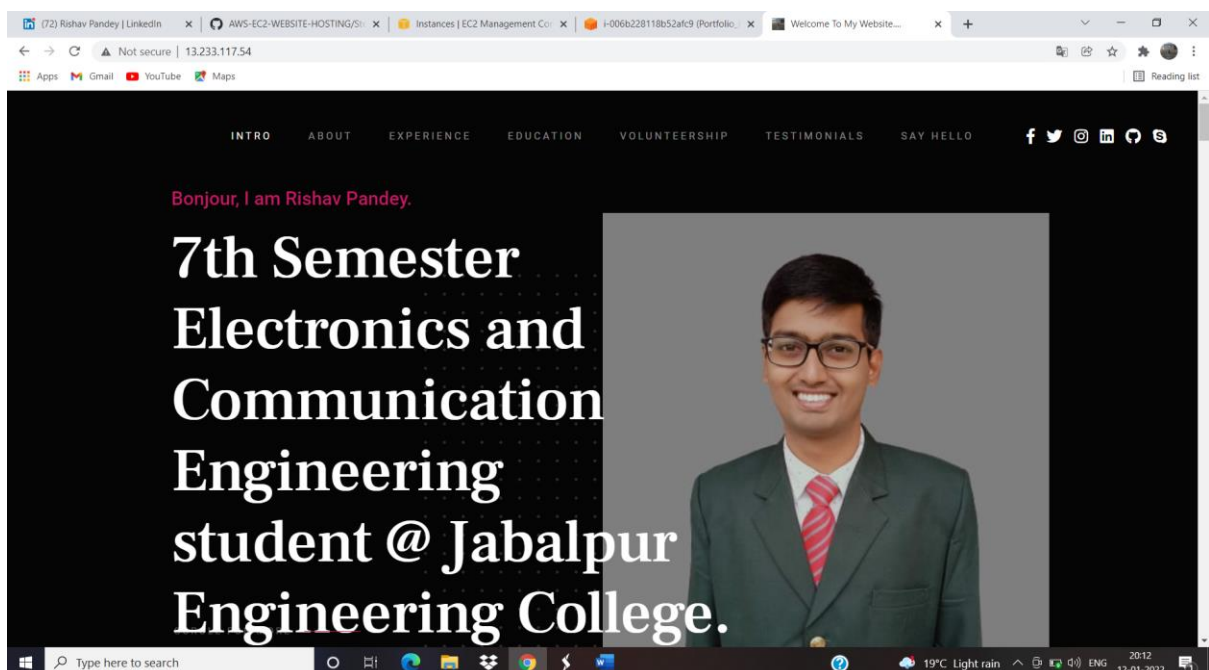
```
sudo su
yum update -y
yum install httpd -y
chkconfig httpd on
cd /var/www/html
aws s3 sync s3://BucketName /var/www/html
service httpd start
```



```
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-2-129 ~]$ sudo su
[root@ip-172-31-2-129 ec2-user]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amazon2-core | 3.7 kB 00:00:00
No packages marked for update
[ec2-user@ip-172-31-2-129 ec2-user]# yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.51-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.51-1.amzn2 for package: httpd-2.4.51-1.amzn2.x86_64
--> Processing Dependency: httpdfilesystem = 2.4.51-1.amzn2 for package: httpd-2.4.51-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.51-1.amzn2.x86_64
--> Processing Dependency: modhttp2 for package: httpd-2.4.51-1.amzn2.x86_64
--> Processing Dependency: httpdfilesystem for package: httpd-2.4.51-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.51-1.amzn2.x86_64
--> Processing Dependency: libaprutil1.so.0()(64bit) for package: httpd-2.4.51-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.51-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpdfilesystem.noarch 0:2.4.51-1.amzn2 will be installed
--> Package httpd-tools.x86_64 0:2.4.51-1.amzn2 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
```

Step 11: Copy the public IPv4 address and paste it in a new tab. Your website will be live.

I Just hosted my portfolio website 😊



Essentials of AWS EC2

- **Scalability:** We can scale up or scale down the storage depending upon the traffic server. If there is a low traffic in our server then we can decrease its storage and vice versa.
- **Flexibility:** In terms of flexibility, AWS is highly flexible. Suppose, if we want to host our website or an application for only 1 hour then we have to pay for only one hour.
- **Security:** AWS takes care of our stored data & also eliminates suspicious activities.
- **Cost Effective:** AWS has something called 'Free Tier Account' where we can use all the features of AWS free for whole one year. It also has a feature 'Pay as you go model' i.e. we have to pay only for those services which we have taken at lease.

[Link of the website running on AWS EC2 Instance](#)

[GitHub Repo](#) 

[Presentation](#) 