AISSMS

COLLEGE OF ENGINEERING

ज्ञानम् सकलजनहिताय

Approved by AICTE, New Delhi, Recognized by Government of Maharashtra
Affiliated to Savitribai Phule Pune University and recognized 2(f) and 12(B) by UGC
(Id.No. PU/PN/Engg./093 (1992)
Accredited by NAAC with "A+" Grade | NBA - 7 UG Programmes

Department of Computer Engineering

"CC Mini-project Report"

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

Submitted By

Name of the Student: Piyusha Rajendra Supe

Roll No: 23CO315

Under the Guidance of

Prof. V. M. Kanavde

ALL INDIA SHRI SHIVAJI MEMORIAL SOCIETY'S COLLEGE OF ENGINEERING

PUNE-411001

Academic Year: 2024-25(Term-II)

Savitribai Phule Pune University

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय

Approved by AICTE, New Delhi, Recognized by Government of Maharashtra
Affiliated to Savitribai Phule Pune University and recognized 2(f) and 12(B) by UGC
(Id.No. PU/PN/Engg./093 (1992)
Accredited by NAAC with "A+" Grade | NBA - 7 UG Programmes

Department of Computer Engineering

CERTIFICATE

This is to certify that **Piyusha Rajendra Supe** from **Third Year Computer Engineering** has successfully completed her work titled "**Cloud computing Mini-project**" at AISSMS College of Engineering, Pune in the partial fulfillment of the Bachelor's Degree in Engineering.

Prof. V. M. Kanavde	**Dr. S. V. Athawale**	**Dr. D. S. Bormane**
(Faculty Guide)	(Head of Department)	(Principal)
Computer Engineering	Computer Engineering	AISSMSCOE, Pune

ACKNOWLEDGEMENT

It is with immense gratitude and respect that I take this opportunity to acknowledge the invaluable support and guidance I have received during the course of my mini project in Cloud Computing. This project has been a significant learning experience, and its successful completion would not have been possible without the constant support and encouragement of many individuals. First and foremost, I would like to express my heartfelt thanks to the entire faculty team for their unwavering support and insightful guidance throughout the project. Their valuable feedback, mentorship, and encouragement have played a crucial role in shaping the direction and quality of this work. I am sincerely grateful for the knowledge shared with me, which has significantly deepened my understanding of cloud platforms, services, and deployment models. I would also like to extend my sincere appreciation to the technical staff, colleagues, and fellow students who contributed directly or indirectly to the successful execution of this project. Their valuable inputs, discussions, and assistance were instrumental in refining my approach and overcoming challenges. Finally, I am deeply thankful to everyone who helped me gain a clearer perspective on Cloud Computing concepts, inspired innovative thinking, and motivated me to keep learning and improving. This project has not only enhanced my technical skills but also enriched my overall learning experience. I remain grateful to all who played a part in making this endeavour a success

Academic Year: 2024-2025

Piyusha Rajendra Supe (23CO315)

TABLE OF CONTENTS

ABSTRACT

This mini project explores the practical implementation of cloud computing by deploying and hosting a dynamic website on the Google Cloud Platform (GCP) using its powerful suite of APIs and services. Cloud computing has revolutionized modern application deployment by offering scalable, secure, and cost-efficient infrastructure. In this project, we aim to demonstrate the end-to-end process of developing a web application and leveraging cloud resources for hosting, storage, and performance optimization. The project begins with the design and development of a responsive and user-friendly website using standard web technologies such as HTML, CSS, JavaScript, and a backend framework (e.g., Node.js or Python Flask). Once the local development is completed, the website is prepared for cloud deployment. Google Cloud Platform is selected as the cloud service provider due to its robust ecosystem, wide range of tools, and seamless integration with modern development workflows.

Deployment is executed using Google Cloud's Compute Engine and Cloud Storage. The backend logic is hosted using virtual machine instances created via Compute Engine, while static assets such as images, stylesheets, and JavaScript files are stored in Cloud Storage for efficient content delivery. APIs provided by GCP, such as the Cloud DNS API for domain management, the Compute Engine API for instance provisioning, and the IAM API for secure access control, are utilized to automate and manage the infrastructure programmatically. In addition, the website is secured using HTTPS with SSL certificates configured through Google-managed certificates, and monitoring is enabled using Google Cloud's Operations Suite (formerly Stackdriver) for real-time performance tracking and error logging. This ensures high availability, security, and maintainability of the hosted application. This project highlights the advantages of using cloud services for modern web hosting, including auto-scaling, load balancing, cost management, and reduced operational overhead. By using GCP's APIs and tools, the deployment process becomes highly efficient and scalable, showcasing how developers can focus more on application logic rather than infrastructure management. Through this hands-on project, we gain valuable insights into the practical applications of cloud computing, service automation, and API integration. It not only strengthens our understanding of web hosting in the cloud but also prepares us for future endeavours in cloud-native application development.

INTRODUCTION

In today's digital era, cloud computing has become an essential pillar for deploying scalable, flexible, and cost-effective applications. Traditional on premise infrastructure has been rapidly replaced by cloud-based solutions due to their numerous advantages, such as high availability, reduced maintenance, and seamless scalability. Among the leading cloud service providers, Google Cloud Platform (GCP) stands out for its robust infrastructure, developer-friendly tools, and extensive set of APIs that streamline cloud-based application deployment and management. This mini project is focused on understanding and applying the concepts of cloud computing by hosting a fully functional website on GCP. The project not only involves the development of a web application but also covers the complete deployment pipeline using GCP's cloud services and APIs. The goal is to gain hands-on experience with the infrastructure-as-a-service (IaaS) and platform-as-a-service (PaaS) offerings of GCP, and to understand how cloud environments can be effectively used to host modern web applications.

The website developed in this project is built using standard web technologies such as HTML, CSS, and JavaScript for the front end, along with a backend written in a server-side language such as Python (Flask) or Node.js. After development and testing on a local machine, the application is migrated to the cloud environment using Google Compute Engine to host the backend logic and Google Cloud Storage to serve static assets efficiently. Google Cloud's APIs play a central role in this project. These include the Compute Engine API for creating and managing virtual machine instances, the Cloud Storage API for handling static content, the Cloud DNS API for domain configuration, and the Identity and Access Management (IAM) API for managing permissions and security. The use of these APIs demonstrates how cloud services can be automated and customized to suit specific project requirements.

Additionally, GCP's operations suite is used to monitor application performance and log system events, ensuring high availability and quick troubleshooting. SSL certificates and HTTPS are configured to provide secure access to the website.

Through this project, we aim to bridge theoretical knowledge of cloud computing with practical implementation. It provides a comprehensive learning experience in cloud deployment, service integration, API usage, and performance management. Overall, this project prepares students and aspiring developers to work with real-world cloud infrastructure and adopt best practices in cloud-native web development.

PROBLEM STATEMENT

Title: Hosting a Website on Google Cloud Platform Using GCP APIs

Description:
In the current landscape of web development, hosting a website using traditional methods poses several challenges such as limited scalability, high maintenance costs, and manual server configuration. As businesses and developers increasingly seek more flexible and automated deployment options, cloud platforms like Google Cloud Platform (GCP) provide an efficient solution. This project addresses the challenge of deploying and managing a website in a cloud environment using GCP services and APIs.

Problem Statement:

Hosting and managing web applications on traditional servers can be inefficient and difficult to scale. Manual deployment, hardware dependency, and lack of real-time monitoring tools make the process time-consuming and error-prone. This project aims to solve these issues by using Google Cloud Platform to host a website with improved scalability, automation, and security.

Key Challenges Addressed:

- Difficulty in manually configuring and deploying websites on physical servers.
- Limited scalability with traditional hosting providers.
- High maintenance overhead and downtime risks.
- Lack of centralized monitoring and logging tools.
- Security vulnerabilities due to poor configuration management.
- Inefficient storage and content delivery for static assets.

By leveraging GCP's APIs and services, this project demonstrates a modern, automated, and scalable approach to web hosting.

OVERVIEW

This project focuses on deploying a functional website using Google Cloud Platform (GCP), highlighting how cloud services and APIs simplify modern web hosting. The aim is to provide a hands-on understanding of cloud infrastructure, automated deployment, and service integration. By utilizing GCP tools such as Compute Engine, Cloud Storage, and IAM APIs, we ensure that the website is scalable, secure, and efficiently managed. This project serves as a practical demonstration of implementing cloud computing concepts using real-world tools.

Overview & Basic Steps:

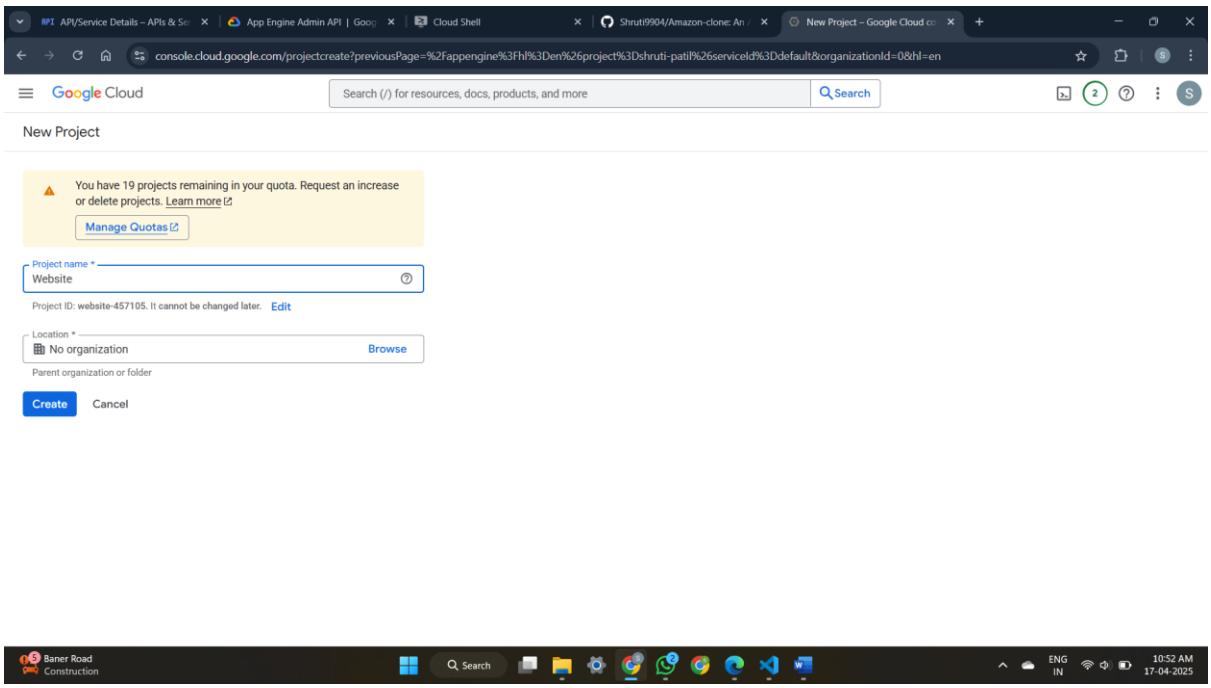
- **Step 1: Website Development**
  Design and develop a responsive website using front-end (HTML, CSS, JavaScript) and back-end technologies (Node.js, Python Flask, etc.).
- **Step 2: Setting Up a GCP Account**
  Create a Google Cloud account and set up a new project within the GCP Console.
- **Step 3: Configuring Cloud Storage and Compute Engine**
  Use Google Cloud Storage to host static files and Compute Engine to run the backend server or application logic.
- **Step 4: Domain and DNS Configuration**
  Set up a custom domain using Cloud DNS and configure it to point to the hosted application.
- **Step 5: API Integration**
  Use GCP APIs (Compute Engine API, Cloud Storage API, IAM API) to automate deployment and manage cloud resources.
- **Step 6: Security and Monitoring**
  Enable HTTPS using SSL certificates and monitor performance using Google Cloud Operations Suite.

This project offers a complete learning cycle—from development to deployment—within the cloud ecosystem

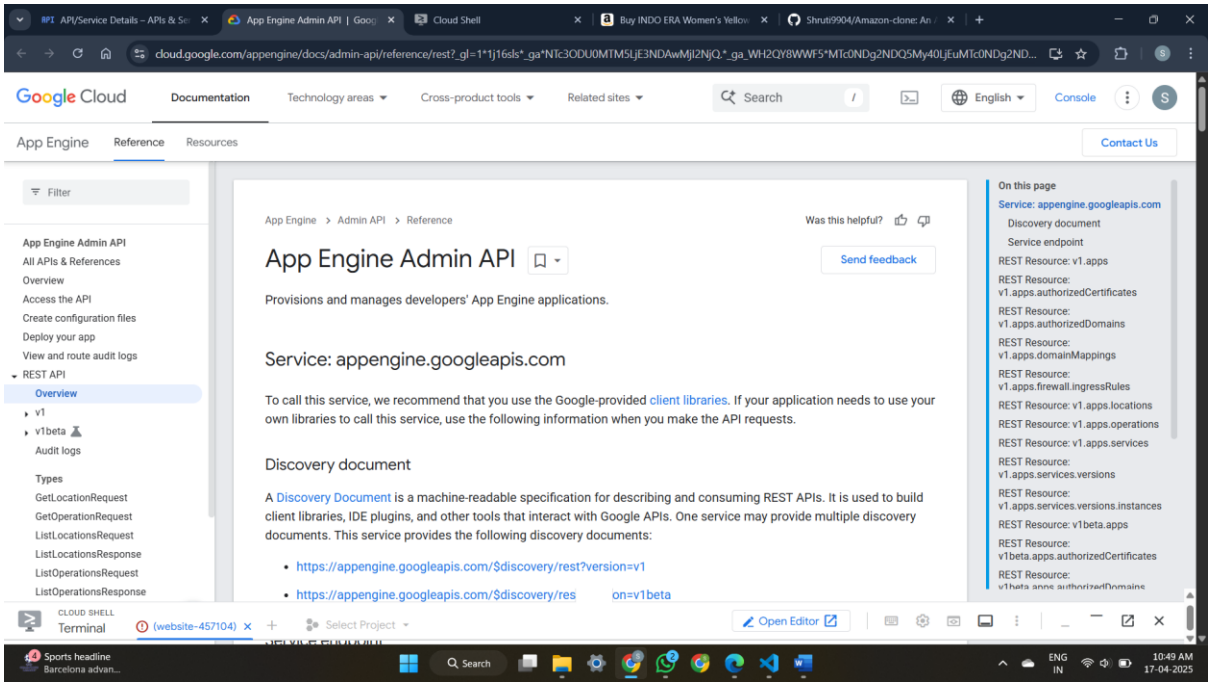
IMPLEMENTATION

The following are the steps followed for implementation of the project

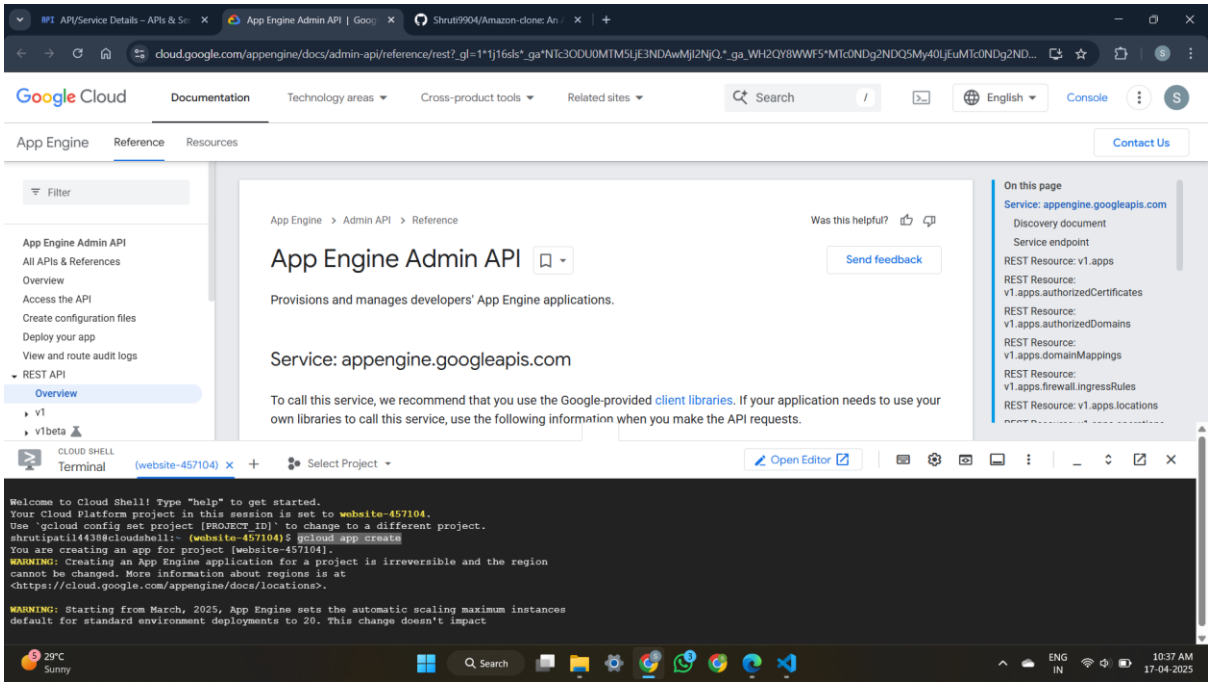
1] Creating a project



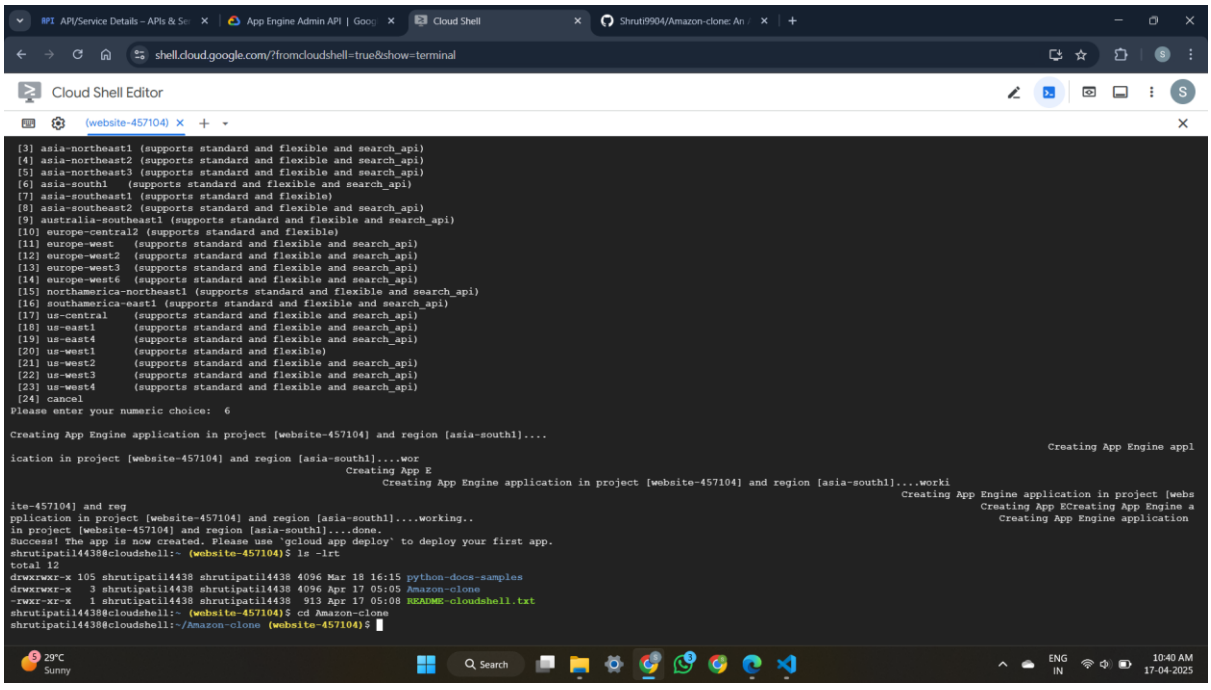
2] Search for app engine admin API



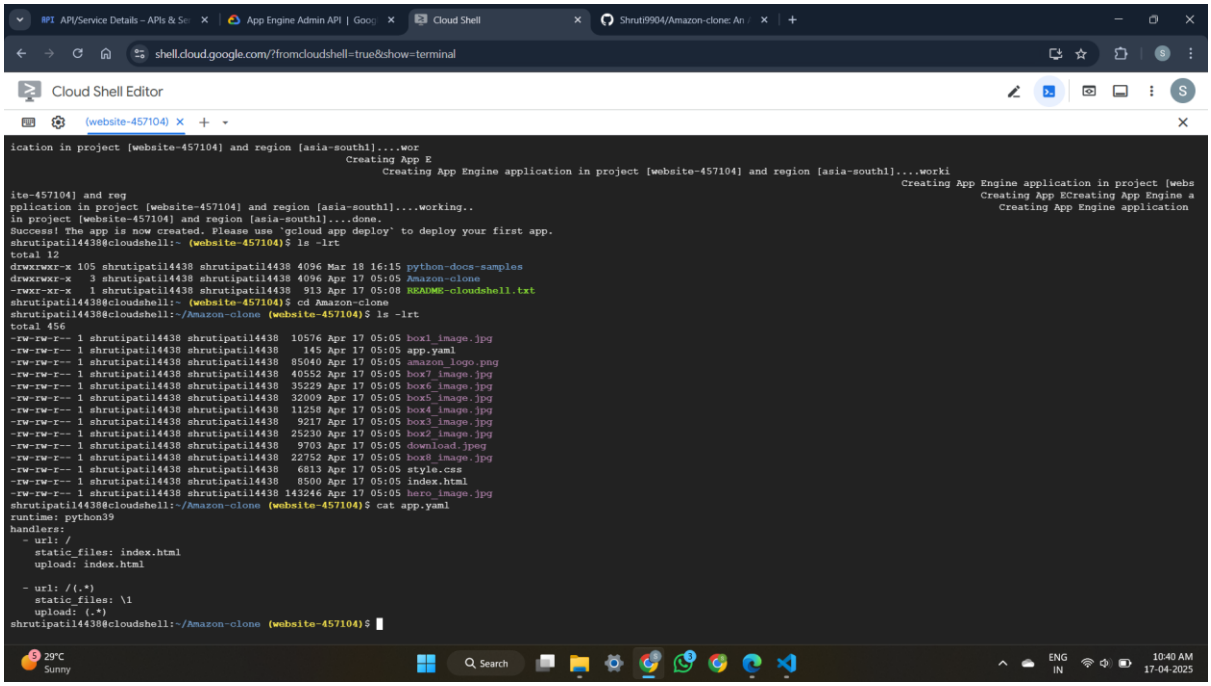
3] Enable the API



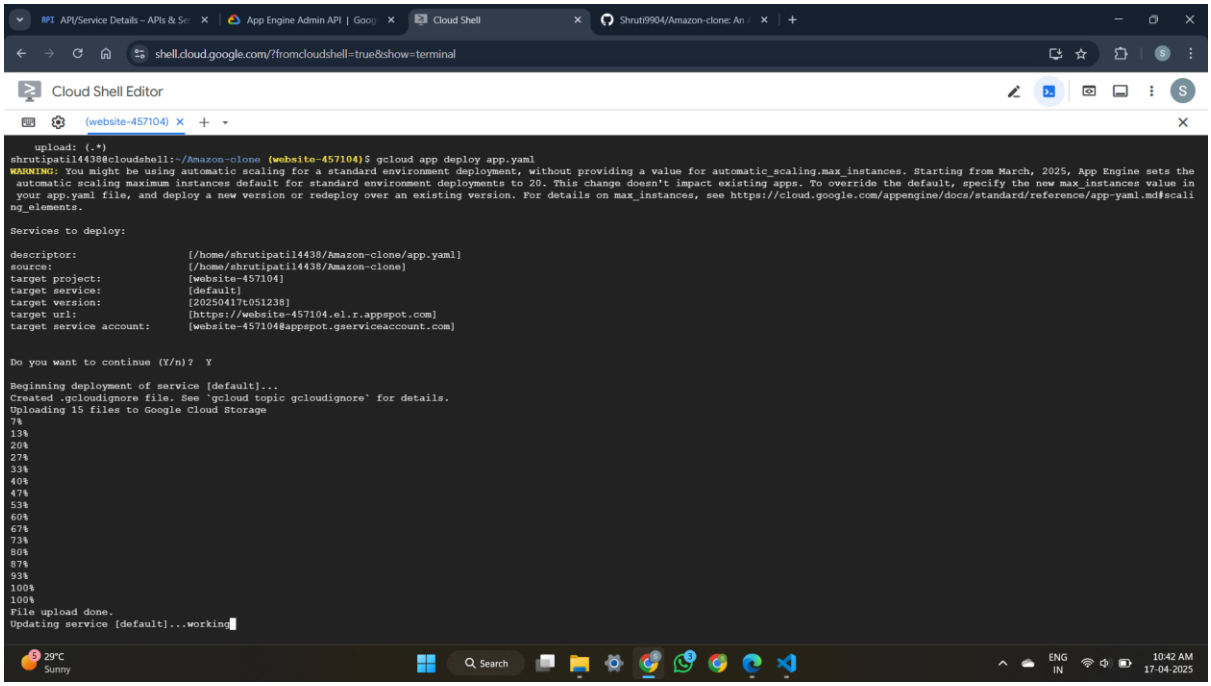
4] Enter to terminal and upload the documents



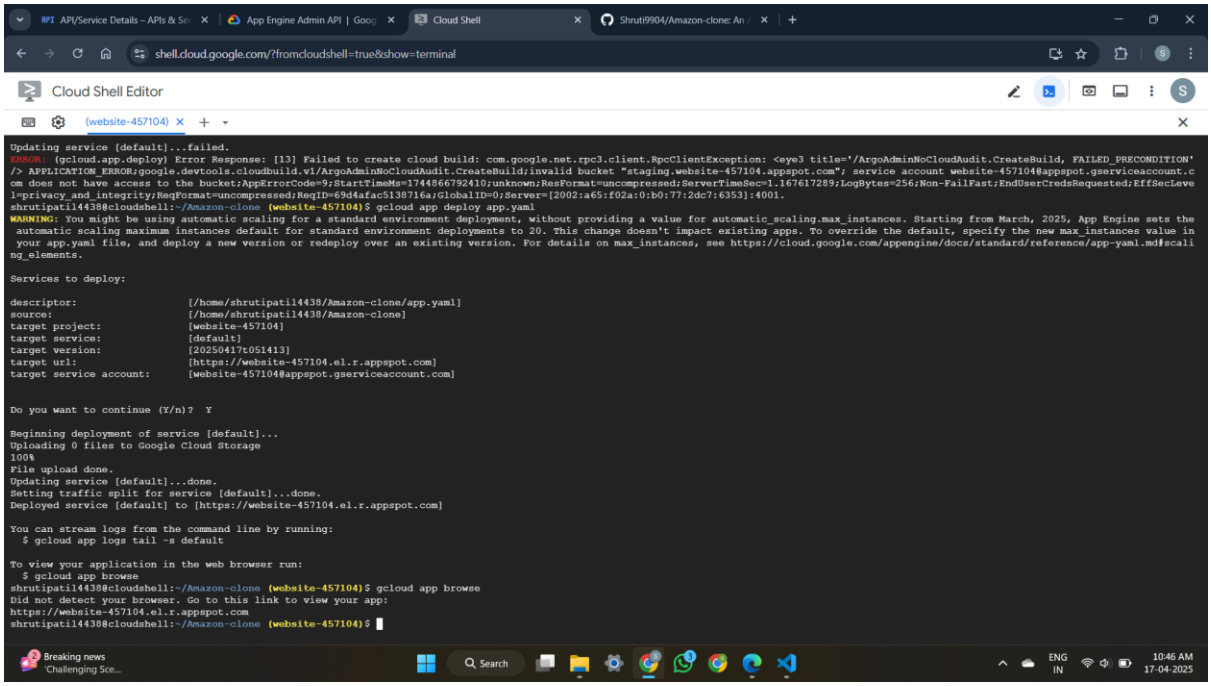
5] Extract and view whether the documents are correct which are to be deployed



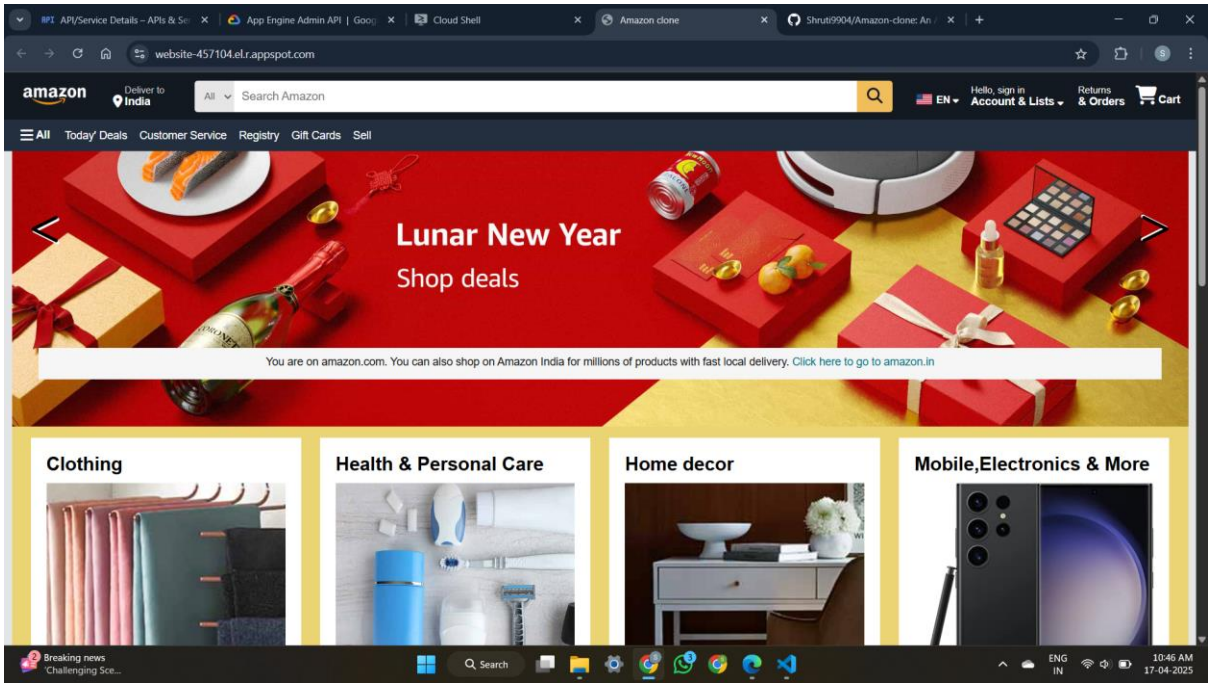
6] Deploy the app using the command

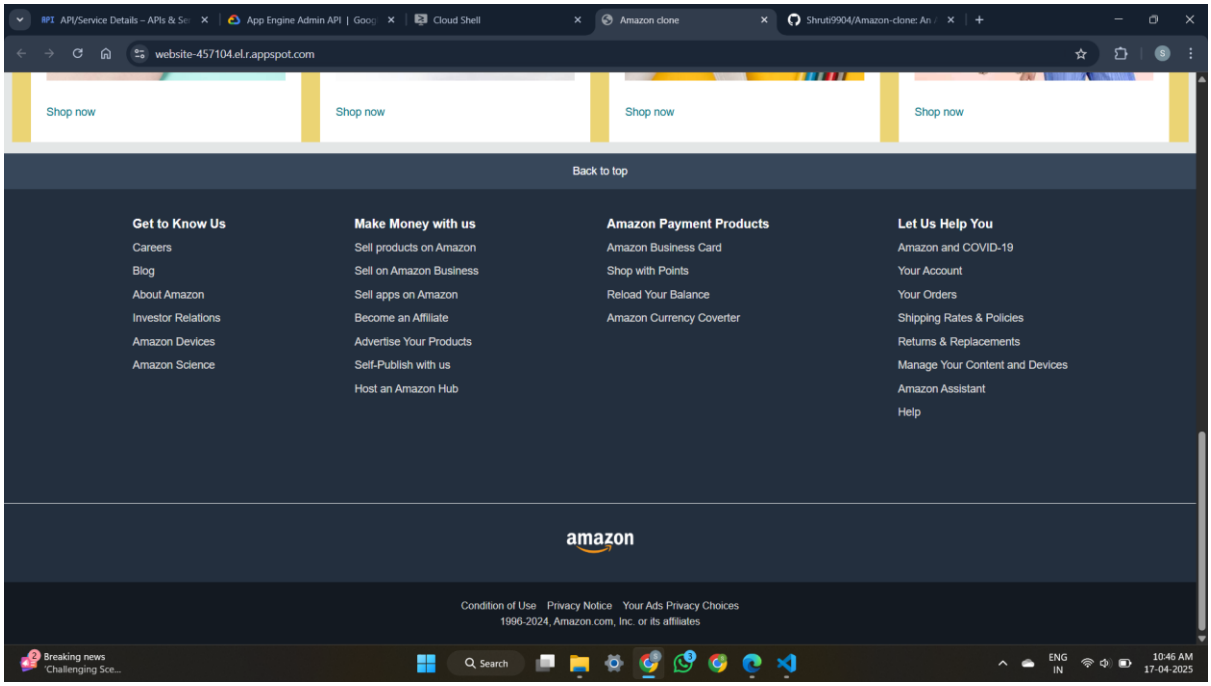


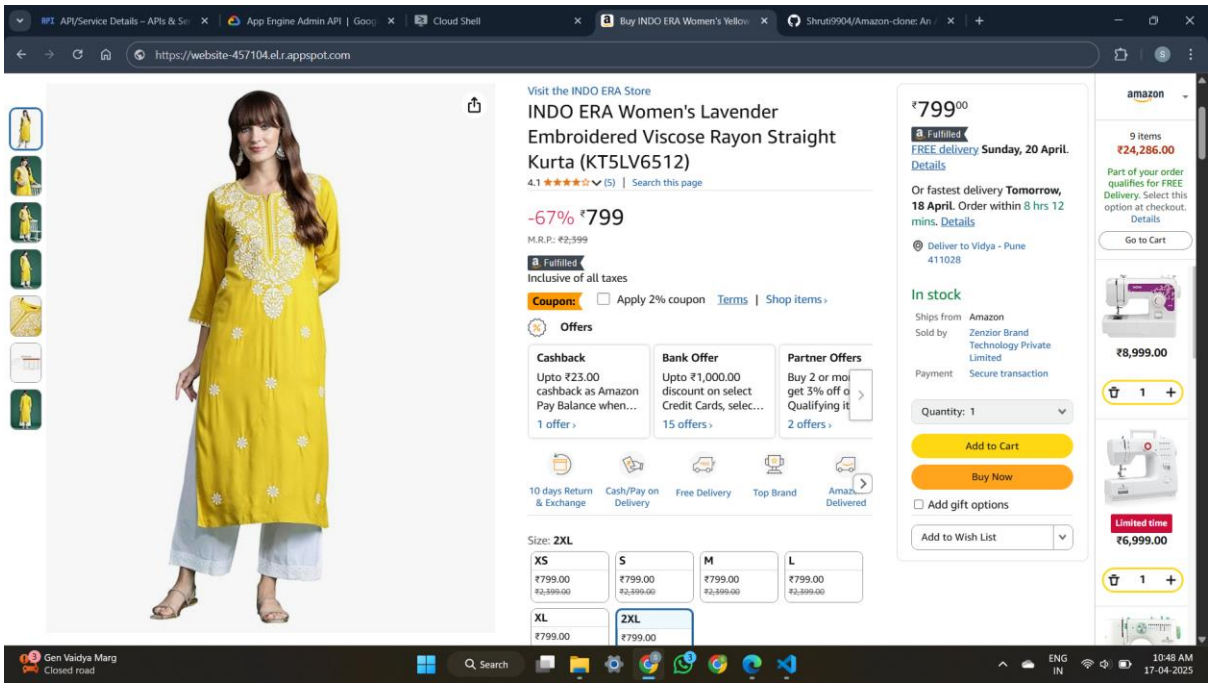
7] Check whether he deployment is correct by browsing the app using the deployed URL



8] Here is how the app looks after deployment




website-457104.el.r.appspot.com
Shop now
Shop now
Shop now
Shop now
Back to top
Get to Know Us
Careers
Blog
About Amazon
Investor Relations
Amazon Devices
Amazon Science
Make Money with us
Sell products on Amazon
Sell on Amazon Business
Sell apps on Amazon
Become an Affiliate
Advertise Your Products
Self-Publish with us
Host an Amazon Hub
Amazon Payment Products
Amazon Business Card
Shop with Points
Reload Your Balance
Amazon Currency Coverter
Let Us Help You
Amazon and COVID-19
Your Account
Your Orders
Shipping Rates & Policies
Returns & Replacements
Manage Your Content and Devices
Amazon Assistant
Help
amazon
Condition of Use Privacy Notice Your Ads Privacy Choices
1996-2024, Amazon.com, Inc. or its affiliates


https://website-457104.el.r.appspot.com
Visit the INDO ERA Store
INDO ERA Women's Lavender Embroidered Viscose Rayon Straight Kurta (KT5LV6512)
-67% ₹799
M.R.P.: ₹2,399
Inclusive of all taxes
Coupon: Apply 2% coupon Terms | Shop items
Offers
Cashback
Upto ₹23.00 cashback as Amazon Pay Balance when...
Bank Offer
Upto ₹1,000.00 discount on select Credit Cards, selec...
Partner Offers
1 offer
15 offers
2 offers
10 days Return & Exchange
Cash/Pay on Delivery
Free Delivery
Top Brand
Size: 2XL
XS ₹799.00
S ₹799.00
M ₹799.00
L ₹799.00
XL ₹799.00
2XL ₹799.00
₹799.00
Fulfilled
FREE delivery Sunday, 20 April. Details
Or fastest delivery Tomorrow, 18 April. Order within 8 hrs 12 mins. Details
Deliver to Vidya - Pune 411028
In stock
Ships from Amazon
Sold by Zenzior Brand Technology Private Limited
Payment Secure transaction
Quantity: 1
Add to Cart
Buy Now
Add gift options
Add to Wish List
9 items
₹24,286.00
Part of your order qualifies for FREE Delivery. Select this option at checkout. Details
Go to Cart
₹8,999.00
Limited time
₹6,999.00

FUNCTIONALITY AND ADVANTAGE

Functionality

- Hosts a fully functional website using Google Cloud Platform (GCP).
- Front-end developed using HTML, CSS, and JavaScript for a responsive user interface.
- Back-end implemented with Node.js or Python Flask to handle server-side logic.
- Static content (images, stylesheets, scripts) stored and served via Google Cloud Storage.
- Website deployed using Google Compute Engine virtual machines.
- Custom domain setup configured using Google Cloud DNS.
- Secure access enabled through SSL certificates and HTTPS.
- Uses GCP APIs (Compute Engine API, Cloud Storage API, IAM API) for automation and infrastructure management.
- Real-time monitoring and logging implemented using Google Cloud Operations Suite.
- Ensures smooth deployment, fast load times, and reliable user experience.

Advantages

- **Scalability:** Automatically adjusts resources based on traffic and demand.
- **High Availability:** Minimizes downtime with GCP's reliable infrastructure.
- **Security:** Includes built-in IAM, firewall rules, and SSL/TLS encryption.
- **Automation:** Simplifies deployment and management through GCP APIs.
- **Cost Efficiency:** Pay-as-you-go model reduces unnecessary spending.
- **Performance Monitoring:** Easy tracking and debugging with built-in monitoring tools.
- **Centralized Management:** All services can be managed from the GCP console.
- **Seamless Integration:** Easily integrates with other GCP services and third-party tools.
- **Flexibility:** Supports both static and dynamic content with scalable backend options.
- **Future-Ready:** Ideal foundation for expanding to cloud-native or server less architectures.

CONCLUSION

In conclusion, this mini project successfully demonstrates the practical implementation of cloud computing by hosting a dynamic and responsive website on Google Cloud Platform using its APIs and services. Through this project, we have explored the complete lifecycle of website deployment—from local development to live hosting—while leveraging various cloud tools such as Compute Engine, Cloud Storage, Cloud DNS, and IAM. The integration of GCP APIs allowed for automation, efficient management, and enhanced control over the infrastructure, reducing manual effort and improving scalability. We also implemented essential features like domain mapping, secure HTTPS access, and performance monitoring to ensure a secure and reliable user experience. This approach highlights the significant benefits of cloud hosting, including cost-efficiency, high availability, and seamless scalability, all of which are crucial in modern web development. Moreover, this project has deepened our understanding of cloud infrastructure, API usage, and DevOps practices, preparing us for real-world applications and future innovations in cloud-native development. Overall, hosting a website on GCP not only meets technical requirements effectively but also aligns with industry standards, making it a smart, future-proof solution for deploying and managing web applications in the cloud.

REFERENCES

- https://cloud.google.com/free
- https://cloud.google.com/endpoints/docs/openapi/deploy-api-backend
- https://cloud.google.com/endpoints/docs/deploy-api
- https://medium.com/google-cloud/deploying-api-via-google-app-engine-1f0209ba5a5e
- https://medium.com/google-cloud/hosting-a-static-website-on-google-cloud-using-google-cloud-storage-ddebcdcc8d5b