UNIVERSITY OF MUMBAI

**DEPARTMENT OF COMPUTER SCIENCE**

****

M.Sc. Computer Science – Semester IV Trends in Cloud Computing

JOURNAL

2024-2025

Seat No.

UNIVERSITY OF MUMBAI

**DEPARTMENT OF COMPUTER SCIENCE**

CERTIFICATE

## This is to certify that the work entered in this journal was done in the University Department of Computer Science laboratory by Mr./Ms. Seat No. for the course of M.Sc. (Computer Science) - Semester IV (NEP 2020) during the academic year 2024- 2025 in a satisfactory manner.

#### Subject In-charge Head of Department

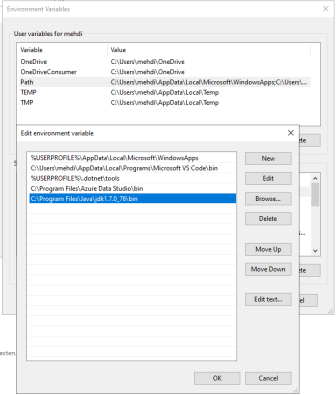
**External Examiner**

INDEX

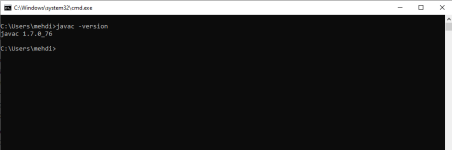
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR NO | TITLE | PG NO | DATE | SIGN |
| 1 | Using the software like /API/Tools JDK 1.7/1.8,Eclipse IDE, Dropbox API, Apache tomcat server 7.0/8.0, Google App Engine API, Servlets, Struts, Spring Framework design and develop Web applications using MVC Framework. |  |  |  |
| 2 | Installing and Configuring the required platform for google app engine |  |  |  |
| 3 | Studying the feature of GAE PaaS model. |  |  |  |
| 4 | Creating and running Web applications (Guest book, MVC) on localhost and deploying the same in google app engine. |  |  |  |
| 5 | Developing an ASP.NET based web application on the Azure platform. |  |  |  |
| 6 | Creating an Application in DropBox to store data securely. Develop a source code using DropBox API for updating and retrieving files. |  |  |  |
| 7 | Installing Cloud Foundry in localhost and exploring CF commands |  |  |  |
| 8 | Installing and Configuring Dockers in localhost and running multiple images on a Docker Platform |  |  |  |

# Practical-1

**Aim:** Using the software like / API / Tools JDK 1.7/1.8, Eclipse IDE, Dropbox API,  Apache tomcat server 7.0/8.0, Google AppEngine API, Servlets, Struts, Spring  framework design and develop Web applications using MVC Framework

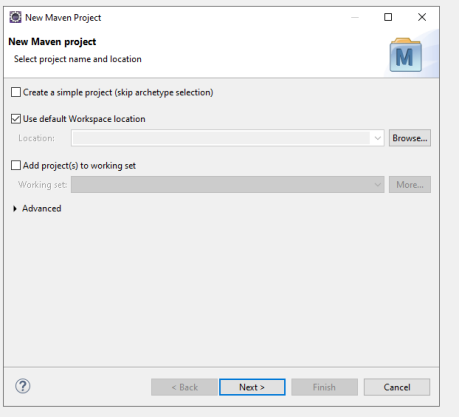
****

#### Check the version of the java installed

****.

#### Create new Maven Project

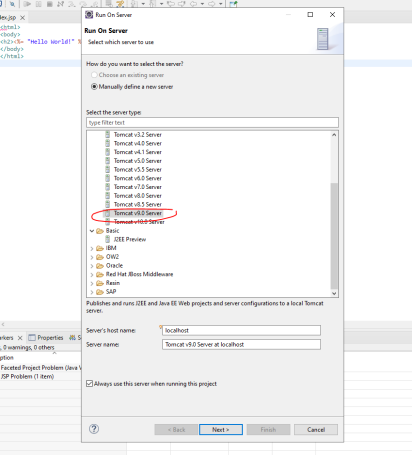
#### 

****

# Add the project name as Artifact Id

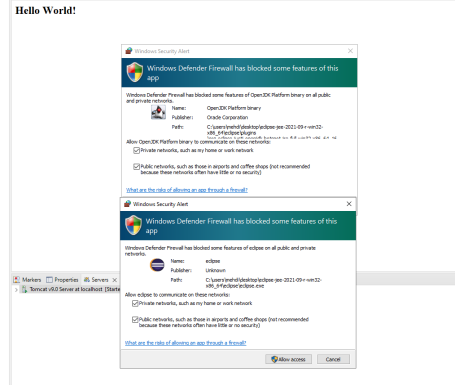
# Test run your project with default files and select the Tomcat server

# 

****

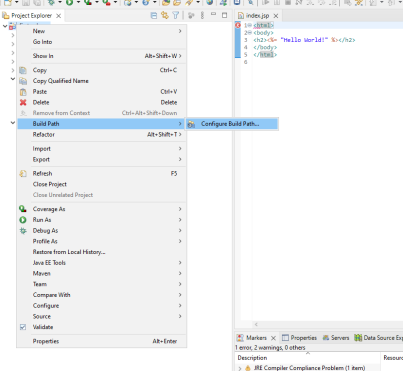
# Download and Install the Tomcat server

# 

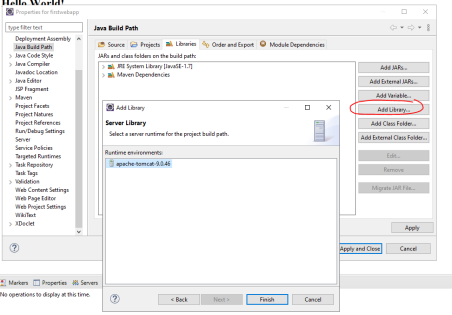
****

# Fix the error issue by correcting the Java version

# 

****

# 

****

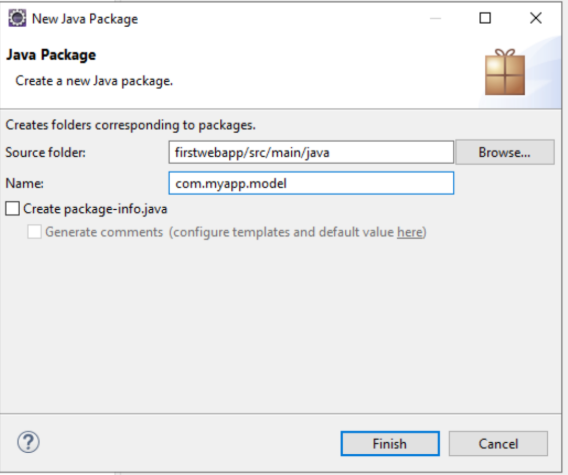
# 

# Now lets create the MVC Structure, First start with by creating packages for

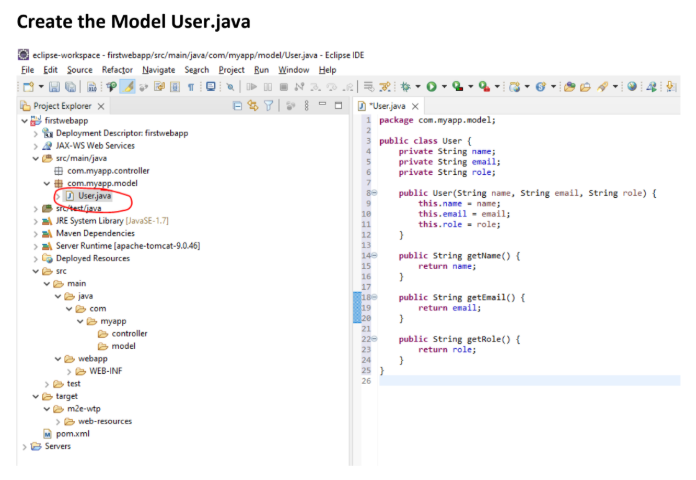
# Model and Controller

# 

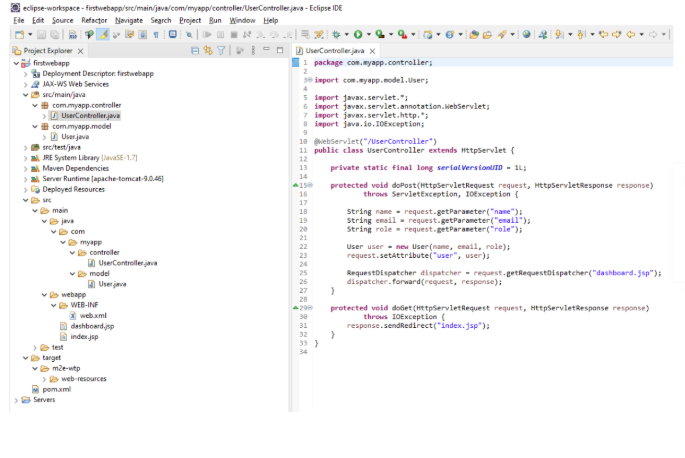
# 

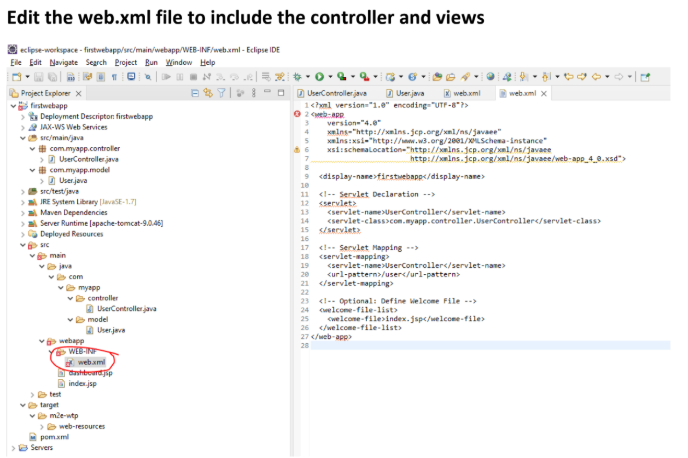


# 

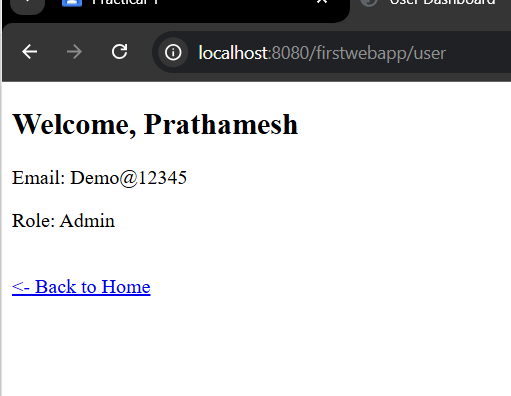
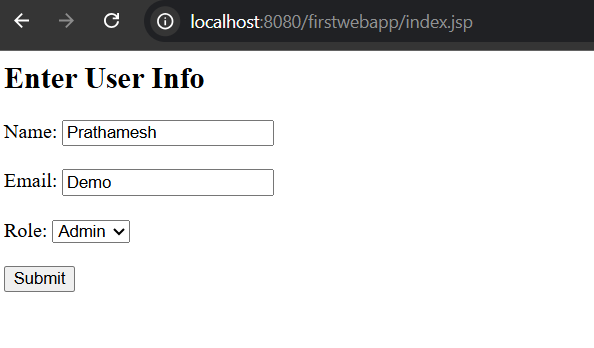


# 





# 

****

# Practical-2

# Aim: Installing and configuring the required platform for Google App Engine.

# Implementation:

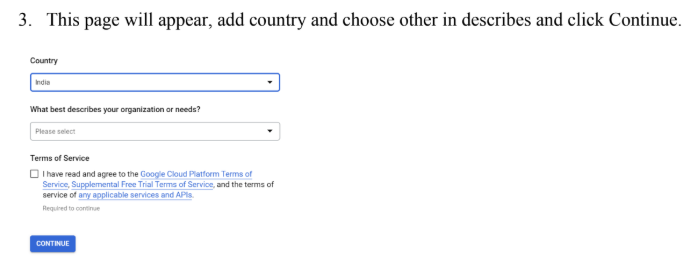
# A. Making Google App Engine account

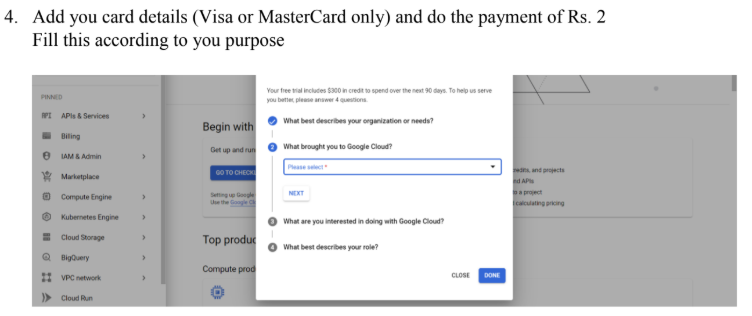
# 1. Open your google account and go to the following link

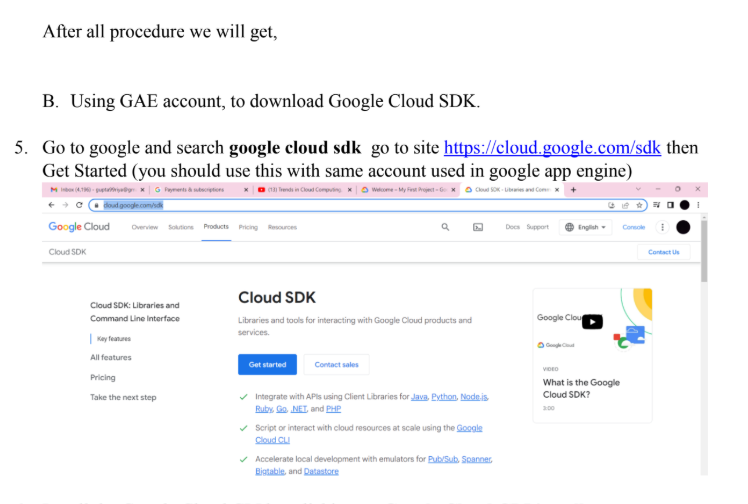
# https://cloud.google.com/appengine

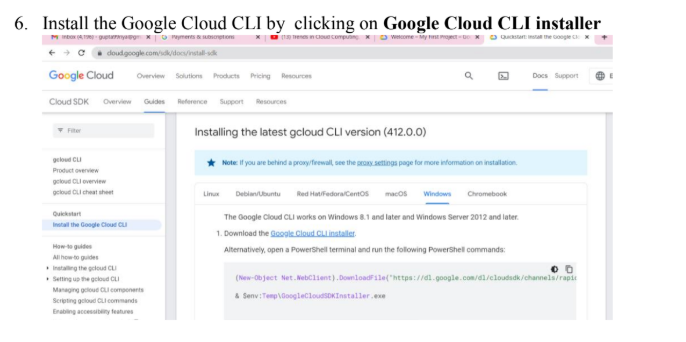
# 2. Click on Try App engine free button.

# 

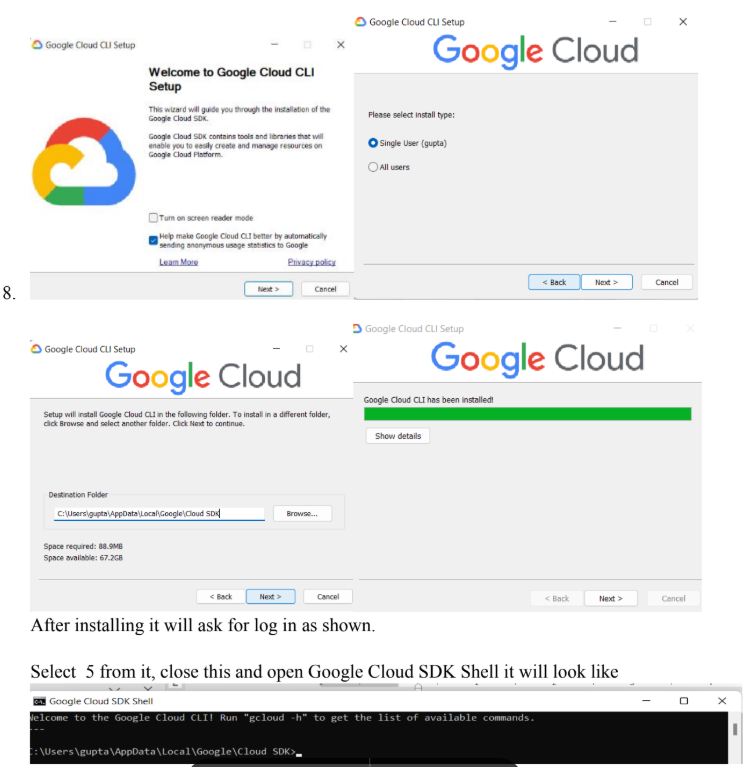






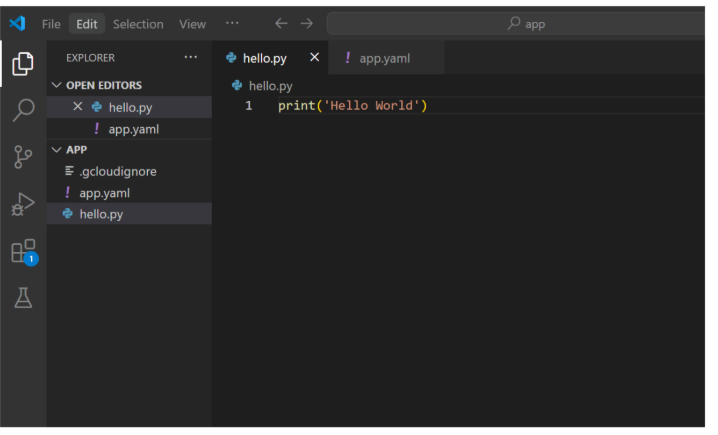


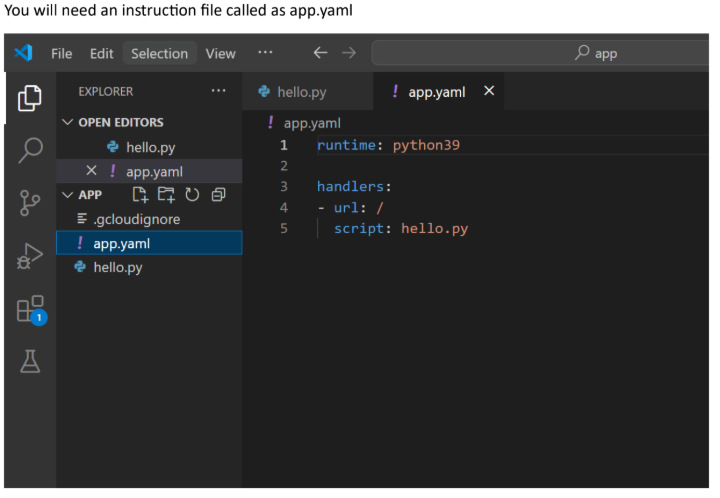
7. Launch the installer and follow the prompts. The installer is signed by Google LLC.



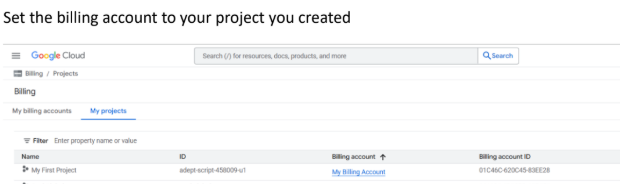
**How to Create a simple python app and Deploy on Google App Engine.**

Create a simple python program to deploy and test google app engine.

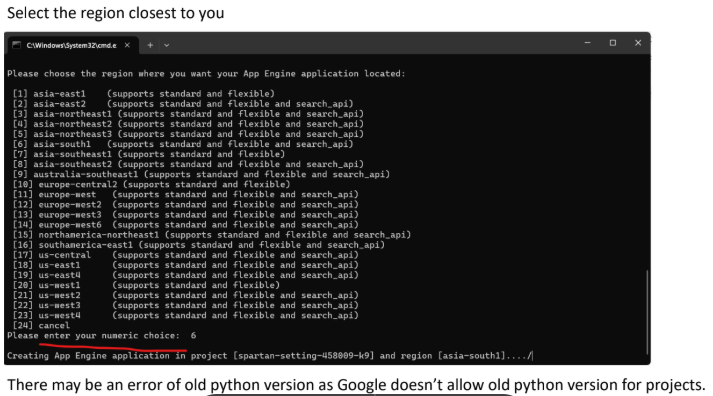




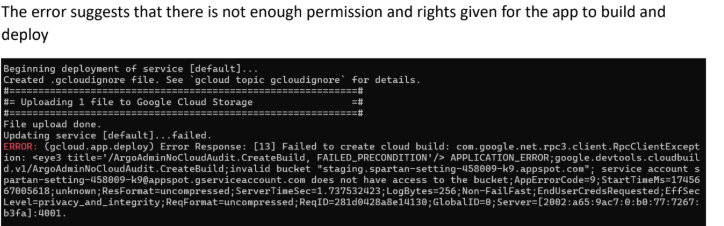
# 

****

# 



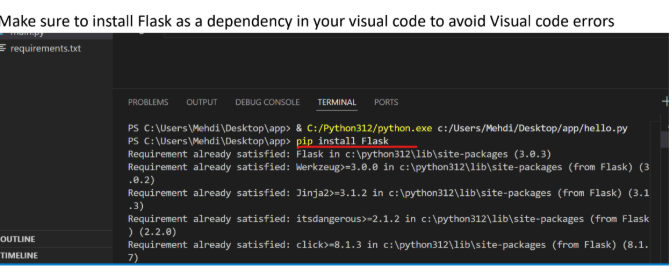
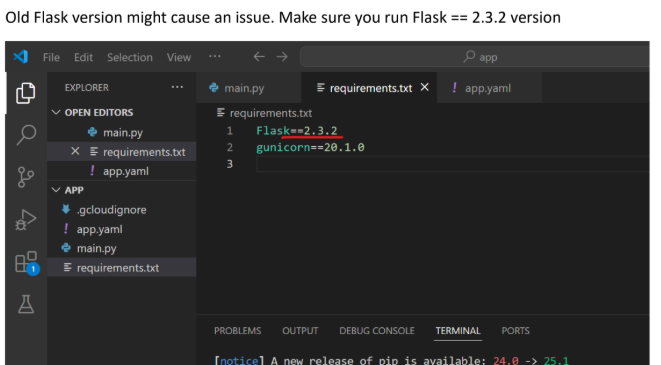
# 



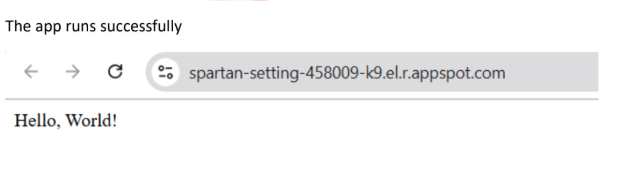
# 

****

# 



# 



# Practical-3

# Aim: Studying the features of the GAE PaaS model.

# Google App Engine (GAE) is a Platform-as-a-Service (PaaS) that simplifies web application development and deployment by handling infrastructure management and scaling. Key features include automatic scaling, support for multiple programming languages, a rich set of APIs, and integration with other Google Cloud services.

# Automatic Scaling and Load Balancing: GAE automatically adjusts resources (CPU, memory, etc.) based on traffic demands, ensuring applications remain responsive and available without manual intervention.

# Language and Framework Support: Developers can utilize various popular languages like Java, Python, PHP, Ruby, and Go, as well as popular frameworks like Spring and Django.

# Managed Infrastructure: GAE handles the underlying infrastructure, including servers, operating systems, and software updates, freeing developers from infrastructure management tasks.

# Integration with Other GCP Services: GAE seamlessly integrates with other Google Cloud Platform services like Cloud Datastore, Cloud SQL, and Cloud Storage, providing a comprehensive ecosystem for application development.

# Built-in Services: GAE offers built-in services like Datastore (a NoSQL database), Memcache (for caching), task queues, and scheduled tasks, simplifying common development tasks.

# Flexible Environment: GAE allows for custom runtimes using Docker containers, enabling developers to bring their own libraries, frameworks, and even custom software.

# Application Versioning: GAE supports hosting different versions of an application, allowing for easy development, testing, and staging of new features.

# 8. Cost-Effective Pricing: GAE employs a pay-per-use pricing model, where you only pay for the resources your application consumes, with a free tier available.

# 9. Developer Tools: GAE provides a suite of developer tools for building, testing, debugging, deploying, and monitoring applications.

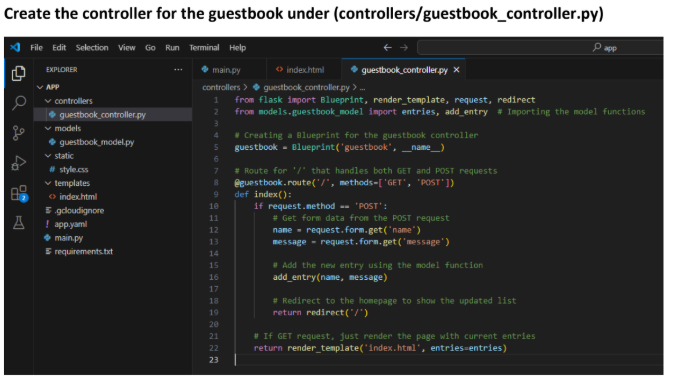
# 10. Security: GAE handles security updates and infrastructure maintenance, ensuring a secure environment for applications.

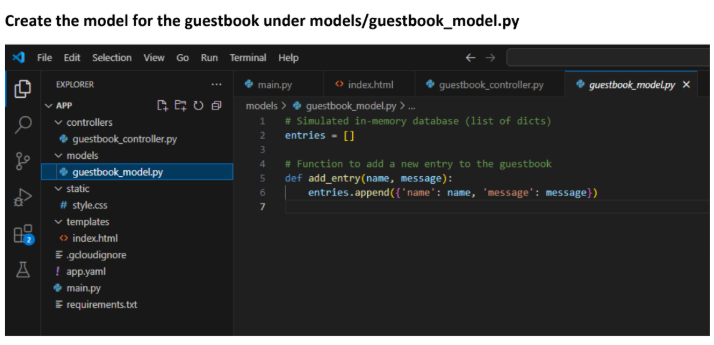
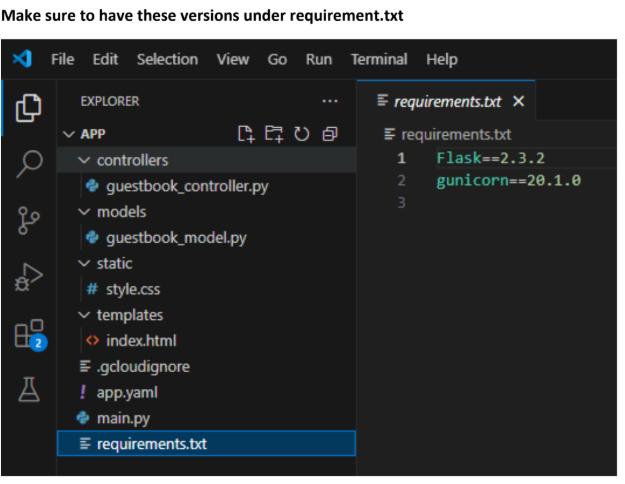
# Practical-4

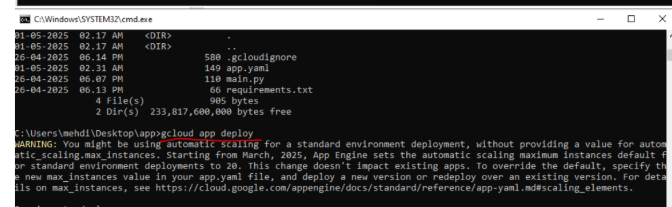
# Aim: Creating and running Web applications (Guest book, MVC) on localhost and deploying the same in Google App Engine.

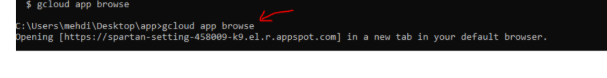
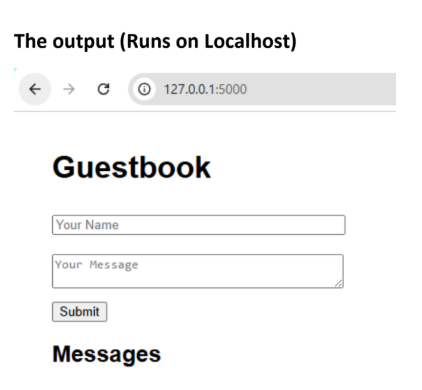
# 

# 







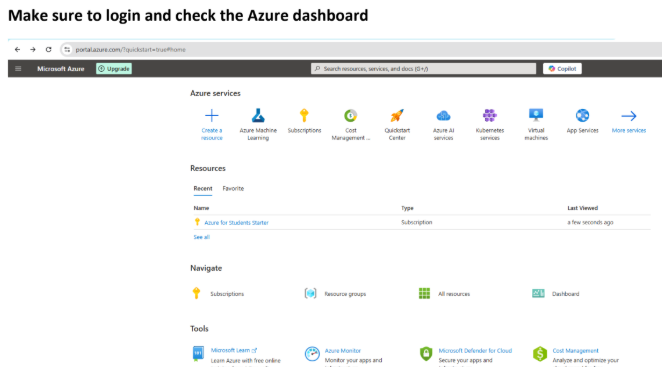


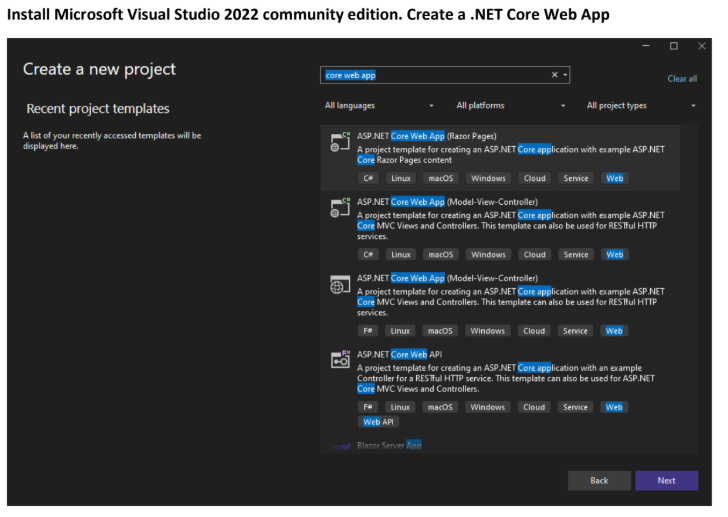
# Practical-5

# Aim: Developing an ASP.NET based web application on the Azure platform.

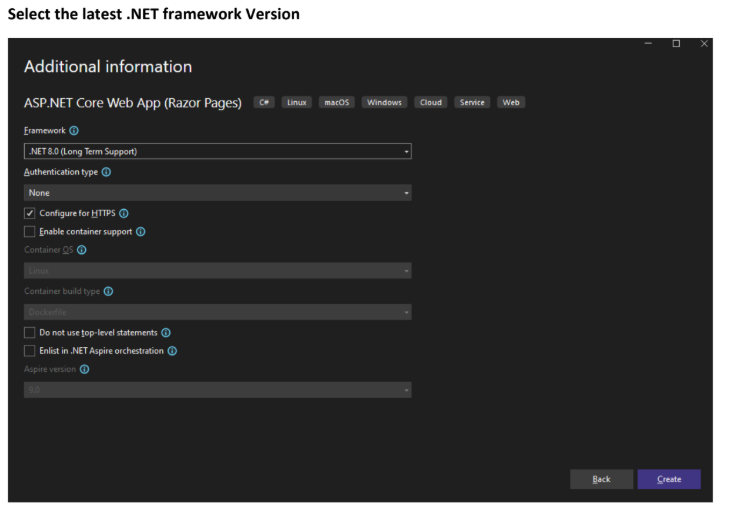
#### Create an account on Microsoft Azure portal and Signup with a debit or credit card

#### 

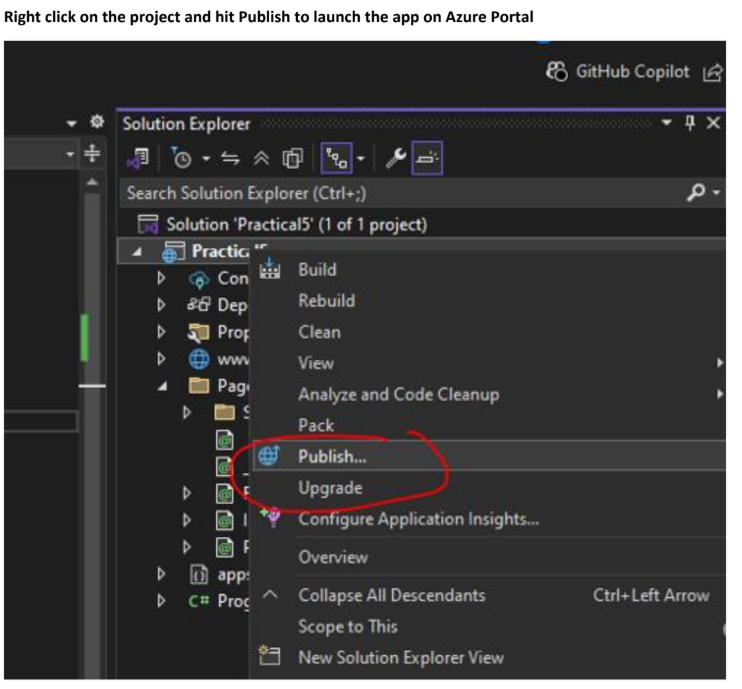




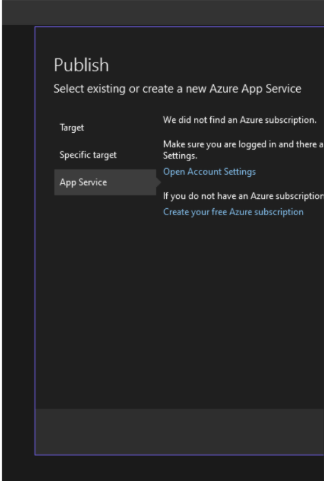
# 



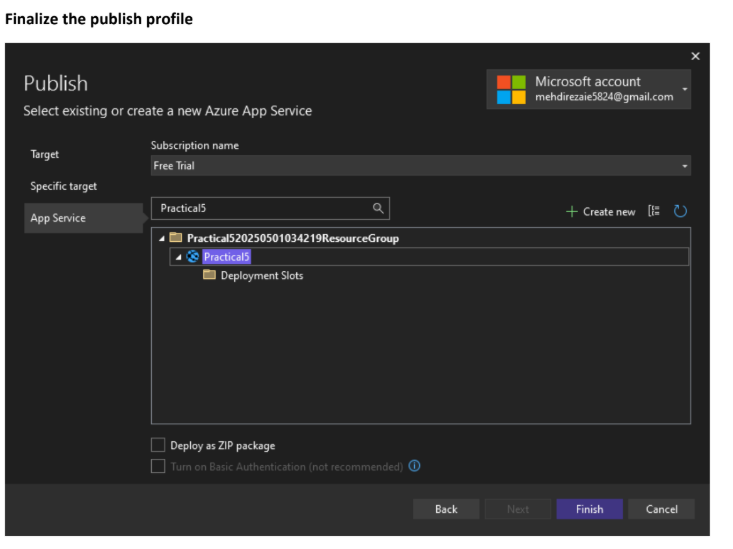
# 



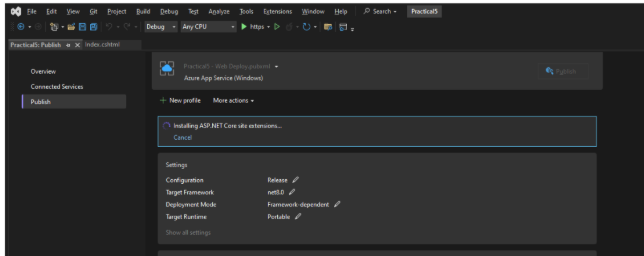
# 



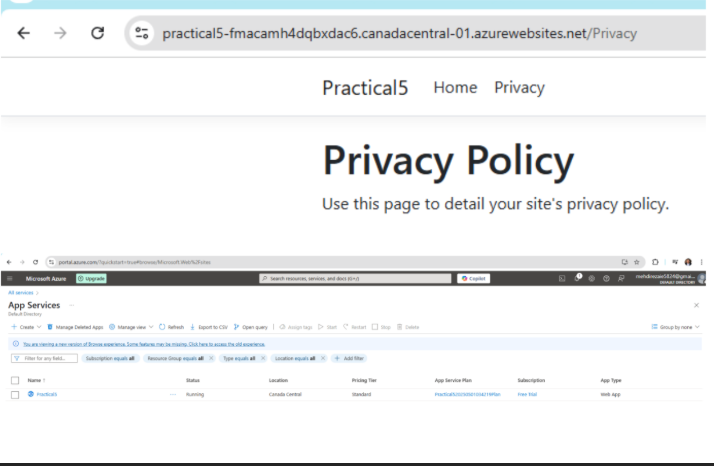
# 



# 

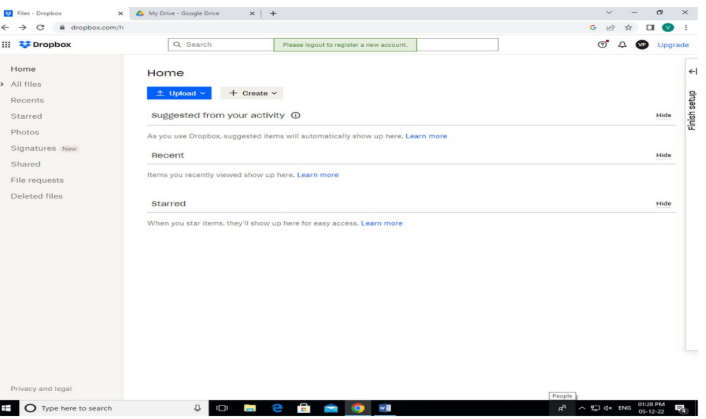
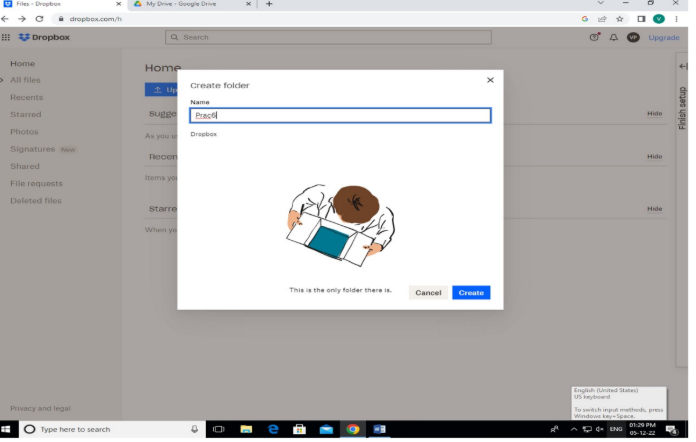
****

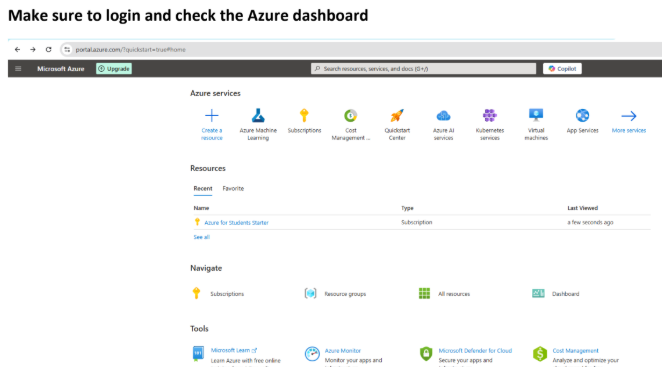
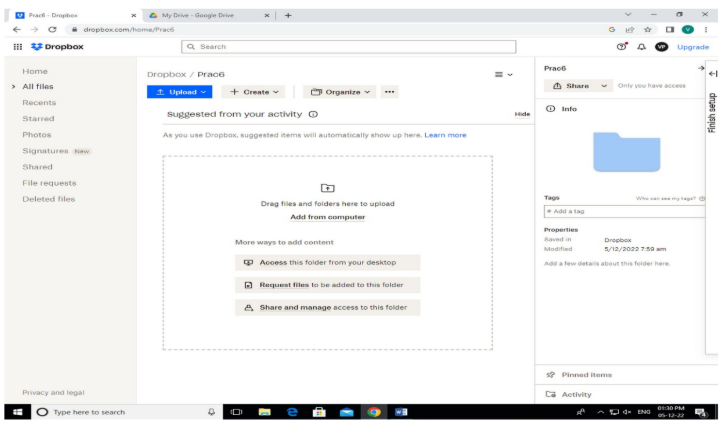
# 

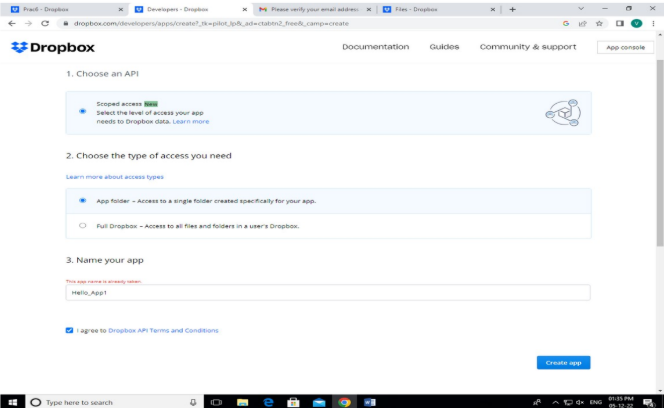
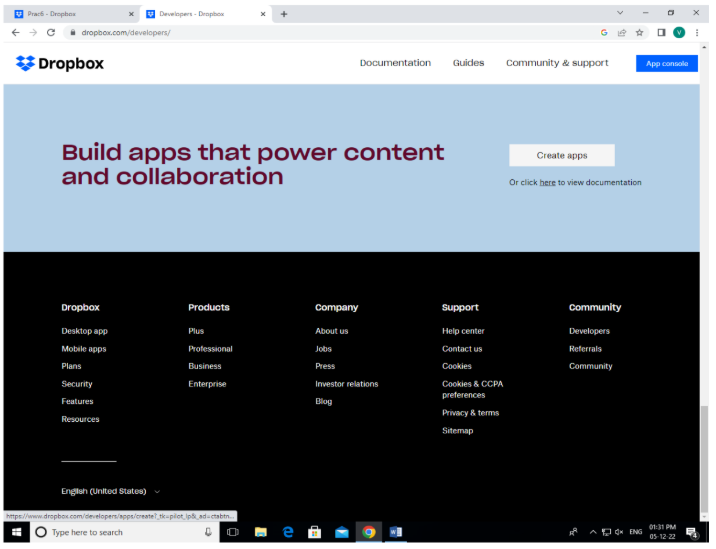


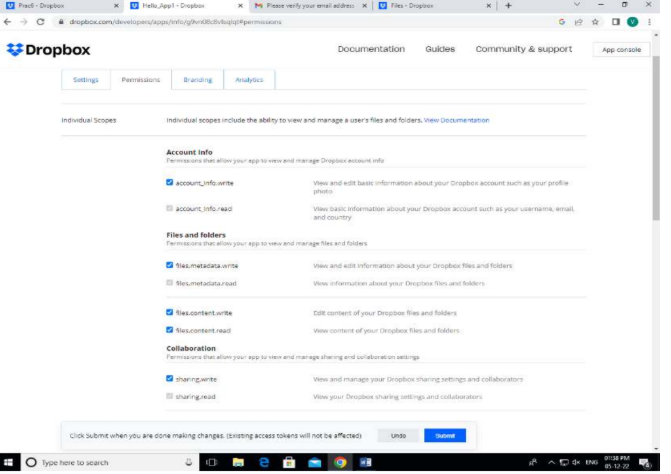
# Practical-6

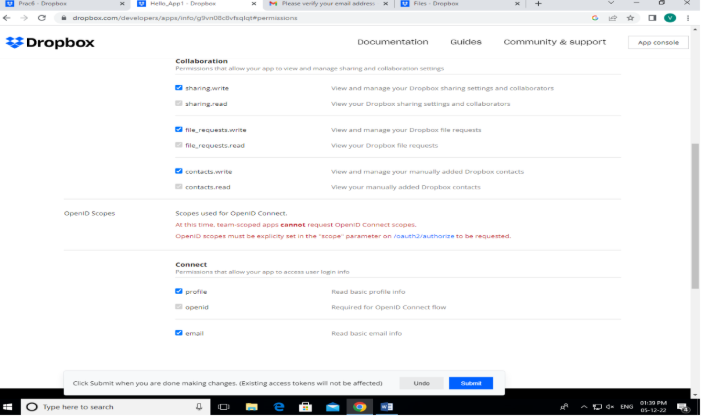
### Aim: Creating an application in Dropbox to store data securely. Develop a source code using Dropbox API for updating and retrieving files.



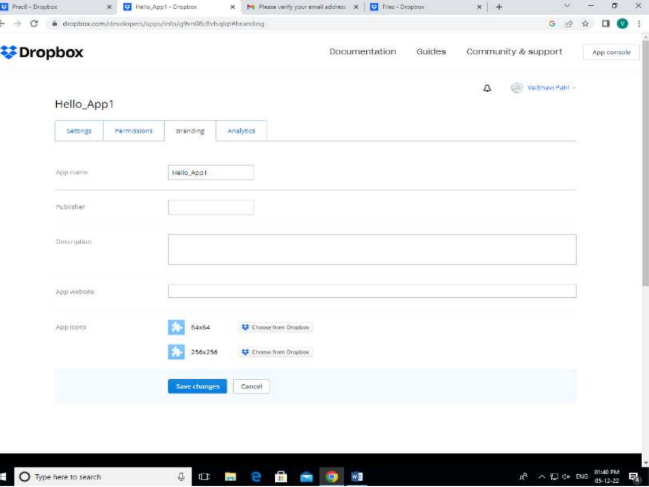




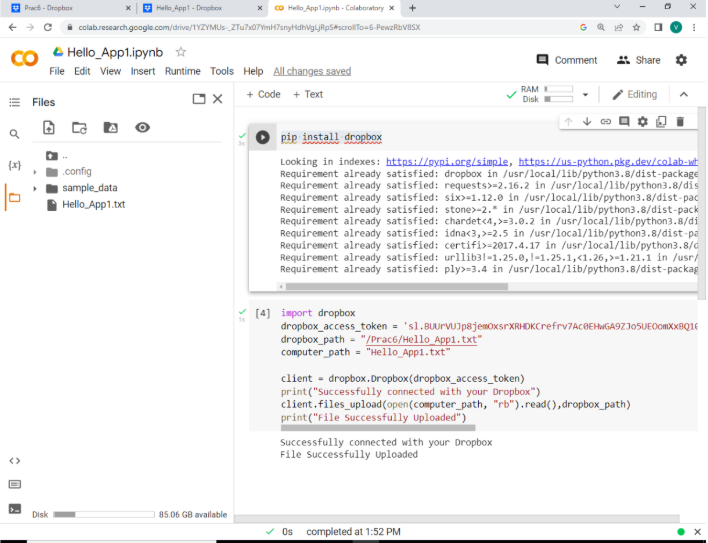




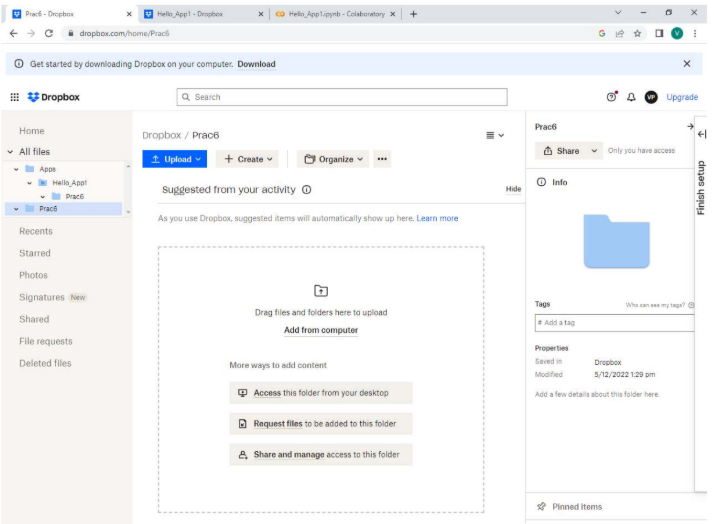
# 

****

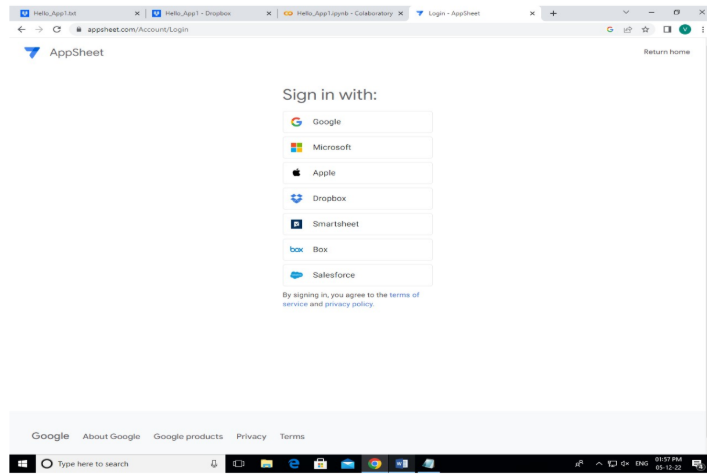
# 

****

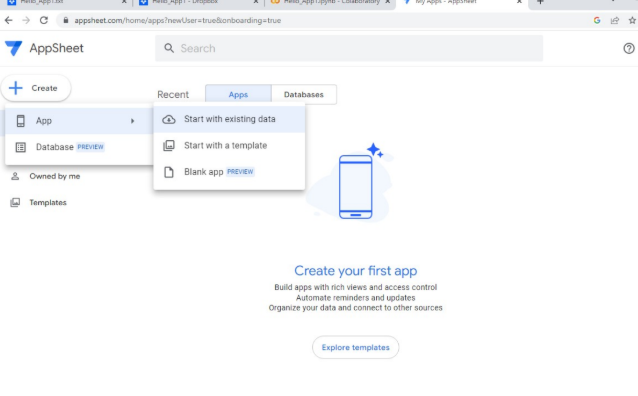
# 



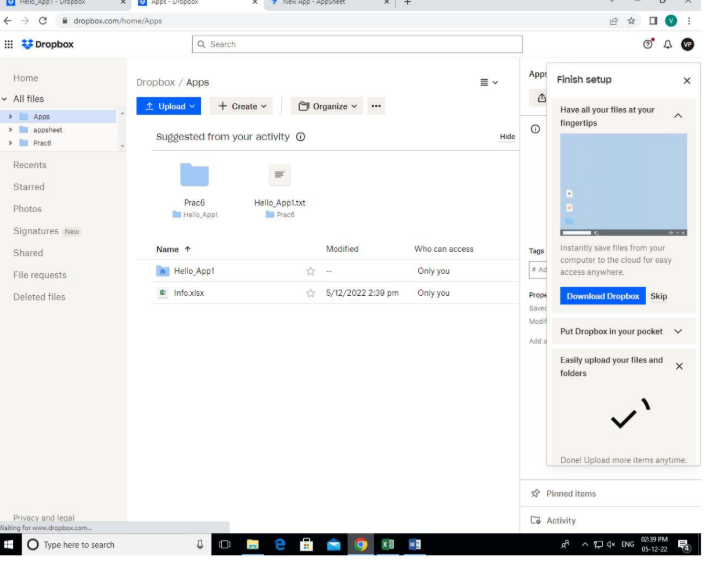
# 

****

# 



# 

****

# 

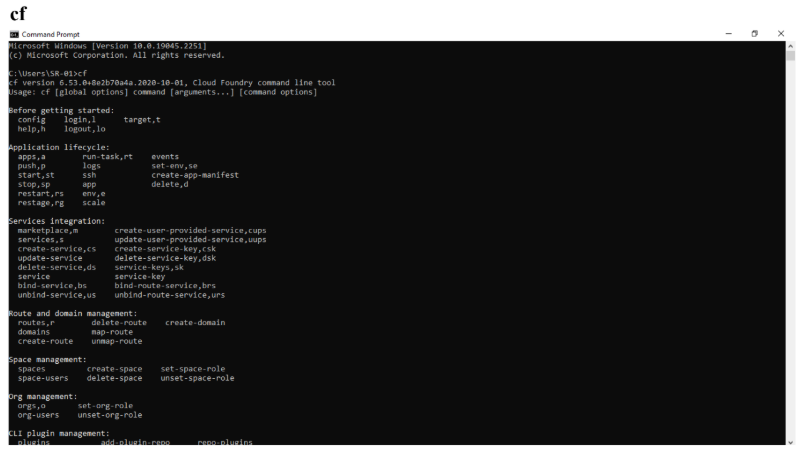
# Practical-7

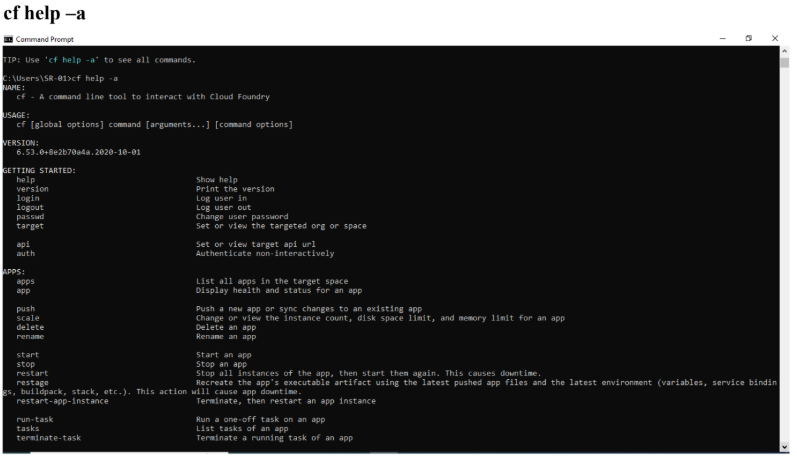
### Aim: Installing Cloud Foundry in localhost and exploring CF commands.

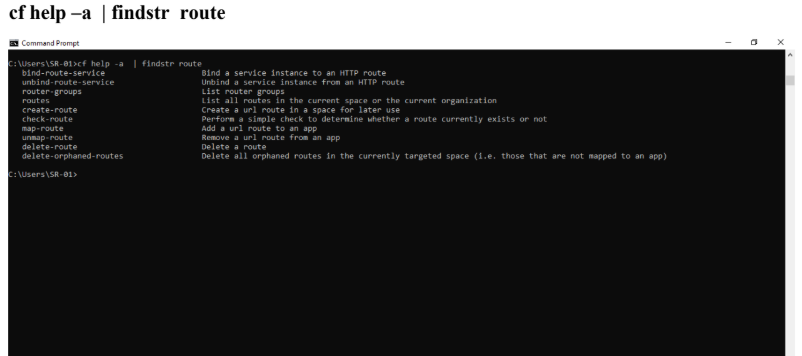
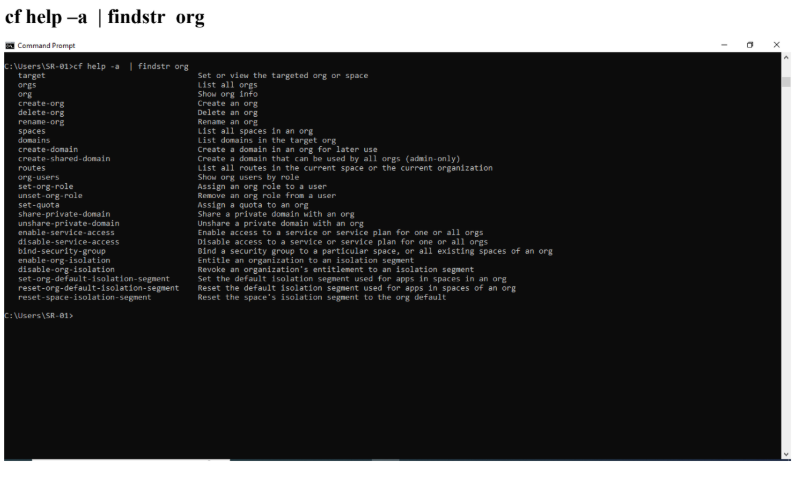
Download and install cf (Cloud Foundry) and run following commands:

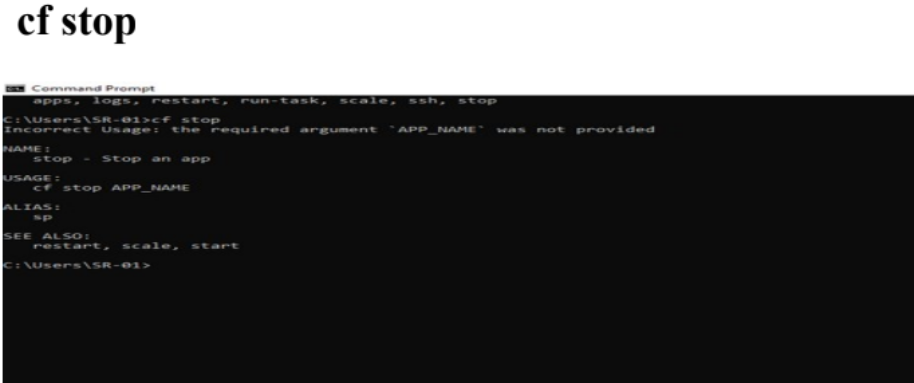
https://github.com/cloudfoundry/cli/wiki/V8-CLI-Installation-Guide#installers-and-

compressed-binaries



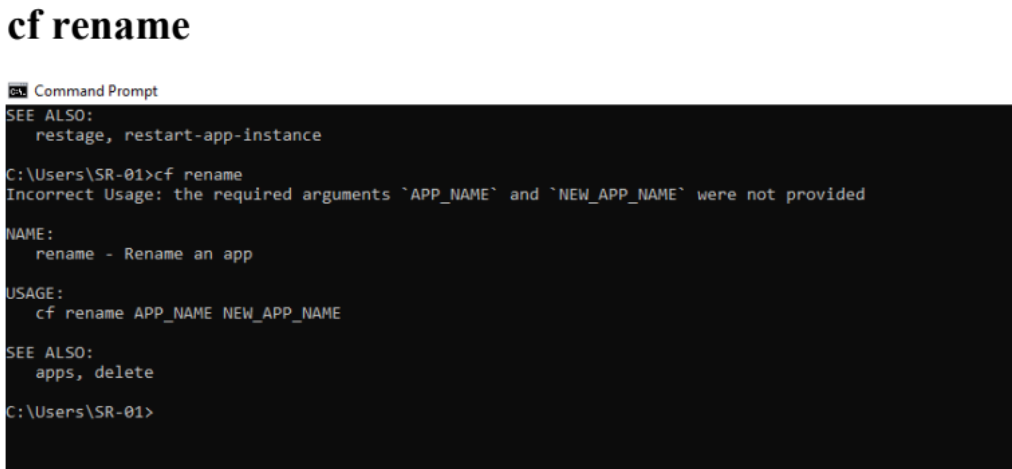




 **cf Start**

**cf stop**

**cf rename**



# Practical-8

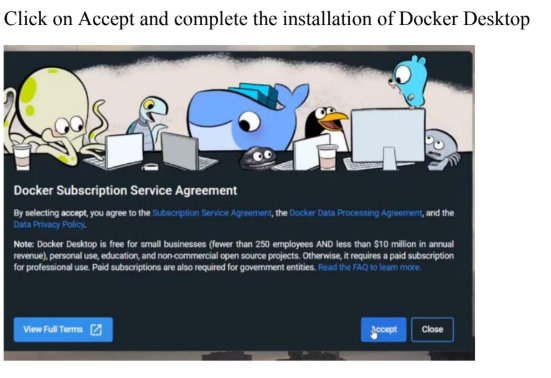
# Aim: Installing and Configuring Dockers in localhost and running multiple images on a

# Docker Platform.

**Implementation:**

1. Installing Docker Desktop https://www.docker.com/products/docker-desktop/





#### 

