C1) Install Google App Engine. Create hello world application using python/java.

import webapp2

class MainPage(webapp2.RequestHandler):

    def get(self):

        self.response.write("<h1>Hello World</h1>")

app = webapp2.WSGIApplication([('/', MainPage),], debug=True)

C2) Install Google App Engine. Create an application that prints your Name, seat number, department 5 times on separate lines using python/java. Make use of f loop.

import webapp2

class MainPage(webapp2.RequestHandler):

    def get(self):

        for i in range(5):

            self.response.write("<h1>Hello World</h1><br/>")

app = webapp2.WSGIApplication([('/', MainPage),], debug=True)

C3) Install Google App Engine. Create an application that prints your seat number, department 10 times on separate lines and using python/java. Make use of while loop.

import webapp2

class MainPage(webapp2.RequestHandler):

    def get(self):

        self.response.write("<h1>Seat No&nbsp&nbspDepartment</h1>")

        i=5

        while (i):

            self.response.write("<h1>39403&nbsp&nbspIT</h1><br/>")

            i=i-1

app = webapp2.WSGIApplication([('/', MainPage),], debug=True)

C4) Install Google App Engine. Create an application to display the table of 5 using python/java. Make use of appropriate loop. Display the table in following format

5 x 1 = 5

5 x 2 = 10

.

import webapp2

class MainPage(webapp2.RequestHandler):

    def get(self):

        for i in range(1,11):

            self.response.write("5&nbsp\*&nbsp")

            self.response.write(i);

            self.response.write("&nbsp=&nbsp");

            self.response.write(i\*5)

            self.response.write("<br/>")

app = webapp2.WSGIApplication([('/', MainPage),], debug=True)

C5) Install Google App Engine. Create an application to display the table of 10 using python/java. Make use of appropriate loop. Display the table in following format

10 x 1 = 10

10 x 2 = 20

.

import webapp2

class MainPage(webapp2.RequestHandler):

    def get(self):

        for i in range(1,11):

            self.response.write("10&nbsp\*&nbsp")

            self.response.write(i);

            self.response.write("&nbsp=&nbsp");

            self.response.write(i\*10)

            self.response.write("<br/>")

app = webapp2.WSGIApplication([('/', MainPage),], debug=T

C6) Install Google App Engine. Create an application to display the first 8 elements of the Fibonacci series using python/java.

import webapp2

class MainPage(webapp2.RequestHandler):

    def get(self):

        a=0

        b=1

        for i in range(1,9):

            c=a+b

            self.response.write(a)

            a=b

            b=c

app = webapp2.WSGIApplication([('/', MainPage),], debug=True)

C7) Create a web-app which takes the zip-code of the area the name of the branch as input, and sends the nearest post-office details. <http://www.postalpincode.in/Api-Details>

Input : Crect Pin-code , expected output: a welcome page

Input: Increct Pin-code, expected output : An err page showing proper err message and ask user to fill crect input.

Validations:-

Restrict user to enter only 6 digits. No alphabet should be displayed and no 7th digit should be accepted.

C8) Create a web-app which uses the weather API to fecast the weather by taking latitude, longitude and an ‘hourly’ parameter and/ ‘daily’ parameter.

<https://open-meteo.com/en/docs#api-documentation>

Input : Crect Input , expected output: Proper location

Input: Increct Input, expected output: An err page showing proper err message and ask user to fill crect input.

Validations:-

Restrict user to enter only required digits. No alphabet should be accepted and displayed.

<https://api.openmeteo.com/v1/forecast?latitude=52.52&longitude=13.41&hourly=temperature_2m>

“<https://api.openmeteo.com/v1/forecast?latitude=>”+latitudevalue+”&longitude=”+longitudevalue”&hourly=”+tempreturevalue

data=urllib(url).read()

data=json.loads(data)

longitude=data[“longitude”]

generationtime\_ms=data[“generationtime\_ms”]

hourly\_units=data[“hourly\_units”][“time”]

hourly\_units=data[“hourly\_units”][“temperature\_2m”]

elevation=data[“elevation”]

latitude=data[“latitude”]

C9) Can’t decide what to watch? Create a web-app which lists anime movies using the following API: GET <https://ghibliapi.herokuapp.com/films>

Input : Crect Input , expected output: Proper location

Input: Increct Input, expected output: An err page showing proper err message and ask user to fill crect input.

Validations:-

Restrict user to enter only required data.

//take the input of the film name from user

moviename=self.request.get(“moviename”);

url=”<https://ghibliapi.herokuapp.com/films?title=>”+moviename

[https://ghibliapi.herokuapp.com/films?title=Castle in the Sky](https://ghibliapi.herokuapp.com/films?title=Castle%20in%20the%20Sky)

import os

import json

import urllib

import webapp2

from google.appengine.ext.webapp import template

class MainPage(webapp2.RequestHandler):

    def get(self):

        template\_values = {}

        path = os.path.join(os.path.dirname(\_\_file\_\_), 'index.html')

        self.response.out.write(template.render(path, template\_values))

    def post(self):

        pincode = self.request.get('zipCode')

        url ="https://ghibliapi.herokuapp.com/films?title="+pincode

        data = urllib.urlopen(url).read()

        data = json.loads(data)

        id=data[0]['id']

        title=data[0]['title']

        director=data[0]['director']

        producer=data[0]['producer']

        template\_values = {

            "post\_office": id,

            "name": title,

            "block": director,

            "district": producer

        }

        path = os.path.join(os.path.dirname(\_\_file\_\_), 'results.html')

        self.response.out.write(template.render(path, template\_values))

app = webapp2.WSGIApplication([('/', MainPage)], debug=True)

data=[

{

"id": "2baf70d1-42bb-4437-b551-e5fed5a87abe",

"title": "Castle in the Sky",

"original\_title": "天空の城ラピュタ",

"original\_title\_romanised": "Tenkū no shiro Rapyuta",

"image": "https://image.tmdb.org/t/p/w600\_and\_h900\_bestv2/npOnzAbLh6VOIu3naU5QaEcTepo.jpg",

"movie\_banner": "https://image.tmdb.org/t/p/w533\_and\_h300\_bestv2/3cyjYtLWCBE1uvWINHFsFnE8LUK.jpg",

"description": "The orphan Sheeta inherited a mysterious crystal that links her to the mythical sky-kingdom of Laputa. With the help of resourceful Pazu and a rollicking band of sky pirates, she makes her way to the ruins of the once-great civilization. Sheeta and Pazu must outwit the evil Muska, who plans to use Laputa's science to make himself ruler of the world.",

"director": "Hayao Miyazaki",

"producer": "Isao Takahata",

"release\_date": "1986",

"running\_time": "124",

"rt\_score": "95",

"people": [

"https://ghibliapi.herokuapp.com/people/598f7048-74ff-41e0-92ef-87dc1ad980a9",

"https://ghibliapi.herokuapp.com/people/fe93adf2-2f3a-4ec4-9f68-5422f1b87c01",

"https://ghibliapi.herokuapp.com/people/3bc0b41e-3569-4d20-ae73-2da329bf0786",

"https://ghibliapi.herokuapp.com/people/40c005ce-3725-4f15-8409-3e1b1b14b583",

"https://ghibliapi.herokuapp.com/people/5c83c12a-62d5-4e92-8672-33ac76ae1fa0",

"https://ghibliapi.herokuapp.com/people/e08880d0-6938-44f3-b179-81947e7873fc",

"https://ghibliapi.herokuapp.com/people/2a1dad70-802a-459d-8cc2-4ebd8821248b"

],

"species": [

"https://ghibliapi.herokuapp.com/species/af3910a6-429f-4c74-9ad5-dfe1c4aa04f2"

],

"locations": [

"https://ghibliapi.herokuapp.com/locations/"

],

"vehicles": [

"https://ghibliapi.herokuapp.com/vehicles/4e09b023-f650-4747-9ab9-eacf14540cfb"

],

"url": "https://ghibliapi.herokuapp.com/films/2baf70d1-42bb-4437-b551-e5fed5a87abe"

}

]

id=data[0][“id”]

title=data[0][“title”]

director=data[0][“director”]

producer=data[0][“producer”]

release\_date=data[0][“release\_date”]

rt\_score=data[0][“rt\_score”]

C10) Use the following public API to create a web-app which searches the universities. GET [http://universities.hipolabs.com/search?name=[name](http://universities.hipolabs.com/search?name=%5Bname) of the university]

url=“[http://universities.hipolabs.com/search?name=”](http://universities.hipolabs.com/search?name=) +"University of Petroleum and Energy Studies"

http://[universities.hipolabs.com/search?name=University of Petroleum and Energy Studies](http://universities.hipolabs.com/search?name=University%20of%20Petroleum%20and%20Energy%20Studies)

import os

import json

import urllib

import webapp2

from google.appengine.ext.webapp import template

class MainPage(webapp2.RequestHandler):

    def get(self):

        template\_values = {}

        path = os.path.join(os.path.dirname(\_\_file\_\_), 'index.html')

        self.response.out.write(template.render(path, template\_values))

    def post(self):

        pincode = self.request.get('zipCode')

        url ="http://universities.hipolabs.com/search?name="+pincode

        data = urllib.urlopen(url).read()

        data = json.loads(data)

        id=data[0]["name"]

        title=data[0]["country"]

        director=data[0]["state-province"]

        producer=data[0]["web\_pages"][0]

        template\_values = {

            "post\_office": id,

            "name": title,

            "block": director,

            "district": producer

        }

        path = os.path.join(os.path.dirname(\_\_file\_\_), 'results.html')

        self.response.out.write(template.render(path, template\_values))

app = webapp2.WSGIApplication([('/', MainPage)], debug=True)

data=[{"alpha\_two\_code": "IN", "domains": ["upes.ac.in"], "country": "India", "state-province": "Dehradun", "web\_pages": ["https://www.upes.ac.in/"], "name": "University of Petroleum and Energy Studies"}]

data=urllib(url).read()

data=json.loads(data)

alpha\_two\_code=data[0][“alpha\_two\_code”]

domains=data[0][“domains”][0]

country=data[0][“country”]

state-province=data[0][“state-province”]

web\_pages=data[0][“web\_pages”][0]

name=data[0][“name”]

C11) Simulate a cloud scenario using CloudSim and run Shortest Job First (SJF) scheduling algorithm.

C12) Simulate a cloud scenario using CloudSim and run First Come First Serve (FCSF) scheduling algorithm.

C13) Simulate a cloud scenario using CloudSim and run Round-Robin scheduling algorithm.

C14) Find a procedure to transfer two text files from one virtual machine to another virtual machine. Also, add some text in the file and display it before and after transfer on both the VMs.

C15) Find a procedure to transfer two text files from one virtual machine to another virtual machine. Also, list all the files present in both the VMs.

C16) Design and develop custom Application using Sales-force Cloud.