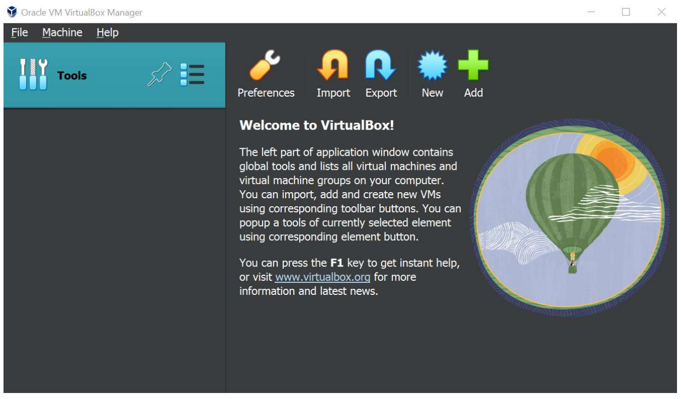
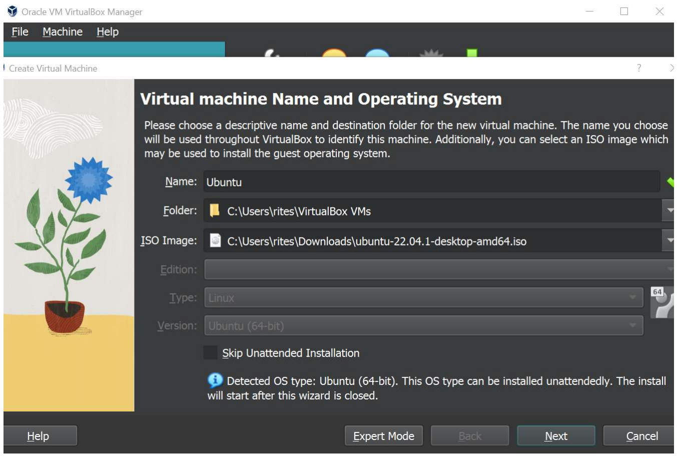
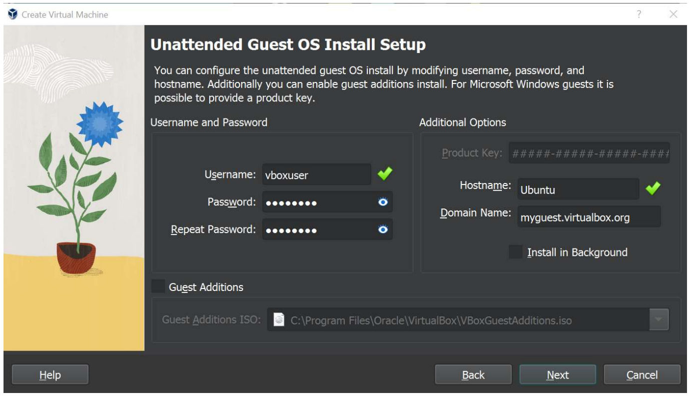
Week 11 Graded Assignment

14 July 2024 14:32

Task	Steps	Step Name	Step Details	Screenshots	Status
Task 1: Host a Ubuntu Virtual Machine using Oracle VM VirtualBox	1	Oracle VM VirtualBox Installation	Download and install Oracle VM VirtualBox.		
	2	Ubuntu ISO Download	Go to the Ubuntu website and download the Desktop version of Ubuntu as an ISO file.		
	3	Create a New Virtual Machine:	○ Open VirtualBox and click on "New" to create a new virtual machine. ○ Choose "Linux" as the type and "Ubuntu (64-bit)" as the version. ○ Allocate memory and create a virtual hard disk.	 	

Oracle VM VirtualBox Manager

Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

Create Virtual Machine

Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: Ubuntu

Folder: C:\Users\rites\VirtualBox VMs

ISO Image: C:\Users\rites\Downloads\ubuntu-22.04.1-desktop-amd64.iso

Edition:

Type: Linux

Version: Ubuntu (64-bit)

Skip Unattended Installation

Detected OS type: Ubuntu (64-bit). This OS type can be installed unattendedly. The install will start after this wizard is closed.

Create Virtual Machine

Unattended Guest OS Install Setup

You can configure the unattended guest OS install by modifying username, password, and hostname. Additionally you can enable guest additions install. For Microsoft Windows guests it is possible to provide a product key.

Username and Password

Username: vboxuser

Password: ••••••••

Repeat Password: ••••••••

Additional Options

Product Key: #####-#####-#####-####

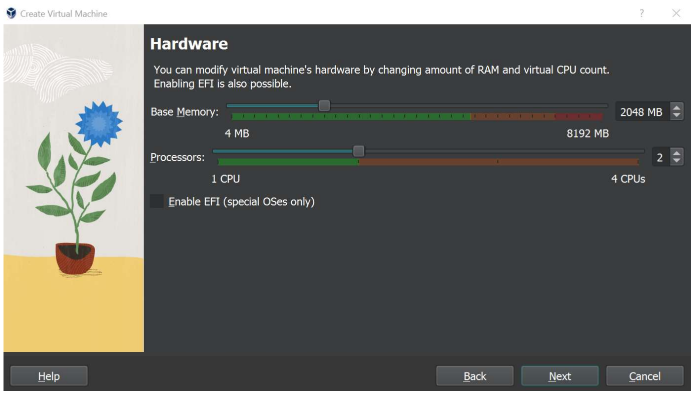
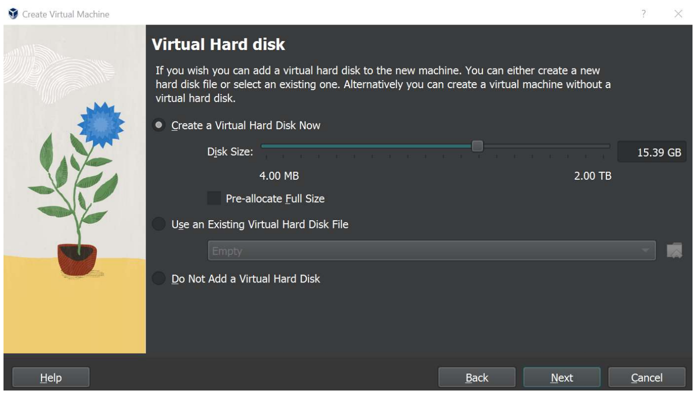
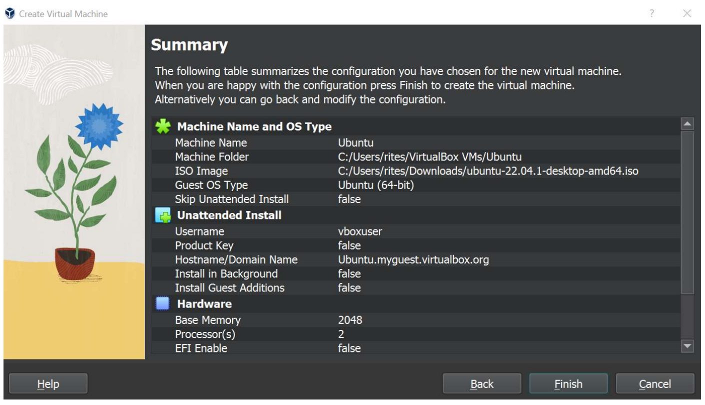
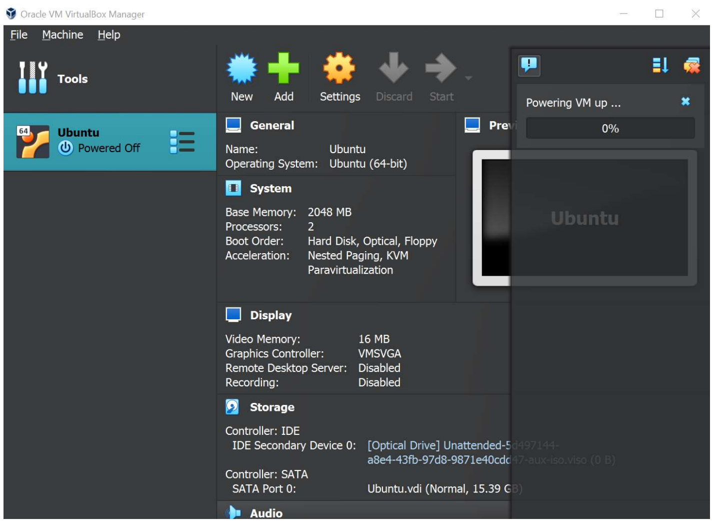
Hostname: Ubuntu

Domain Name: myguest.virtualbox.org

Install in Background

Guest Additions

Guest Additions ISO: C:\Program Files\Oracle\VirtualBox\VBoxGuestAdditions.iso

				  	
	4	Install Ubuntu	○ Select the newly created virtual machine and click "Start". ○ When prompted, select the Ubuntu ISO file you downloaded. ○ Follow the installation instructions to install Ubuntu on the virtual machine.		

Create Virtual Machine

Hardware

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Base Memory: 2048 MB (4 MB – 8192 MB)

Processors: 2 (1 CPU – 4 CPUs)

Enable EFI (special OSes only)

Help | Back | Next | Cancel

Create Virtual Machine

Virtual Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

● Create a Virtual Hard Disk Now

Disk Size: 15.39 GB (4.00 MB – 2.00 TB)

Pre-allocate Full Size

Use an Existing Virtual Hard Disk File

Empty

Do Not Add a Virtual Hard Disk

Help | Back | Next | Cancel

Create Virtual Machine

Summary

The following table summarizes the configuration you have chosen for the new virtual machine. When you are happy with the configuration press Finish to create the virtual machine. Alternatively you can go back and modify the configuration.

Machine Name and OS Type	
Machine Name	Ubuntu
Machine Folder	C:/Users/rites/VirtualBox VMs/Ubuntu
ISO Image	C:/Users/rites/Downloads/ubuntu-22.04.1-desktop-amd64.iso
Guest OS Type	Ubuntu (64-bit)
Skip Unattended Install	false
Unattended Install	
Username	vboxuser
Product Key	false
Hostname/Domain Name	Ubuntu.myguest.virtualbox.org
Install in Background	false
Install Guest Additions	false
Hardware	
Base Memory	2048
Processor(s)	2
EFI Enable	false

Help | Back | Finish | Cancel

Oracle VM VirtualBox Manager

File Machine Help

Tools | New | Add | Settings | Discard | Start

Ubuntu — Powered Off

Powering VM up ... 0%

General

Name: Ubuntu
Operating System: Ubuntu (64-bit)

System

Base Memory: 2048 MB
Processors: 2
Boot Order: Hard Disk, Optical, Floppy
Acceleration: Nested Paging, KVM Paravirtualization

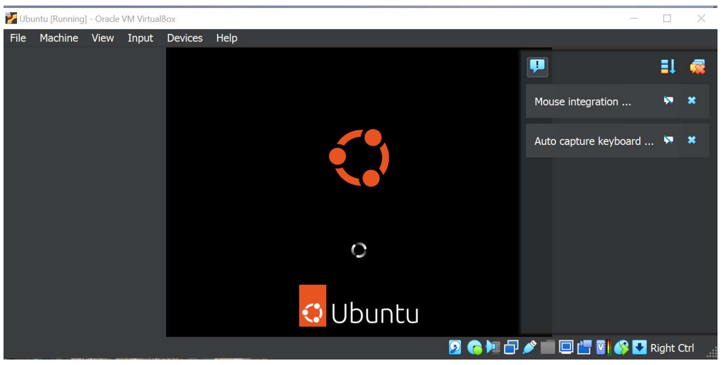
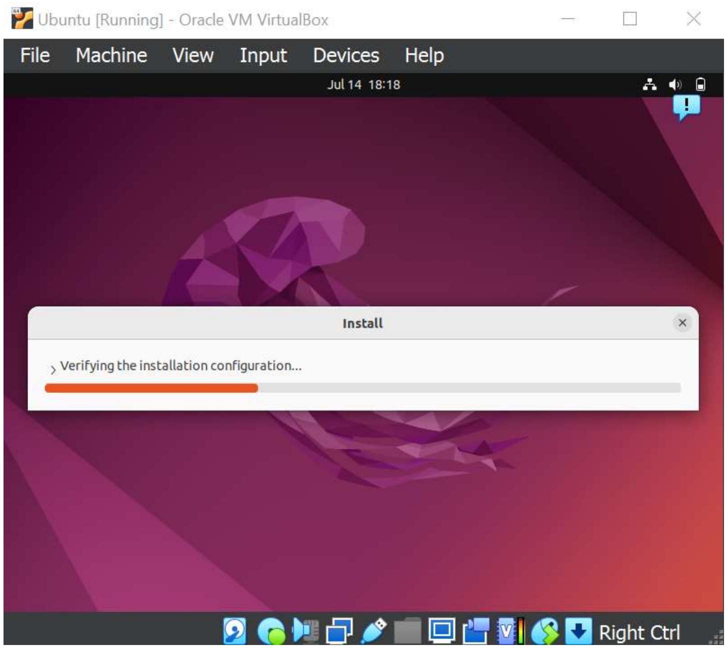
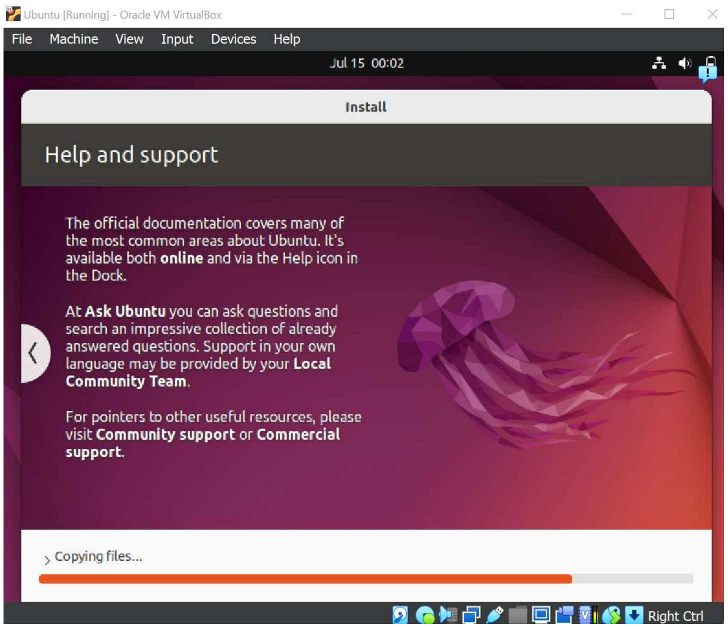
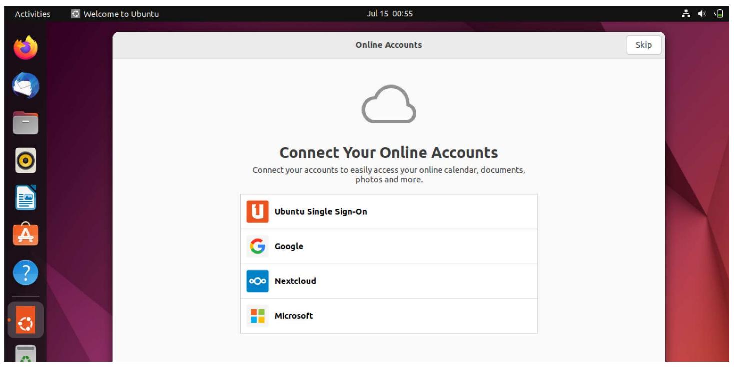
Display

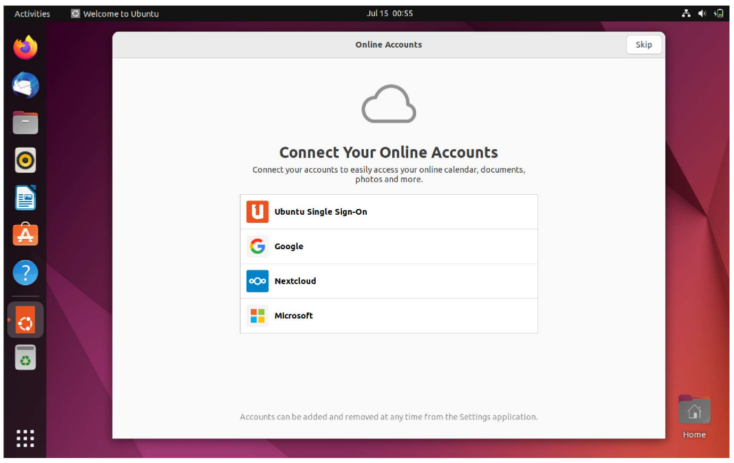
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

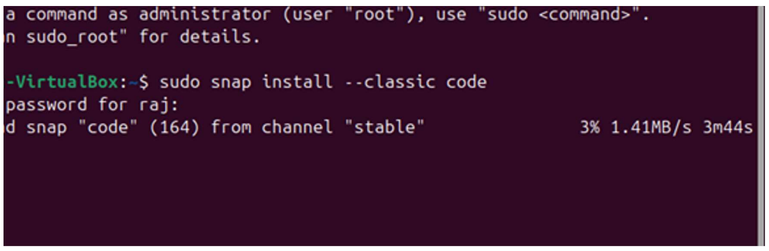
Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Unattended-5d497144-a8e4-43fb-97d8-9871e40cdc07-aux-iso.viso (0 B)
Controller: SATA
SATA Port 0: Ubuntu.vdi (Normal, 15.39 GB)

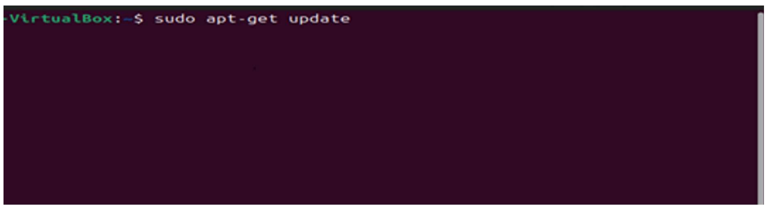
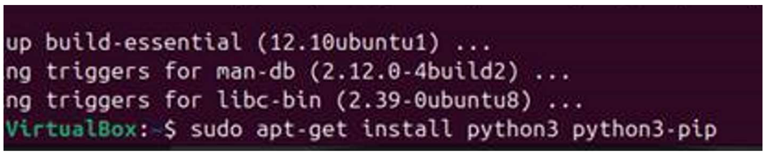
Audio

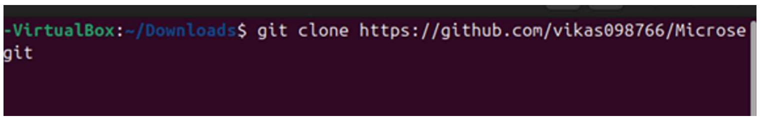
	5	Set Up Networking:	○ Ensure the virtual machine is configured to use NAT or bridged networking so it can access the internet.	 Ubuntu [Running] - Oracle VM VirtualBox File Machine View Input Devices Help Mouse integration ... Auto capture keyboard ... Ubuntu Right Ctrl  Ubuntu [Running] - Oracle VM VirtualBox File Machine View Input Devices Help Jul 14 18:18 Install Right Ctrl Verifying the installation configuration…  Ubuntu [Running] - Oracle VM VirtualBox File Machine View Input Devices Help Jul 15 00:02 Install Right Ctrl Help and support The official documentation covers many of the most common areas about Ubuntu. It's available both **online** and via the Help icon in the Dock. At **Ask Ubuntu** you can ask questions and search an impressive collection of already answered questions. Support in your own language may be provided by your **Local Community Team**. For pointers to other useful resources, please visit **Community support** or **Commercial support**. Copying files…  Activities Welcome to Ubuntu Jul 15 00:55 Online Accounts Skip Ubuntu Single Sign-On Google Nextcloud Microsoft Connect Your Online Accounts Connect your accounts to easily access your online calendar, documents, photos and more.

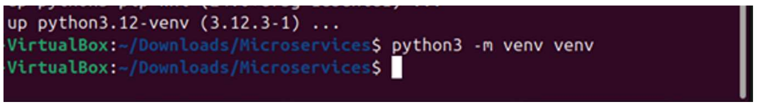
				 Activities · Welcome to Ubuntu · Jul 15 00:55 **Online Accounts** · Skip **Connect Your Online Accounts** Connect your accounts to easily access your online calendar, documents, photos and more. Ubuntu Single Sign-On Google Nextcloud Microsoft Accounts can be added and removed at any time from the Settings application. Home	

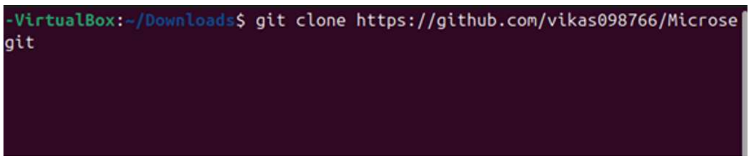
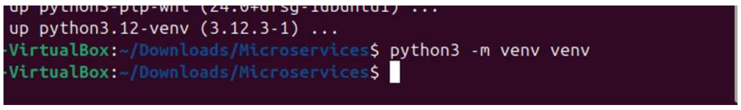
Task 2: Set up Visual Studio Code on Ubuntu VM	1	Download Visual Studio Code:	○ On Ubuntu, you can download Visual Studio Code from the official website or sudo snap install --classic code
			
			○ Open Visual Studio Code and install extensions for Python, Docker, etc., based on your requirements.
				 `a command as administrator (user "root"), use "sudo <command>".`
				`n sudo_root" for details.`
				
				`-VirtualBox:~$ sudo snap install --classic code`
				`password for raj:`
				`d snap "code" (164) from channel "stable"          3% 1.41MB/s 3m44s`	

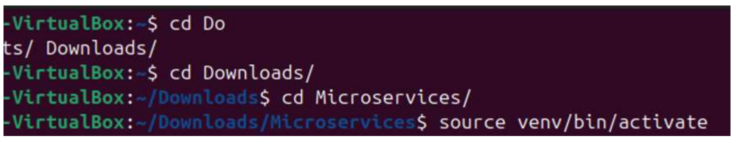
	2	Install Visual Studio Code Extensions			

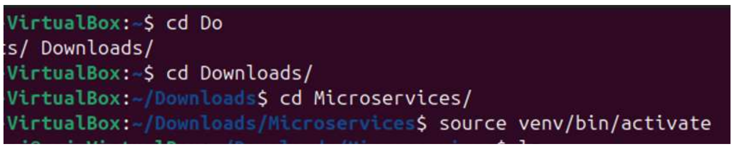
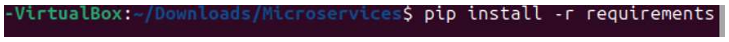
Task 3: Set up Python	1	Install Python:	○ Ubuntu typically comes with Python pre-installed. To install Python 3 and pip (Python package manager):
			
			sudo apt update
			sudo apt install python3 python3-pip
				 `VirtualBox:~$ sudo apt-get update`
				
				 `up build-essential (12.10ubuntu1) ...`
				`ng triggers for man-db (2.12.0-4build2) ...`
				`ng triggers for libc-bin (2.39-0ubuntu8) ...`
				`VirtualBox:~$ sudo apt-get install python3 python3-pip`	

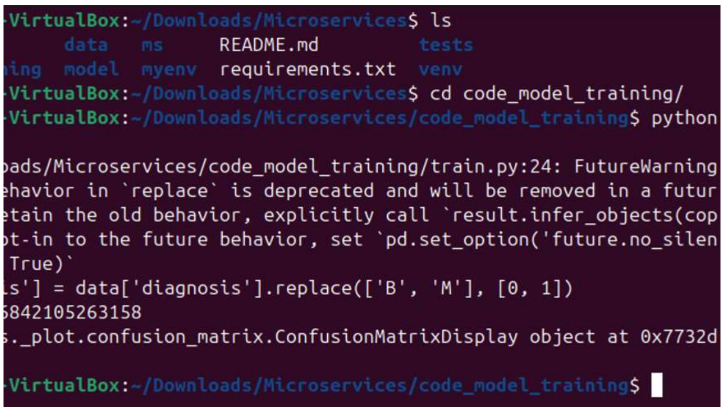
Task 4: Clone the GitHub Repository	1	Clone the Repository:	○ Open a terminal in Ubuntu and clone the repository:
			
			git clone
				 `-VirtualBox:~/Downloads$ git clone https://github.com/vikas098766/Microse`
				`git`
				
				`up python3-pip-whl (24.0+dfsg-1ubuntu1) ...`	

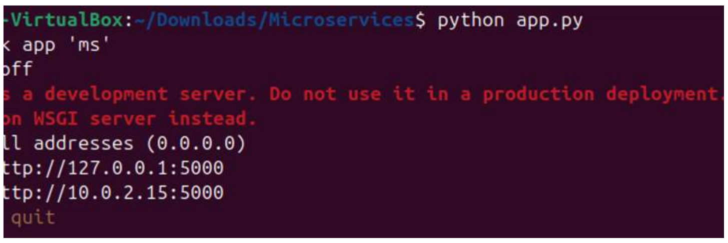
			https://github.com/Vikas098766/Microservices.git	
				`up python3-pip-whl (24.0+dfsg-1ubuntu1) ...`
				`up python3.12-venv (3.12.3-1) ...`
				`VirtualBox:~/Downloads/Microservices$ python3 -m venv venv`
				`VirtualBox:~/Downloads/Microservices$`	

Task 5: Create a Virtual Environment	1		1. Navigate to the Project Directory:
			cd Microservices
			2. Create a Virtual Environment:
			python3 -m venv venv
				
				`-VirtualBox:~/Downloads$ git clone https://github.com/vikas098766/Microse`
				`git`
				
				`up python3-pip-whl (24.0+dfsg-1ubuntu1) ...`
				`up python3.12-venv (3.12.3-1) ...`
				`VirtualBox:~/Downloads/Microservices$ python3 -m venv venv`
				`VirtualBox:~/Downloads/Microservices$`	

	2	Activate the Virtual Environment:	source venv/bin/activate	
				`-VirtualBox:~$ cd Do`
				`ts/ Downloads/`
				`-VirtualBox:~$ cd Downloads/`
				`-VirtualBox:~/Downloads$ cd Microservices/`
				`-VirtualBox:~/Downloads/Microservices$ source venv/bin/activate`	

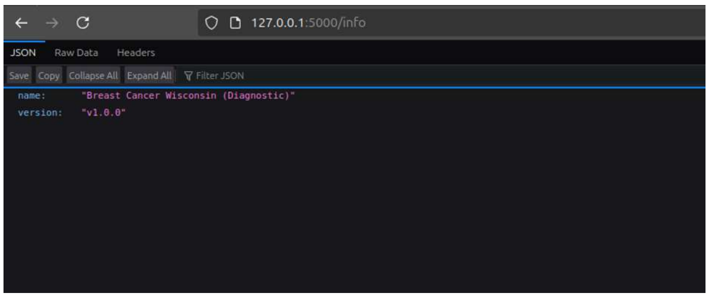
Task 6: Install Dependencies from requirements.txt			1. Install Dependencies:
			pip install -r requirements.txt
				
				`VirtualBox:~$ cd Do`
				`ts/ Downloads/`
				`VirtualBox:~$ cd Downloads/`
				`VirtualBox:~/Downloads$ cd Microservices/`
				`VirtualBox:~/Downloads/Microservices$ source venv/bin/activate`
				
				`-VirtualBox:~/Downloads/Microservices$ pip install -r requirements`	

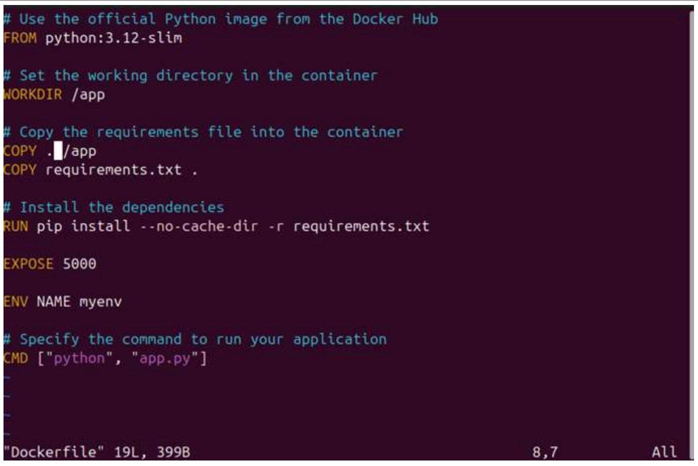
Task 7: Train and Save the Model			1. Follow the instructions specific to the repository for training the model.	
				`VirtualBox:~/Downloads/Microservices$ ls`
				`      data    ms      README.md          tests`
				`ing   model   myenv   requirements.txt   venv`
				`VirtualBox:~/Downloads/Microservices$ cd code_model_training/`
				`VirtualBox:~/Downloads/Microservices/code_model_training$ python`
				`oads/Microservices/code_model_training/train.py:24: FutureWarning`
				``ehavior in `replace` is deprecated and will be removed in a futur``
				``etain the old behavior, explicitly call `result.infer_objects(cop``
				``pt-in to the future behavior, set `pd.set_option('future.no_silen``
				`` True)` ``
				`s'] = data['diagnosis'].replace(['B', 'M'], [0, 1])`
				`5842105263158`
				`s._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7732d`
				`VirtualBox:~/Downloads/Microservices/code_model_training$`	

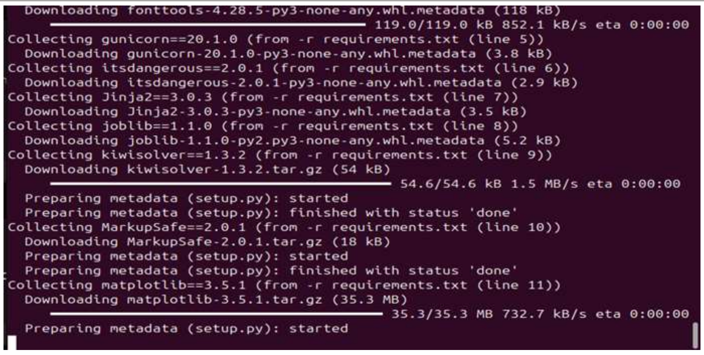
Task 8: Test the Flask Web Application	1	Run the Flask Application:	python app.py	
				`-VirtualBox:~/Downloads/Microservices$ python app.py`
				`k app 'ms'`
				`off`
				`s a development server. Do not use it in a production deployment.`
				`on WSGI server instead.`
				`ll addresses (0.0.0.0)`
				`ttp://127.0.0.1:5000`
				`ttp://10.0.2.15:5000`
				` quit`	

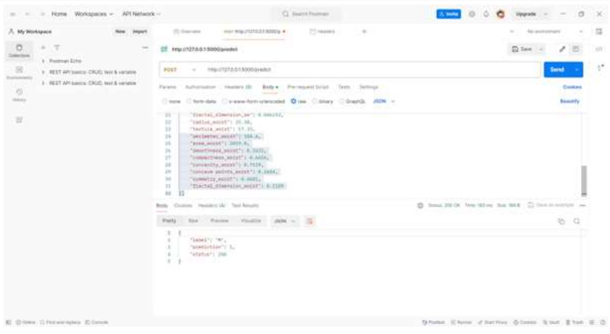
	2	Access the			

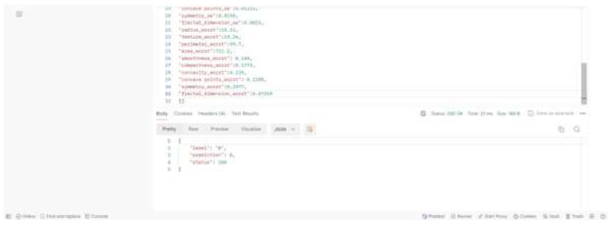
		Application:	○ Open a web browser on your host machine and go to http://localhost:5000 to test the Flask app.	

Task 9: Test the Application and Make Predictions	1	Navigate to the tests folder and run the example calls provided to test the predictions.	1. Navigate to the tests folder and run the example calls provided to test the predictions.	 127.0.0.1:5000/info
				`name:    "Breast Cancer Wisconsin (Diagnostic)"`
				`version: "v1.0.0"`

Task 10: Create a Docker Image	1	Dockerfile:	○ Create a Dockerfile in the project directory with instructions to build the Docker image.	 `# Use the official Python image from the Docker Hub`
				`FROM python:3.12-slim`
				
				`# Set the working directory in the container`
				`WORKDIR /app`
				
				`# Copy the requirements file into the container`
				`COPY . /app`
				`COPY requirements.txt .`
				
				`# Install the dependencies`
				`RUN pip install --no-cache-dir -r requirements.txt`
				
				`EXPOSE 5000`
				
				`ENV NAME myenv`
				
				`# Specify the command to run your application`
				`CMD ["python", "app.py"]`
				`~`
				`~`
				`"Dockerfile" 19L, 399B                    8,7          All`

	2	Build the Docker Image:	docker build -t myapp .	 `Downloading fonttools-4.28.5-py3-none-any.whl.metadata (118 kB)`
				`━━━━━━━━━━━━ 119.0/119.0 kB 852.1 kB/s eta 0:00:00`
				`Collecting gunicorn==20.1.0 (from -r requirements.txt (line 5))`
				`Downloading gunicorn-20.1.0-py3-none-any.whl.metadata (3.8 kB)`
				`Collecting itsdangerous==2.0.1 (from -r requirements.txt (line 6))`
				`Downloading itsdangerous-2.0.1-py3-none-any.whl.metadata (2.9 kB)`
				`Collecting Jinja2==3.0.3 (from -r requirements.txt (line 7))`
				`Downloading Jinja2-3.0.3-py3-none-any.whl.metadata (3.5 kB)`
				`Collecting joblib==1.1.0 (from -r requirements.txt (line 8))`
				`Downloading joblib-1.1.0-py2.py3-none-any.whl.metadata (5.2 kB)`
				`Collecting kiwisolver==1.3.2 (from -r requirements.txt (line 9))`
				`Downloading kiwisolver-1.3.2.tar.gz (54 kB)`
				`━━━━━━━━━━━━ 54.6/54.6 kB 1.5 MB/s eta 0:00:00`
				`Preparing metadata (setup.py): started`
				`Preparing metadata (setup.py): finished with status 'done'`
				`Collecting MarkupSafe==2.0.1 (from -r requirements.txt (line 10))`
				`Downloading MarkupSafe-2.0.1.tar.gz (18 kB)`
				`Preparing metadata (setup.py): started`
				`Preparing metadata (setup.py): finished with status 'done'`
				`Collecting matplotlib==3.5.1 (from -r requirements.txt (line 11))`
				`Downloading matplotlib-3.5.1.tar.gz (35.3 MB)`
				`━━━━━━━━━━━━ 35.3/35.3 MB 732.7 kB/s eta 0:00:00`
				`Preparing metadata (setup.py): started`

Task 11: Run the Containerized Application and Test Locally			1. Run the Docker Container: docker run -p 5000:5000 myapp 2. Test the Application: ○ Access http://localhost:5000 from your web browser to ensure the Flask app is running.	 POST http://127.0.0.1:5000/predict [illegible] POST http://127.0.0.1:5000/predict

				○ Use tools like curl or Postman to send example calls to the API running inside the Docker container.		

```
"concave points_se":0.01151,
"symmetry_se":0.0190,
"fractal_dimension_se":0.0023,
"radius_worst":15.11,
"texture_worst":19.26,
"perimeter_worst":99.7,
"area_worst":711.2,
"smoothness_worst": 0.144,
"compactness_worst":0.1773,
"concavity_worst":0.239,
"concave points_worst": 0.1288,
"symmetry_worst":0.2977,
"fractal_dimension_worst":0.07259
}]
```

```
{
    "label": "B",
    "prediction": 0,
    "status": 200
}
```