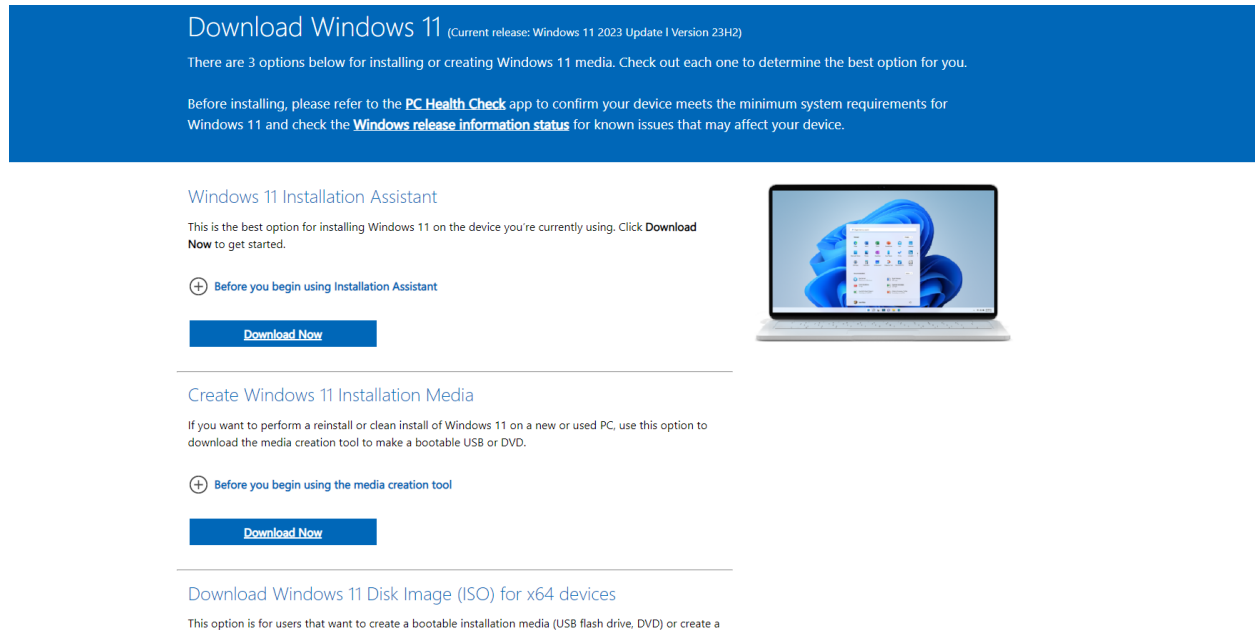


(Q1.)

Step 1: Download Windows 11

1. Visit the Windows 11 Download Page:

Go to [Windows 11 Download <https://www.microsoft.com/software-download/windows11>]



2. Select Download Option:

- There are three options available: Windows 11 Installation Assistant, Create Windows 11 Installation Media, and Download Windows 11 Disk Image (ISO). For a fresh install, we'll use the ISO method.
- Click on the "Download now" button under the "Download Windows 11 Disk Image (ISO)" section.

3. Choose Your Language:

- Select the language you want and click "Confirm."

4. Download the ISO File:

- Click on the 64-bit Download button to download the ISO file to your computer.

Step 2: Create a Bootable USB Drive

To install Windows 11, you'll need to create a bootable USB drive using the downloaded ISO file.

1. Download and Install Rufus:

- Go to [Rufus] (<https://rufus.ie/>) and download the latest version.
- Install and launch Rufus.

2. Insert a USB Drive:

- Insert a USB drive with at least 8 GB of storage into your computer.

3. Create Bootable USB:

- In Rufus, select your USB drive under "Device."
- Under "Boot selection," choose "Disk or ISO image" and click "SELECT" to browse and select the Windows 11 ISO file you downloaded.
- Leave other settings as default and click "START" to create the bootable USB drive.

Step 3: Install Windows 11

1. Restart Your Computer and Hold Shift:

- Insert the bootable USB drive into the computer where you want to install Windows 11.

- Restart your computer.

2. Enter BIOS/UEFI:

- Access the BIOS/UEFI settings by pressing the required key (usually F2, F12, Delete, or Esc) during the startup screen. The key varies depending on your computer's manufacturer.

3. Boot from USB

- In the BIOS/UEFI settings, set the USB drive as the primary boot device. Save changes and exit the BIOS/UEFI.

4. Windows 11 Installation Setup:

- Your computer should boot from the USB drive and display the Windows 11 installation setup screen.

5. Install Windows 11:

- Choose your language, time, and keyboard preferences, and click "Next."
- Click "Install Now."
- Enter your Windows 11 product key, or click "I don't have a product key" if you want to enter it later.
- Accept the license terms and click "Next."
- Choose the type of installation: "Custom: Install Windows only (advanced)" for a clean installation.
- Select the drive/partition where you want to install Windows 11 and click "Next." This will start the installation process.

6. Complete Installation:

- Follow the on-screen instructions to complete the installation. Your computer will restart several times during the process.

7. Initial Setup:

After installation, you'll go through the initial setup process, including setting up your Microsoft account, privacy settings, and other preferences.

3. Check for Updates:

- Click "Check for updates" to download and install the latest updates.

(Q2.)

Step 1: Download Visual Studio Code

1. Go to the Visual Studio Code Download Page

- Open your web browser and go to [Visual Studio Code - Download]<https://code.visualstudio.com/Download>)

2. Select Your Operating System:

- Click on the appropriate download link for your operating system (Windows, macOS, or Linux). In this guide, we will focus on Windows.

Step 2: Install Visual Studio Code

1. Run the Installer:

- Once the download is complete, open the downloaded file to run the installer. You might see a security warning; click "Run" to proceed.

2. Accept the License Agreement:

- Read through the license agreement, check the box to accept the terms, and click "Next."

3. Select Installation Location:

- Choose the destination folder where you want to install Visual Studio Code, or leave it at the default location, and click "Next."

4. Select Additional Tasks:

- You can choose additional tasks such as creating a desktop icon, adding an "Open with Code" action to the Windows Explorer file context menu, etc. Check the options you prefer and click "Next."

5. Install Visual Studio Code:

- Click the "Install" button to begin the installation process.

6. Finish Installation:

- Once the installation is complete, check the option to "Launch Visual Studio Code" and click "Finish."

Step 3: Launch Visual Studio Code

1. First Launch:

- After finishing the installation, Visual Studio Code will launch automatically. You can also launch it by clicking the desktop icon or finding it in the Start menu.

2. Explore the Interface:

- Take a moment to familiarize yourself with the VS Code interface. Key components include the Activity Bar (left side), Side Bar, Editor, and Status Bar (bottom).

Step 4: Install Extensions

1. Open Extensions View:

- Click on the Extensions view icon on the Activity Bar or press ``Ctrl+Shift+X``.

2. Search and Install Extensions:

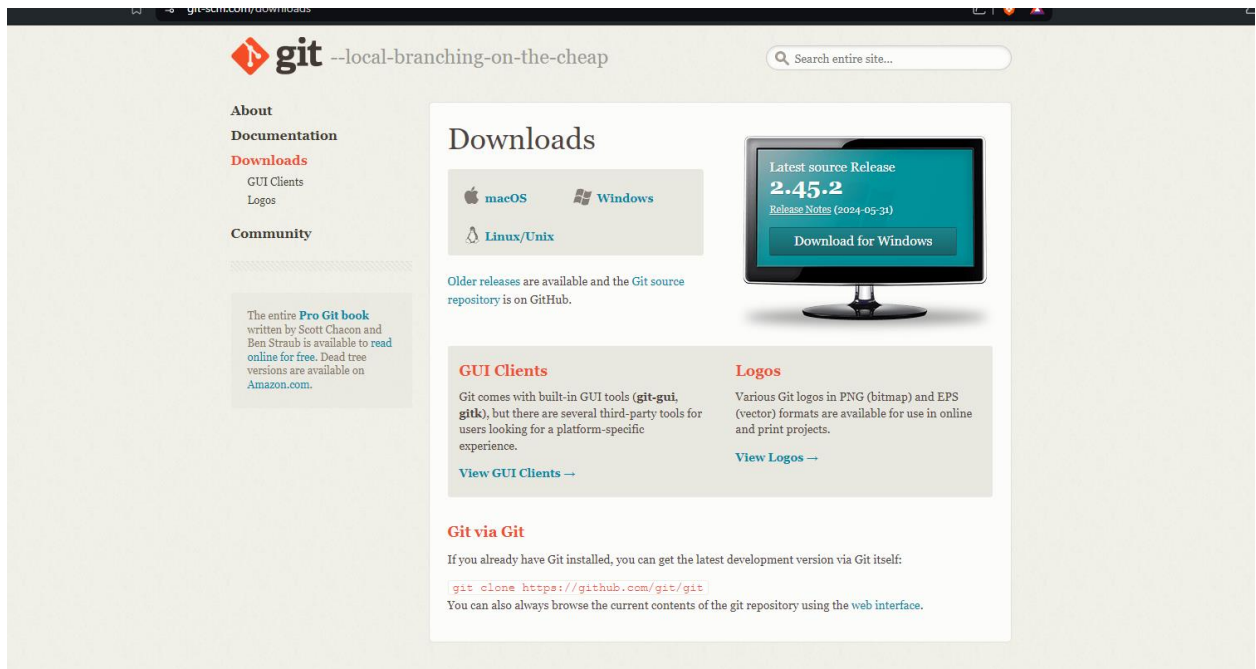
- Use the search bar to find extensions for the languages and tools you use. For example, search for "Python" and click "Install" to add the Python extension.

(Q3.)

Step 1: Install Git

1. Download Git:

- Go to [Git - Download] <https://github.com> in your web browser.
- The website should automatically detect your operating system. If not, select your operating system (Windows, macOS, Linux) from the options provided.



2. Run the Installer:

- Once the download is complete, run the downloaded installer file.
- If you're using Windows, you might see a User Account Control prompt asking if you want to allow the app to make changes to your device. Click "Yes" to continue.

3. Configure Installation Settings:

- Follow the prompts in the installer:
- Select the components to install (leave default unless you have specific preferences).
- Choose the installation location (default is recommended).
- Select your preferred editor for Git (default is recommended).
- Choose the default branch name (leave as `master` unless you have a specific reason to change it).

- Choose HTTPS transport backend (leave as default unless you have reasons to use another option).

4. Complete the Installation:

- Click "Next" through the remaining prompts and then click "Install."
- Once the installation is complete, click "Finish."

5. Verify the Installation:

Open a command prompt (on Windows, you can use Git Bash which was installed with Git).


- Verify the installation by typing the following command and pressing Enter:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\james> git remote add origin https://github.com/yourusername/your-repository.git|
```


You should see the Git version number printed in the command prompt, confirming that Git was installed successfully.

A screenshot of a Windows command prompt window. The prompt shows the user 'james' at the 'C:\Users\' directory. The command 'git --version' has been entered and executed, resulting in the output 'git version 2.45.2.windows.1'. The prompt is now waiting for the next command.

```
S C:\Users\james> git --version
git version 2.45.2.windows.1
S C:\Users\james> |
```

Step 2: Configure Git

1. Set Up Your Identity:

- Configure your Git username and email address. Open Git Bash (or your command prompt) and run the following commands, replacing `"Your Name"` and `"your.email@example.com"` with your GitHub username and email address:

```
Windows PowerShell
S C:\Users\james> git --version
git version 2.45.2.windows.1
S C:\Users\james> git config --global user.name "Your Name"
S C:\Users\james> git config --global user.email "your.email@example.com"
S C:\Users\james> |
```

2. Check Your Configuration

- You can check your Git configuration settings by running:

```
Windows PowerShell
PS C:\Users\james> git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Your Name
user.email=your.email@example.com
PS C:\Users\james> |
```

###

Step 3: Create a GitHub Account

1. Visit GitHub:

- Go to [GitHub] (<https://github.com>) in your web browser.

2. Sign Up:

- Click on "Sign up" at the top right corner of the GitHub homepage.
- Follow the prompts to create your GitHub account. You'll need to provide a username, email address, and password.

Step 4: Initialize a Git Repository and Make Your First Commit

1. Create a New Project Directory:

- Create a new directory on your local machine where you want to store your project files.

2. Initialize a Git Repository:

- Open Git Bash (or your command prompt) and navigate (`cd``) to your project directory.
- Run the following command to initialize a new Git repository:

```
james@DESKTOP-JLP0FM3 MINGW64 ~  
$ git init  
cls  
Initialized empty Git repository in C:/Users/james/.git/  
bash: cls: command not found  
  
james@DESKTOP-JLP0FM3 MINGW64 ~ (master)  
$ |
```

- This command initializes an empty Git repository in your current directory.

3. Create Some Files:

- Create some files or add existing files into your project directory.

4. Add Files to the Staging Area:

- To start tracking these files with Git, add them to the staging area. For example, to add all files, use:

```
james@DESKTOP-JLP0FM3 MINGW64 ~ (master)  
$ git add .
```

5. Commit Your Changes:

- Once files are staged, commit them to the Git repository with a commit message describing your changes. For example:
-

```
git commit -m "Initial commit"
```

- Replace `"Initial commit"` with a meaningful message that describes the changes you've made.

Step 5: Link Local Repository to GitHub Repository

1. Create a New Repository on GitHub:

- Log in to your GitHub account.
- Click on the "+" icon in the top right corner and select "New repository."
- Give your repository a name, optionally provide a description, choose visibility (public or private), and click "Create repository."

2. Add Remote Repository:

- After creating the repository on GitHub, you'll see instructions on how to push an existing repository from the command line.
- Copy the command under "...or push an existing repository from the command line" that looks similar to:

```
bash`
```

- Paste this command into Git Bash (or your command prompt) in your project directory to add your GitHub repository as the remote origin.

3. Push Your Changes to Git

- Finally, push your committed changes from your local repository to GitHub:

```
bash
```

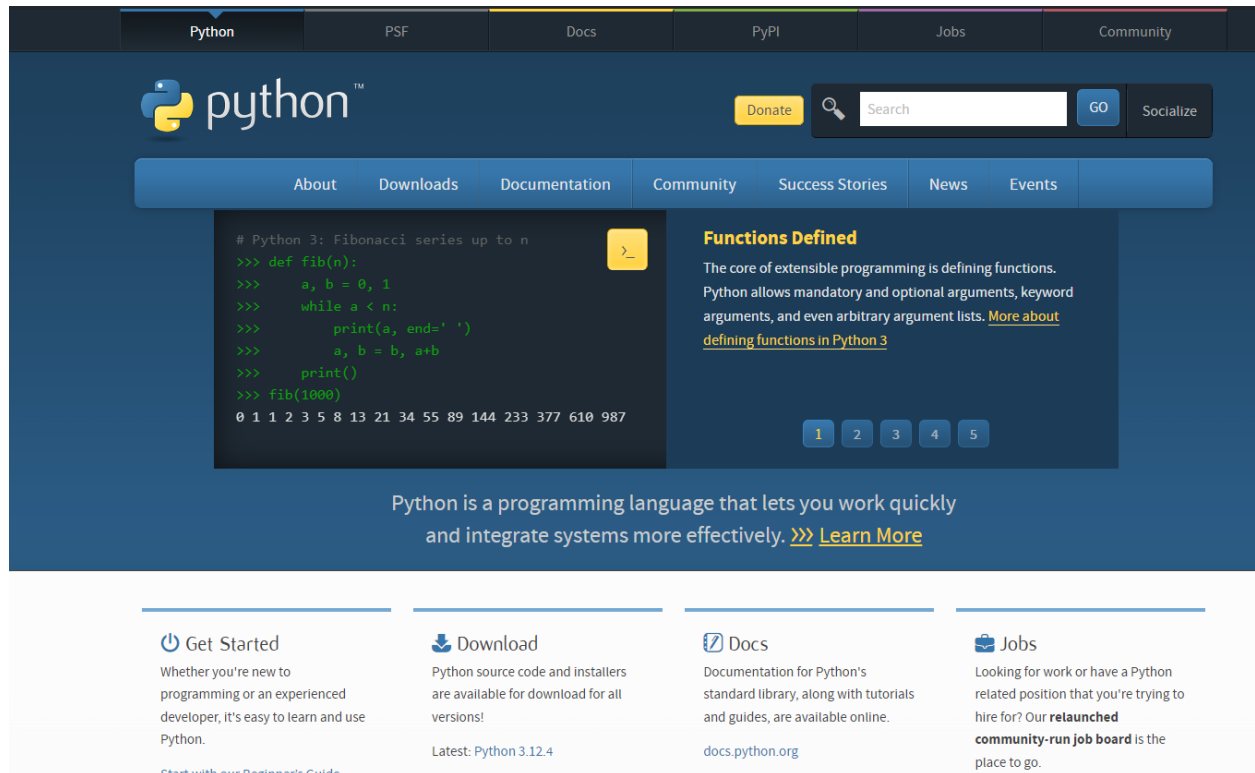
```
git push -u origin master
```

(Q4.)

Step 1: Download Python

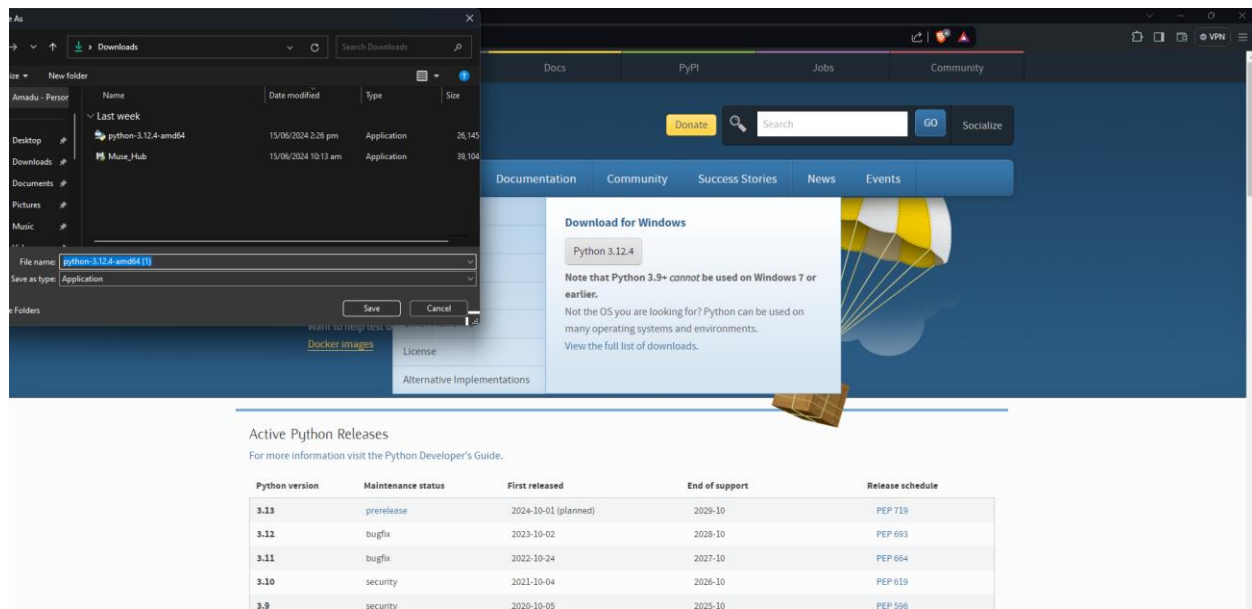
1. Visit the Python Official Website:

Go to [Python. org] (<https://www.python.org>) in your web browser.



2. Download Python:

- On the Python homepage, click on the "Downloads" menu at the top of the page.



3. Choose Python Version:

- Python releases come in two major versions: Python 2 and Python 3. Python 2 is deprecated, so it's recommended to download Python 3.
- Click on the latest version of Python 3.x (e.g., Python 3.9.6) to download it.

4. Download Installer:

- Scroll down to find the installer for your operating system (Windows, macOS, or Linux).
- Click on the installer link appropriate for your system to start the download.

Step 2: Install Python

1.*Run the Installer:

- Once the download is complete, run the downloaded installer file.
- If you're using Windows, you may need to confirm that you want to allow the installer to make changes to your system.

2. Configure Python Installation:

- In the Python installer, make sure to check the box that says "Add Python to PATH." This will allow you to run Python from the command line.
- Optionally, you can customize the installation by choosing which features to install or changing the installation location. For most users, the default settings are fine.

3. Complete the Installation:

- Click "Install Now" to begin the installation process.
- Wait for the installation to complete. This may take a few moments.

4. Verify Python Installation:

Once installed, you can verify the installation by opening a command prompt (on Windows, you can use Command Prompt or PowerShell) and typing:

A screenshot of a terminal window with a black background and green text. The prompt is 'james@DESKTOP-JLP0FM3 MINGW64 ~ (master)'. The user has entered the command '\$ python --version' and the output is 'Python 3.12.4'. The prompt is repeated below the output.

```
james@DESKTOP-JLP0FM3 MINGW64 ~ (master)
$ python --version
Python 3.12.4
james@DESKTOP-JLP0FM3 MINGW64 ~ (master)
$
```

- This command should print the version of Python installed, confirming that Python is now available on your system.

Step 3: Install Additional Tools (Optional)

Depending on your project requirements, you may need to install additional tools such as:

- Python package manager (pip): Python typically comes with pip installed. You can verify it by typing `pip version` in your command prompt.
- -Python Integrated Development Environment (IDE):** Consider installing an IDE like Visual Studio Code (already covered in a previous response) or PyCharm for a more integrated development experience.
- Virtual Environment: It's good practice to use virtual environments for Python projects to manage dependencies and isolate project environments.

(Q5.)

Step 1: Download MySQL Installer

1. Visit MySQL Download Page:

Go to [MySQL Installer

Download](<https://dev.mysql.com/downloads/windows/installer/5.7.html>) in your web browser.

The screenshot shows the MySQL Community Downloads page for the MySQL Installer. The page has a header with the MySQL logo and the text "MySQL Community Downloads". Below the header, there is a breadcrumb trail: "MySQL Installer". The main content area is titled "MySQL Installer 5.7.44". It includes a note about MySQL 8.0 being the final series with the MySQL Installer, and a recommendation to use the MSI or Zip archive for installation. Below the note, there are two dropdown menus: "Select Version:" with "5.7.44" selected, and "Select Operating System:" with "Microsoft Windows" selected. Below these, there is a table with two rows of download links. The first row is for "Windows (x86, 32-bit), MSI Installer" with a size of 2.1M and a download button. The second row is for "Windows (x86, 32-bit), MSI Installer" with a size of 373.7M and a download button. Below the table, there is a note suggesting to use MD5 checksums and GnuPG signatures to verify the integrity of the packages. At the bottom of the page, there is a footer with the Oracle logo and the text "© 2024 Oracle".

MySQL Community Downloads

MySQL Installer

General Availability (GA) Releases Archives

MySQL Installer 5.7.44

Note: MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server.

Select Version:
5.7.44

Select Operating System:
Microsoft Windows

Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-5.7.44.0.msi)	5.7.44	2.1M	Download MD5: 6cc27e2a42a54b593a9d3544f2529a53 Signature
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-5.7.44.0.msi)	5.7.44	373.7M	Download MD5: e89af3ba9bb4716ff5e647b0fd2edab2 Signature

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

ORACLE © 2024 Oracle

Privacy / Do Not Sell My Info | Terms of Use | Trademark Policy

2. Choose MySQL Installer Version:

- Scroll down to find the MySQL Installer appropriate for your Windows version. Look for the version labeled "MySQL Installer for Windows."

3. Download MySQL Installer:

Step 2: Install MySQL Server Using MySQL Installer

1. Run the MySQL Installer:

- Once the download is complete, locate the downloaded `.msi` file (e.g., `mysql-installer-community-5.7.35.0.msi`) and double-click to run it.
- If prompted by User Account Control, click "Yes" to allow the installer to make changes to your system.

2. Choose Setup Type:

- In the MySQL Installer welcome screen, click "Install MySQL Products."

3. Select Products:

- Choose "Server only" or "Full" (includes tools and connectors) based on your requirements. For most cases, "Server only" is sufficient.

4. Check Requirements and Install:

- Review the prerequisites and system checks. If any prerequisites are missing, the installer will prompt you to install them.

5. Configure MySQL Server:

- On the "Type and Networking" screen, leave the default options unless you have specific requirements.
- Set the MySQL root password. Choose a strong password and remember it, as you will need it to access MySQL.

6. Complete the Installation

- Continue through the installer, accepting the default configuration options unless you have specific reasons to change them.
- Click "Execute" to begin the installation process. Wait for the installation to complete.

7. Finish Installation:

- Once the installation completes successfully, click "Next" and then "Finish" to exit the installer.

Step 3: Verify MySQL Installation

1. Open MySQL Command Line Client:

- Open the MySQL Command Line Client from the Start Menu or by searching for "MySQL Command Line Client" in the search bar.
- You may need to enter the MySQL root password you set during installation.

2. Verify MySQL Server Status:

- Once connected to the MySQL Command Line Client, type the following command to check if the MySQL server is running:

```
```sql
```

```
SHOW VARIABLES LIKE 'version';
```

...

- This command should display the MySQL server version, confirming that MySQL is installed and running correctly.

## **Step 4: Optional - Install MySQL Workbench (GUI Tool)**

### **1. Download MySQL Workbench:**

- If you prefer a graphical user interface (GUI) for managing MySQL, you can download MySQL Workbench from [MySQL Workbench Download](<https://dev.mysql.com/downloads/workbench/>).

### **2. Install MySQL Workbench:**

- Run the downloaded installer (`.msi` file) and follow the installation instructions.
- MySQL Workbench allows you to visually administer MySQL databases, manage SQL development, and perform database design and modeling tasks.

## **Step 5: Connect MySQL to Your Projects**

### **1. Connect from Applications:**

- Use MySQL Connector/ODBC or MySQL Connector/J depending on your programming language (Java, Python, etc.) to connect your applications to MySQL databases.

### **2. Create and Manage Databases:**

- Use MySQL Workbench or MySQL Command Line Client to create databases, tables, and manage data according to your project requirements.

**(Q6.)**

### **Step 1: Install Docker Desktop**

#### 1. Download Docker Desktop:

- Go to [Docker Desktop](<https://www.docker.com/products/docker-desktop>) in your web browser.

#### 2. Choose Your Operating System:

- Click on "Download for Windows" if you are using Windows. Docker Desktop is also available for macOS and Linux.

Download](<https://www.docker.com/sites/default/files/d8/2019-07/hub-hero-windows.png>)

#### 3. Run the Installer:

- Once the download is complete, run the Docker Desktop installer (`.exe` file).
- Follow the prompts in the installer to complete the installation. This may involve accepting terms and conditions and configuring Docker settings.

## Step 2: Verify Docker Installation

### 1. Start Docker Desktop:

- After installation, Docker Desktop should start automatically. You should see the Docker whale icon in the system tray (Windows) or status bar (macOS).

### 2. Open Docker Dashboard:

- Click on the Docker icon to open the Docker Dashboard. This dashboard provides an overview of your Docker environment and containers.
- 

### 3. Verify Docker Version:

- Open a terminal (Command Prompt, PowerShell, or Terminal on macOS) and type:

```
```bash
```

```
docker --version
```

```
```
```

- This command should display the Docker version installed, confirming that Docker is ready to use.

## Step 3: Create and Run a Docker Container

### 1. Pull a Docker Image

- Docker containers are launched from Docker images. To start, you can pull a basic image like Ubuntu from Docker Hub. Open a terminal and run:

```
```bash
```

```
docker pull ubuntu
```

```
```
```

### 2. Run a Docker Container:

- Once the image is downloaded, you can run a container based on that image. For example, to start an interactive Ubuntu container:

```
```bash
```

```
docker run -it ubuntu bash
```

```
```
```

- -This command starts a new Docker container interactively (`-it`) using the `ubuntu` image and opens a Bash shell (`bash`) inside the container.

### 3. \*\*Explore Docker Commands:\*\*

- Docker provides a rich set of commands for managing containers, images, networks, volumes, and more. Some useful commands include `docker ps` (list running containers), `docker images` (list downloaded images), `docker stop` (stop a running container), and `docker rm` (remove a container).

## ### Step 4: Use Docker for Development

## 1. **\*\*Dockerize Your Application:\*\***

- Create a `Dockerfile` in your project directory to define your application's environment and dependencies. Here's a basic example for a Python application:

```
```dockerfile
```

```
# Use an official Python runtime as a parent image
```

```
FROM python:3.9-slim
```

```
# Set the working directory in the container
```

```
WORKDIR /app
```

```
# Copy the current directory contents into the container at /app
```

```
COPY . /app
```

```
# Install any needed packages specified in requirements.txt
```

```
RUN pip install --no-cache-dir -r requirements.txt
```

```
# Make port 80 available to the world outside this container
```

```
EXPOSE 80
```

```
# Define environment variable
```

```
ENV NAME World
```


Run app.py when the container launches

```
CMD ["python", "app.py"]
```

```
...
```

Replace `app.py` with your Python application file and `requirements.txt` with your dependencies file.

2. ****Build and Run Your Docker Image:****

- Build your Docker image from the `Dockerfile` in your project directory:

```
```bash
```

```
docker build -t my-python-app .
```

```
...
```

- This command builds a Docker image named `my-python-app` using the `Dockerfile` in the current directory (`.`).

## 3. Run Your Docker Container

- Once built, you can run your Docker container based on the image:

```
```bash
```

```
docker run -p 4000:80 my-python-app
```

```
...
```

- This command starts a container named `my-python-app`, mapping port `4000` on your local machine to port `80` inside the container.

Step 5: Integration with Version Control and Deployment

1. Link Docker with Version Control (Optional):

- You can link your Dockerized application with version control systems like Git by including Docker configurations (`Dockerfile` and `docker-compose.yml``) in your project repository.

2. Deploy Docker Containers:

- Docker containers can be easily deployed to various platforms such as Docker Hub, Amazon ECS, Kubernetes, etc., for production environments.

.

(Q7.)

Developer Environment Setup Documentation

Overview

This document provides a detailed guide on setting up the developer environment for [Your Project Name]. It includes installation steps for necessary tools, configurations made, customizations applied, and any troubleshooting encountered during the setup process.

Table of Contents

1. [Operating System Setup](#operating-system-setup)
2. [Text Editor or IDE Installation](#text-editor-or-ide-installation)
3. [Version Control System Setup](#version-control-system-setup)
4. [Programming Languages and Runtimes Installation](#programming-languages-and-runtimes-installation)

5. [Database Setup](#database-setup)
6. [Development Environments and Virtualization](#development-environments-and-virtualization)
7. [Documentation of Setup](#documentation-of-setup)

1. Operating System Setup

- Operating System: Windows 11-Installation Steps:
- Downloaded Windows 11 from [Microsoft's official website](https://www.microsoft.com/software-download/windows11).
- Created a bootable USB drive using the Windows Media Creation Tool.
- -Installed Windows 11 on a new partition, preserving existing data on other partitions.

2. Text Editor or IDE Installation

- Text Editor/IDE: Visual Studio Code
- Installation Steps: Downloaded Visual Studio Code from [Visual Studio Code website](https://code.visualstudio.com/Download).
- Ran the installer and followed on-screen prompts to complete the installation.
- Installed recommended extensions for Python development (e.g., Python, GitLens, Code Runner).

3. Version Control System Setup

- Version Control System: Git
- Installation Steps:**
- Downloaded Git from [Git official website](https://git-scm.com/download).
- Ran the installer and selected default options.
- Configured Git username and email globally using Git Bash:

```
```bash
```

```
git config --global user.name "Your Name"
```

```
git config --global user.email "your.email@example.com"
```

- Initialized a Git repository for the project and made initial commits.

#### 4. Programming Languages and Runtimes Installation

- Programming Language:\*\* Python
- Installation Steps:\*\*
- Downloaded Python installer from [Python official website](https://www.python.org/downloads/).
- Ran the installer, selecting "Add Python to PATH" option and installing for all users.
- Installed additional Python packages using pip:

```
```bash
```

```
pip install package_name
```

```
```
```

- Verified Python installation and package versions.

#### 5. Database Setup

- Database: MySQL
- Installation Steps:
- Downloaded MySQL Installer from [MySQL official website](https://dev.mysql.com/downloads/windows/installer/).

- Installed MySQL Server using MySQL Installer, configuring root password and port settings.
- Installed MySQL Workbench for database management and query execution.
- Tested MySQL connection from command line and MySQL Workbench.

## 6. Development Environments and Virtualization

### Virtualization Tool: Docker (Optional)

- Setup Steps:
- Installed Docker Desktop from [Docker official website](<https://www.docker.com/products/docker-desktop>).
- Configured Docker settings including resource allocation and networking.
- Created Dockerfiles for Dockerizing applications and used docker-compose for multi-container setups.
- Ran Docker containers for development and testing environments.

## 7. Documentation of Setup

- Documentation Steps:
- Created this document to record all setup steps, configurations, and customizations made.
- Included screenshots where necessary to illustrate installation and configuration steps.
- Documented any issues encountered during setup and the solutions applied.

- Organized the document with clear headings and a table of contents for easy navigation.