Setting Up My Developer Environment

Tasks

1. Select Your Operating System (OS)

Task: Choose an operating system and install it.

Step-by-Step Instructions:

1. **Download Windows 11:**
    - Visit the official Microsoft Windows 11 download page: Windows 11 Download.
    - Click on the "Download Now" button and follow the prompts to download the installation media creation tool.
2. **Install Windows 11:**
    - Run the downloaded tool and create installation media (USB flash drive or ISO file).
    - Follow the on-screen instructions to complete the installation process on your machine.

2. Install a Text Editor or Integrated Development Environment (IDE)

Task: Select and install a text editor or IDE. For this, I used Visual Studio Code (VS Code).

Step-by-Step Instructions:

1. **Download Visual Studio Code:**
    - Visit the official Visual Studio Code download page: VS Code Download.
    - Download the installer for your operating system (Windows).
2. **Install Visual Studio Code:**
    - Run the downloaded installer and follow the setup wizard instructions.
    - Launch VS Code after installation.

3. Set Up Version Control System

Task: Install Git, configure it, create a GitHub account, initialize a Git repository, and make your first commit.

Step-by-Step Instructions:

1. **Install Git:**
    - Visit the official Git download page: Git Download.
    - Download and run the installer, following the default settings.

2. **Configure Git:**
    - Open Git Bash (installed with Git).

Set your username and email:

```
git config --global user.name "kepher99"
git config --global user.email "otienokepher9@gmail.com"
```

3. **Create a GitHub Account:**
    - Visit GitHub and sign up for an account.
4. **Initialize a Git Repository:**
    - Open Git Bash in your project directory.

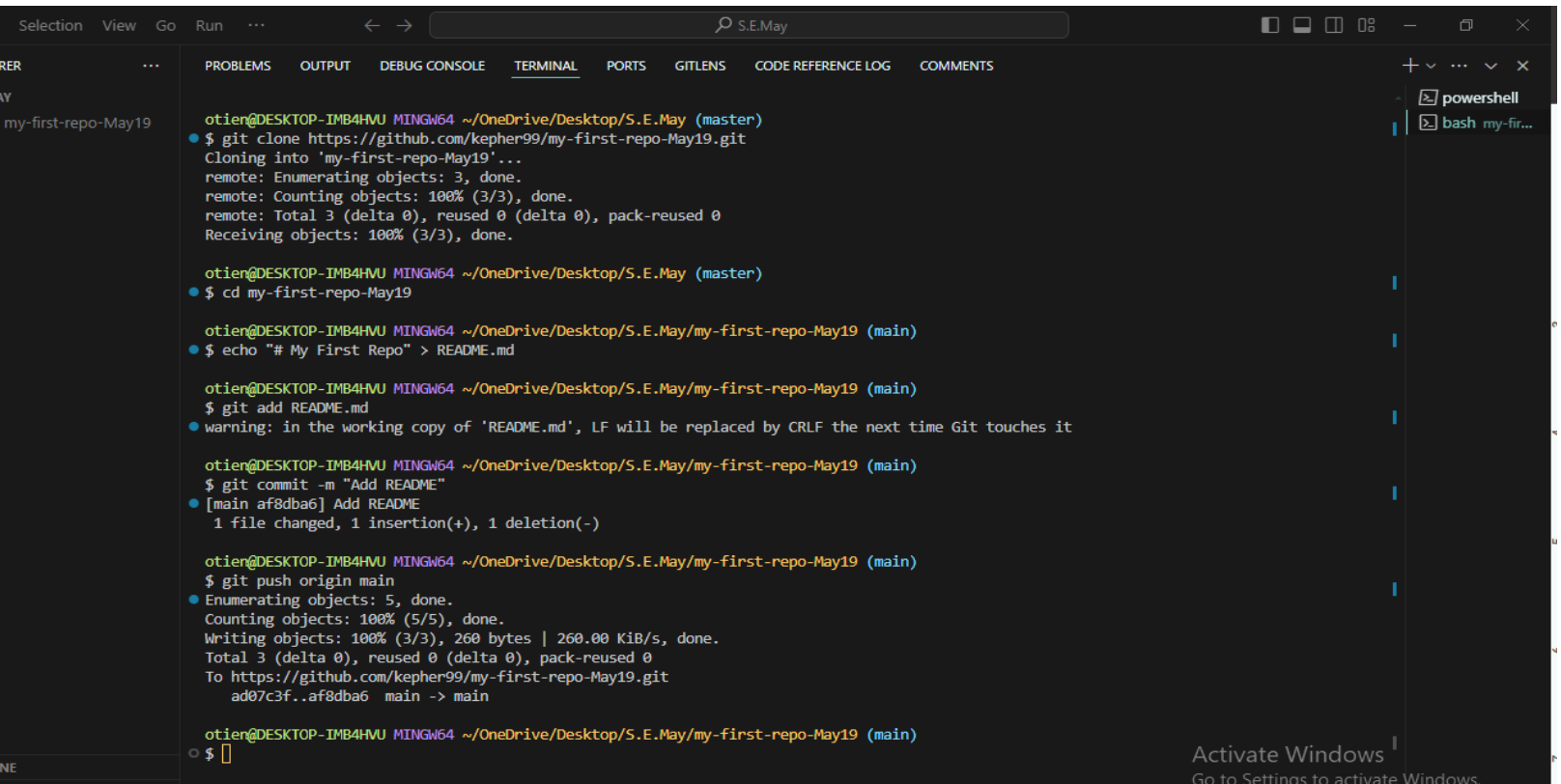
Initialize the repository:

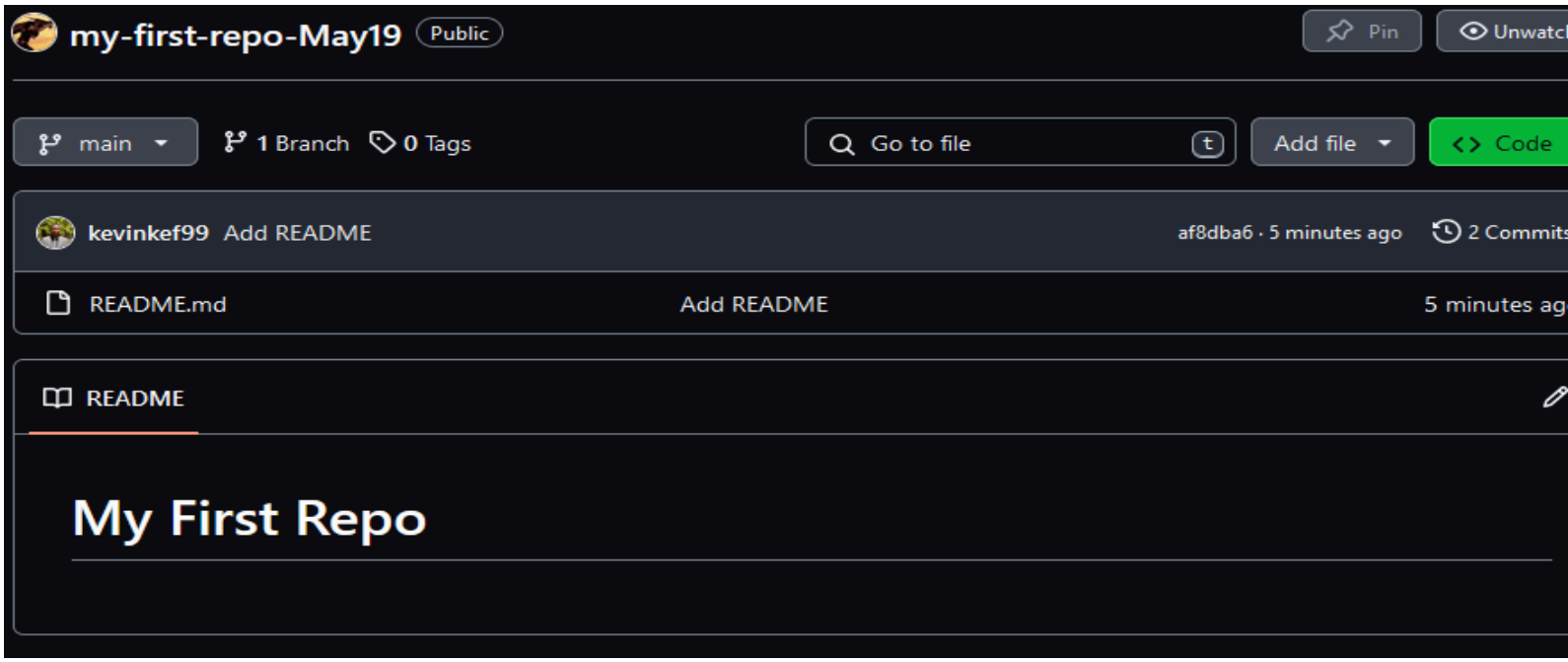
```
git init
```

    -
5. **Make Your First Commit:**
    - Create a sample file, e.g., `README.md`.

Add and commit the file:

```
echo "# MyProject" > README.md
git add README.md
git commit -m "Initial commit"
```





4. Install Necessary Programming Languages and Runtimes

Task: Install Python and its necessary tools.

Step-by-Step Instructions:

1. **Download Python:**
   - Visit the official Python download page: Python Download.
   - Download the installer for Python 3.x.
2. **Install Python:**
   - Run the downloaded installer and ensure you check the option to "Add Python to PATH."
   - Follow the setup wizard to complete the installation.

5. Install Package Managers

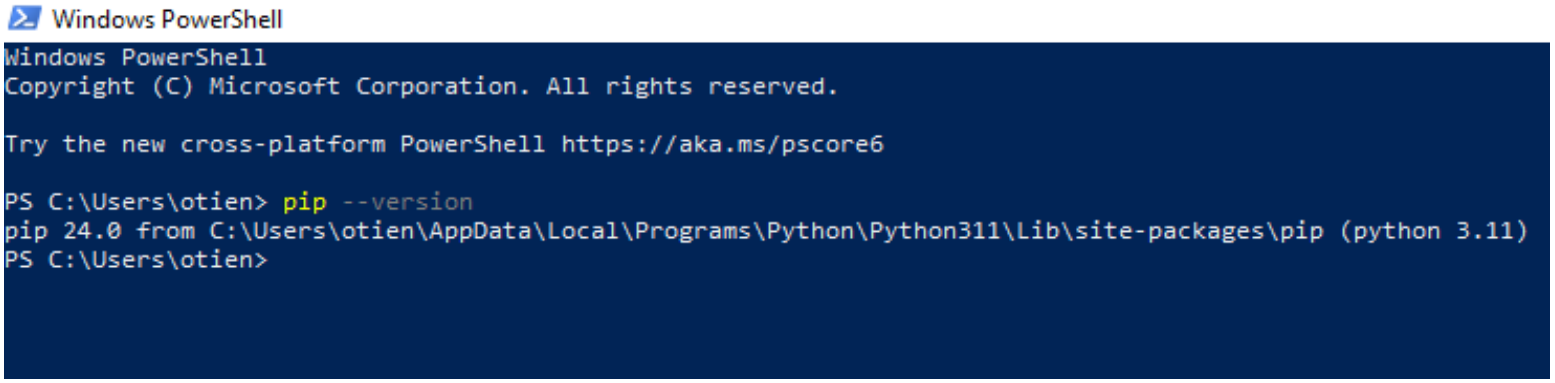
Task: Install pip for Python.

Step-by-Step Instructions:

1. **Verify pip Installation:**

Open Command Prompt and run:

```
pip --version
```



-

Pip is installed by default with Python. If not, install it by running:

```
python -m ensurepip --upgrade
```

-

6. Configure a Database (MySQL)

Task: Download and install MySQL Workbench.

Step-by-Step Instructions:

1. **Download MySQL Workbench:**
   - Visit the official MySQL Workbench download page: MySQL Workbench Download.
   - Select your operating system and download the appropriate installer.

2. **Install MySQL Workbench:**
   - Run the downloaded installer.
   - Follow the setup wizard instructions.
   - Choose the default setup type unless you have specific requirements.
   - Complete the installation process.

3. **Launch and Configure MySQL Workbench:**
   - Open MySQL Workbench after installation.
   - Set up a new MySQL connection:

        - Click on the "+" symbol next to "MySQL Connections."
        - Enter the connection details (e.g., connection name, hostname, port, username, password).
        - Test the connection to ensure it works correctly.
    - Save the connection and start using MySQL Workbench to manage your database

7. Install Dart and Flutter

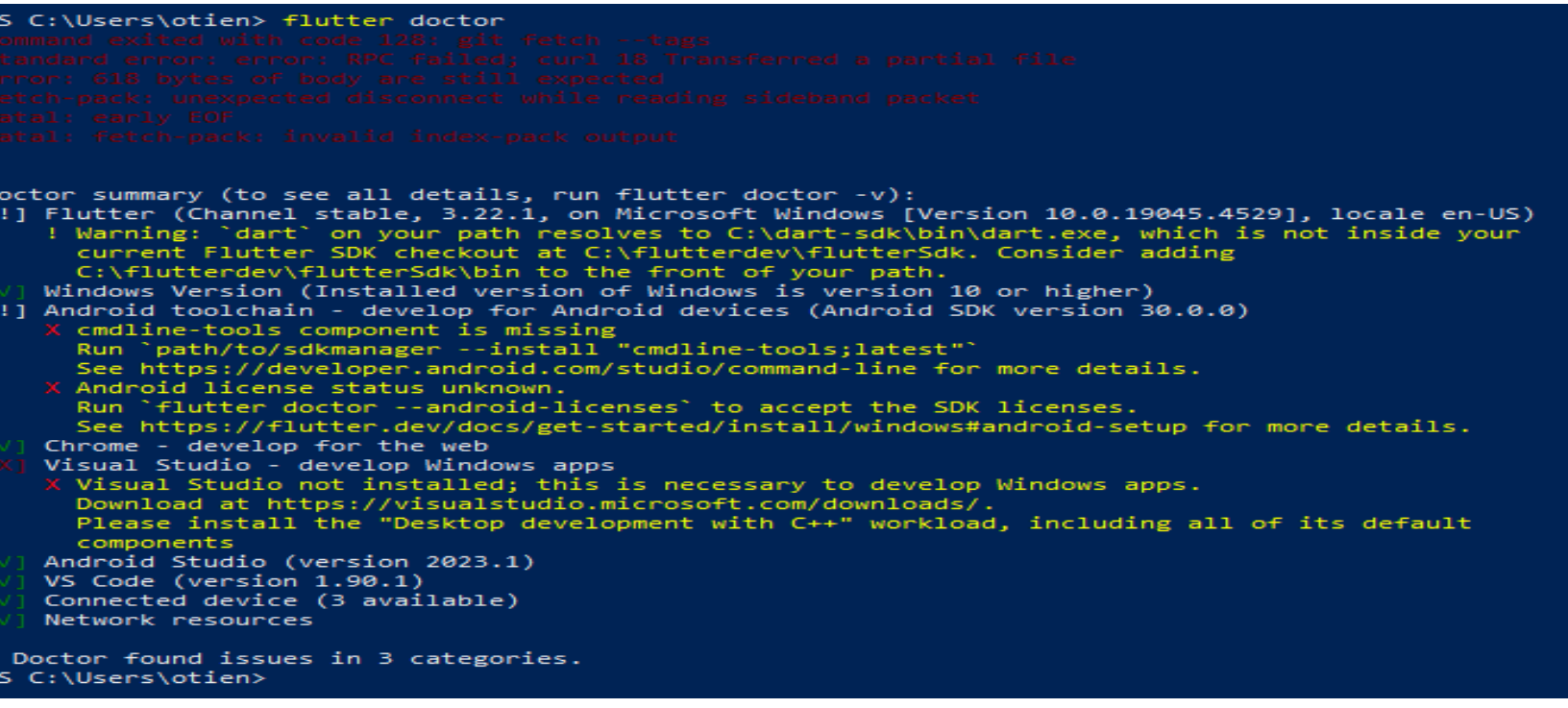
Task: Install Dart and Flutter for developing cross-platform applications.

Step-by-Step Instructions:

1. **Download and Install Flutter:**
    - Visit the official Flutter download page: Flutter Download.
    - Download the Flutter SDK for Windows.
2. **Set Up Flutter:**
    - Extract the downloaded Flutter SDK to a desired location (e.g., `C:\flutter`).
    - Add the Flutter `bin` directory to your system PATH:
        - Search for "Environment Variables" in the Windows search bar.
        - Click on "Edit the system environment variables."
        - Click on the "Environment Variables" button.
        - Under "System variables," select the `Path` variable and click "Edit."
        - Click "New" and add the path to the Flutter `bin` directory (e.g., `C:\flutter\bin`).
3. **Verify Flutter Installation:**

    Open a new Command Prompt window and run:

    ```
    flutter doctor
    ```

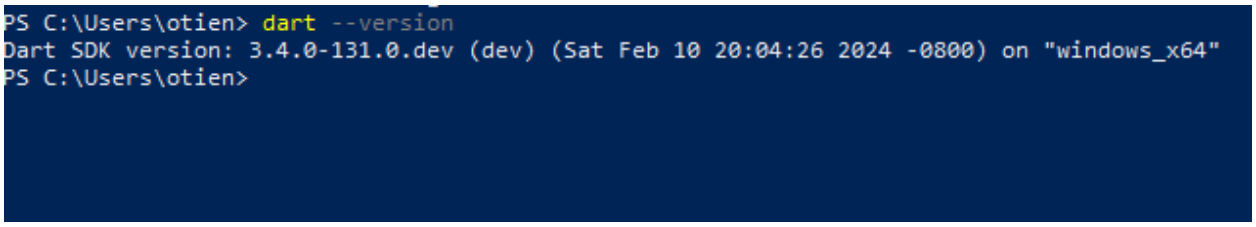


- Follow any additional setup instructions provided by the `flutter doctor` output.

4. **Install Dart:**
   - Dart is included with Flutter, so no separate installation is required.

   Verify Dart installation by running:

   ```
   dart --version
   ```



8. Explore Extensions and Plugins

Task: Enhance VS Code with useful extensions.

Recommended Extensions:

1. **Python:**
   - Install the Python extension for VS Code from the Extensions Marketplace.
2. **GitLens:**
   - Install the GitLens extension to enhance Git capabilities within VS Code.
3. **Prettier:**
   - Install the Prettier extension for code formatting.
4. **Dart and Flutter:**
   - Install the Dart extension from the Extensions Marketplace.
   - Install the Flutter extension from the Extensions Marketplace

Deliverables

1. **Setup Documentation:**
   - Create a document with step-by-step instructions and screenshots.
   - Save the document as `Setup_Documentation.md`.

2. **GitHub Repository:**
    - Create a sample project repository on GitHub.
    - Add the `README.md` file and any necessary configuration files (e.g., `.gitignore`).
    - Link: ihttps://github.com/kepher99/my-first-repo-May19.git

3. **Reflection:**
    - Write a reflection on the challenges faced and strategies to overcome them.

Common Challenges Faced During Setup

1. **Compatibility Issues with Operating System:**

- **Challenge:** Some software and tools may not be fully compatible with the latest version of the operating system, leading to installation errors or malfunctioning features.

  Solution: Research and ensure that the software versions you are installing are compatible with your OS. Use community forums and official documentation for troubleshooting steps. If necessary, use an alternative version of the software that is known to be stable with your OS.

2. **Environment Variables Configuration:**

   Challenge: Incorrectly setting up environment variables can lead to issues with running installed software and tools.

   Solution: Follow detailed instructions for setting environment variables carefully. Double-check paths for accuracy. Utilize online resources and community help if there are issues. Testing the setup with simple commands can ensure that the variables are configured correctly.

3. **Version Control Setup:**

- **Challenge:** Initial setup of Git and connecting to GitHub can be confusing for beginners, especially when dealing with SSH keys and repository permissions.
- **Solution:** Follow step-by-step guides for Git and GitHub setup. Start with HTTPS connections if SSH setup is challenging. Use GitHub Desktop for a more user-friendly interface. Practice basic Git commands to build confidence.

4. **Database Configuration:**

- **Challenge:** Configuring databases like MySQL Workbench can be complex, involving multiple steps and potential configuration issues.
- **Solution:** Follow detailed installation guides specific to your operating system. Use the default settings initially and then customize as needed. Refer to official MySQL documentation and community forums for troubleshooting.