

Step 1: Choosing the Operating System

- **Operating System Selection**: The choice was made to use Windows 11 due to its modern design, improved performance, and enhanced security features. Windows 11 offers a streamlined experience for developers, with better integration for development tools and services.

Step 2: Downloading and Installing Windows 11

- **Downloading Windows 11**: The Windows 11 ISO file was downloaded from the official Microsoft website
- **Preparation**: Before starting the installation, it was ensured that the PC met the minimum system requirements for Windows 11, including having a 64-bit processor and sufficient storage space
- **Installation**: The ISO file was mounted to create a virtual bootable disc, and the setup.exe file was double-clicked to initiate the installation process

Step 3: Post-Installation Configuration

- **Language, Time, and Keyboard Preferences**: During the initial setup, language, time, and keyboard preferences were selected according to personal preference [1].
- **Update Check**: After installation, Windows Update was checked to ensure the system was up-to-date with the latest patches and improvements [1].

Step 5: Troubleshooting and Customization

- **Troubleshooting**: During the setup process, common issues such as driver conflicts and software compatibility were addressed by visiting the manufacturer's website for updated drivers and consulting online forums for solutions.
- **Customization**: The desktop environment was customized with favorite wallpapers and applications pinned to the taskbar for quick access. Privacy settings were adjusted to balance convenience and security.

Step 6: Finalizing the Developer Environment

- **Backup and Recovery Plan**: A backup plan was established using Windows Backup and Restore, ensuring critical data could be recovered in case of failure.
- **Security Enhancements**: Additional security measures were implemented, including enabling BitLocker encryption and setting up a firewall to protect against unauthorized access.

Step 1: Selecting Visual Studio Code

Visual Studio Code was chosen due to its extensive support for multiple programming languages, customizable interface, and a vast library of extensions that enhance functionality and productivity.

Step 2: Downloading Visual Studio Code

Visited the official Visual Studio Code website: <https://code.visualstudio.com/>

Click on the "Download" button for the version compatible with Windows 11

Follow the prompts to download the installer.

Step 3: Installing Visual Studio Code

Run the downloaded installer.

Accept the license agreement and click "Next".

Choose the installation directory if prompted, or accept the default location.

Selected the features to install. For a basic setup, selected "Editor", "Source Control", and "Debug".

Click "Install" and wait for the installation to complete.

Step 4: Configuring Visual Studio Code

Extensions

Open VS Code.

Go to the Extensions view by clicking on the square icon on the sidebar or pressing Ctrl+Shift+X.

Search for and install essential extensions such as:

Python extension for Python development.

GitLens for enhanced Git capabilities within VS Code.

Customize settings for each extension through the Extensions view or by editing the settings.json file (File > Preferences > Settings > Extensions > Edit in settings.json).

Keyboard Shortcuts

Access keyboard shortcuts by going to File > Preferences > Keyboard Shortcuts.

Customize shortcuts to optimize your workflow.

Step 5: Troubleshooting

Installation Issues

If the installation fails, ensure your system meets the minimum requirements and try downloading the installer again.

Extension Compatibility

Some extensions may require additional software or dependencies. Ensure all prerequisites are installed before attempting to install these extensions.

Performance Issues

If VS Code runs slowly, consider disabling unnecessary extensions, clearing the cache (Help > Toggle Developer Tools > Network tab), or restarting the application.

To set up version control system (VCS) on your local machine, install Git, create a GitHub account, initialize a Git repository for your project, and make your first commit, follow these steps:

Step 1: Install Git

1. ****Windows****: Downloaded and install Git from [Git for Windows](https://gitforwindows.org/). During installation, ensured to check the option to "Adjust your PATH environment" so that Git commands can be run from any command prompt

Step 2: Configure Git

After installing Git, configured username and email address, which will be associated with my commits:

```
```bash
```

```
git config --global user.name "ThaboMokoni"
```

```
git config --global user.email "thabomokoni7@gmail.com"
```

#### ### Step 3: Create a GitHub Account/ sign in

1. Go to [GitHub](https://github.com).

2. Click on "Sign in".

#### ### Step 4: Initialize a Git Repository

Navigated to my project directory in the terminal or command prompt and initialize a new Git repository:

```
` `` bash
```

```
cd path/to/your/project
```

```
git init
```

### ### Step 5: Make Your First Commit

Created a file in my project directory. Then, stage and committed those changes:

```
` `` bash
```

```
Add all files to staging area
```

```
git add.
```

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

### ### Step 6: Link Your Local Repository to GitHub

1. On GitHub, click the "+" icon in the top-right corner and selected "New repository".
2. Named the repository, added a description (optional), chose it to be public, and click "Create repository".
3. In the terminal, linked local repository to the remote GitHub repository:

```
` `` bash
```

```
git remote add origin https://github.com/username/repository-name.git
```

```
` ``
```

4. Push commits to GitHub:

```
` `` bash
```

```
git branch -M main # Rename the default branch to main if it was master
```

```
git push -u origin main
```

```
` ``
```

Download Python Installer:

Visit the official Python website at <http://www.python.org>

Scroll down to the Downloads section and download the latest Python installer for Windows.

Install Python:

Run the downloaded .exe file.

Chose the Customize installation option.

Ensured that the following options are checked:

Add Python to PATH

Install launcher for all users (recommended)

Check the box to install the pip package manager.

Complete the installation process.

Verify Installation:

Open Command Prompt and type `python --version`. You should see the installed Python version displayed.

Type `pip --version` to verify that pip is installed.

Data base MySQL installation

Download the Installer: Go to the MySQL downloads page at <https://dev.mysql.com/downloads/windows/installer/5.7.html>. Since MySQL 8.0 is the last series supported by MySQL Installer, you should select MySQL 5.7 if you prefer using the installer. Choose between the `mysql-installer-web-community` file if you have an online connection during installation, or the `mysql-installer-community` file if you don't have an online connection.

Installation Process:

Run the downloaded installer. Note that the MySQL Installer is 32-bit but can install both 32-bit and 64-bit binaries.

Follow the wizard-based installation process. The installer will guide you through selecting components to install, such as MySQL Server, MySQL Router, MySQL Shell, MySQL Workbench, sample databases, and documentation.

Configuration:

During the installation, you'll be prompted to set up a root password and other configuration options. Make sure to remember the root password as you'll need it to manage your MySQL server.

You can also customize the installation path and other settings according to your preferences.

#### Post-installation Steps:

After installation, you might want to run the MySQL Server and MySQL Workbench to start managing your databases.

Explore the documentation available on the MySQL website for further guidance on configuring and optimizing your MySQL server.