Windows 11 Installation Process

1. Check System Requirements

- **Processor**: 1 GHz or faster with at least 2 cores on a compatible 64-bit processor.
- **RAM**: 4 GB or more.
- **Storage**: 64 GB or larger storage device.
- **System Firmware**: UEFI, Secure Boot capable.
- **TPM**: Trusted Platform Module (TPM) version 2.0.
- **Graphics Card**: DirectX 12 compatible graphics / WDDM 2.x.
- **Display**: >9” with HD Resolution (720p).
- **Internet Connection**: Internet connectivity is necessary to perform updates and to download and take advantage of some features.

2. Backup Important Data

- Ensure you backup your important files and data to an external drive or cloud storage.

3. Download Windows 11 Installation Media

- Visit the Microsoft Windows 11 Download Page (https://www.microsoft.com/software-download/windows11)
- Click on "Download now" under the Windows 11 Installation Assistant.

4. Run the Installation Assistant

- Open the downloaded file (Windows11InstallationAssistant.exe).
- Follow the on-screen prompts:
  - Accept the Licence Terms.
  - The assistant will check your PC for compatibility.
  - If your PC is compatible, click "Next" to begin the installation.
  - The tool will download Windows 11 and start the installation process.
  - Your PC will restart several times during the installation process.

5. Complete Installation

- After the final restart, you'll see the Windows 11 setup screen.
- Follow the on-screen prompts to:
  - Choose your region and keyboard layout.
  - Connect to a network.
  - Set up your account (Microsoft account recommended).
  - Configure privacy settings.
  - Customise your preferences (e.g., language, time zone).

6. Post-Installation

- Check for updates:
  - Go to **Settings** -> **Windows Update** -> **Check for updates**.

- Install necessary drivers and software:
  - Visit your PC manufacturer's website for the latest drivers.
  - Reinstall essential applications.

Git Installation and Configuration on Windows 11

1. Download Git

- Visit the Git official website (https://git-scm.com/).
- Click on the "Download for Windows" button to download the installer.

2. Install Git

- Locate the downloaded file (typically `Git-<version>-64-bit.exe`) and double-click to run the installer.
- **Setup Wizard Steps**:
  1. **Welcome Screen**: Click "Next".
  2. **Select Destination Location**: Choose the default or specify a different location, then click "Next".
  3. **Select Components**: Leave the default options checked, then click "Next".
  4. **Select Start Menu Folder**: Click "Next" to use the default folder.
  5. **Choosing the default editor used by Git**: Select your preferred text editor (e.g., Visual Studio Code), then click "Next".
  6. **Adjusting your PATH environment**: Choose "Git from the command line and also from 3rd-party software", then click "Next".
  7. **Choosing HTTPS transport backend**: Select "Use the OpenSSL library", then click "Next".
  8. **Configuring the line ending conversions**: Select "Checkout Windows-style, commit Unix-style line endings", then click "Next".
  9. **Configuring the terminal emulator to use with Git Bash**: Choose "Use MinTTY (the default terminal of MSYS2)", then click "Next".
  10. **Choosing the default behavior of `git pull`**: Leave the default option selected, then click "Next".
  11. **Choose a credential helper**: Select "Git Credential Manager", then click "Next".
  12. **Configuring extra options**: Leave the default options checked, then click "Next".
  13. **Configuring experimental options**: Uncheck any experimental options unless you want to try them, then click "Install".
- Wait for the installation to complete, then click "Finish".

3. Verify Installation

- Open Command Prompt or PowerShell.
- Type `git --version` and press Enter to verify Git is installed correctly.

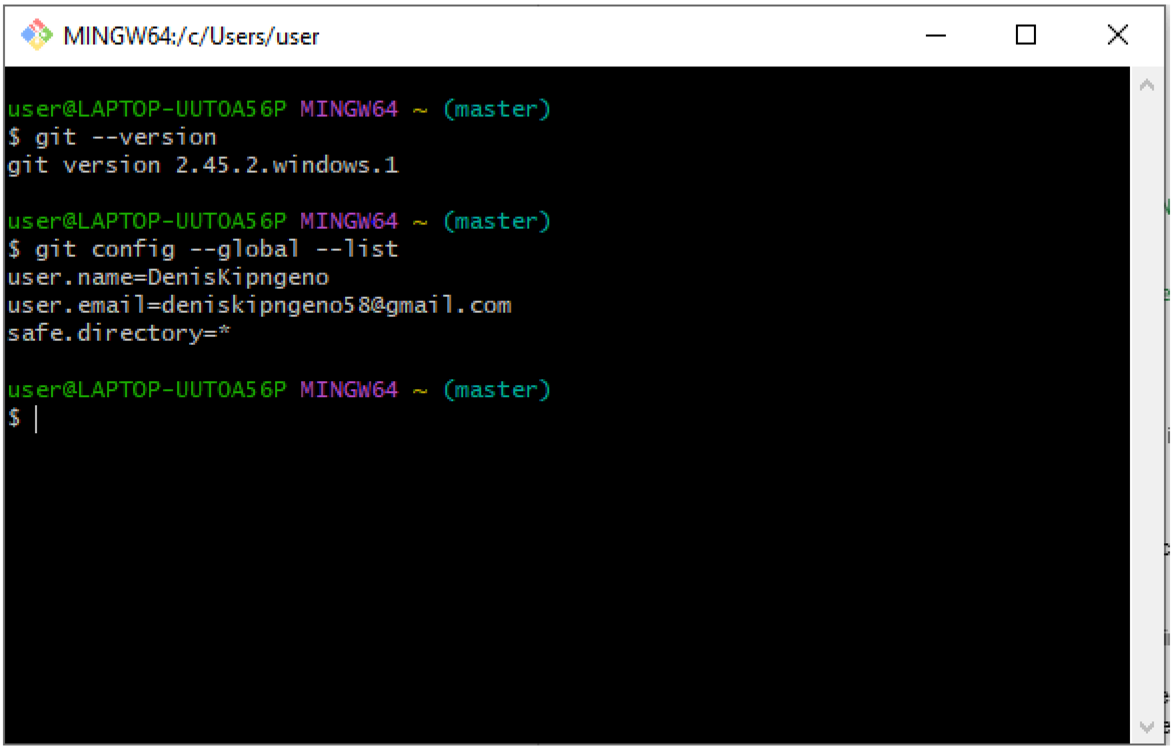
4. Configure Git

- **Open Git Bash** (installed with Git):

Set your username:
`git config --global user.name "Your Name"`

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

Verify the configuration:
`git config --list`



```
user@LAPTOP-UUTOA56P MINGW64 ~ (master)
$ git --version
git version 2.45.2.windows.1

user@LAPTOP-UUTOA56P MINGW64 ~ (master)
$ git config --global --list
user.name=DenisKipngeno
user.email=deniskipngeno58@gmail.com
safe.directory=*

user@LAPTOP-UUTOA56P MINGW64 ~ (master)
$ |
```

5. Add Git to PATH (If not already done during installation)

- **Manually Adding Git to PATH**:
  1. **Open System Properties**:
     - Press `Windows + X`, select "System", then click "Advanced system settings".
  2. **Environment Variables**:
     - In the System Properties window, click on "Environment Variables".
  3. **Edit Path Variable**:
     - In the Environment Variables window, find the "Path" variable under "System variables" and select it, then click "Edit".
  4. **Add Git to Path**:

In the Edit Environment Variable window, click "New" and add the path to your Git `bin` and `cmd` directories. Typically:
plaintext

Copy code

```
C:\Program Files\Git\bin

C:\Program Files\Git\cmd
```

5. **Save and Apply**:
    - Click "OK" to close each window and apply the changes

Python 3.12.3 Installation and Configuration on Windows 11

1. Download Python 3.12.3

- Visit the official Python website(https://www.python.org/downloads/).
- Click on the "Download Python 3.12.3" button.

2. Install Python 3.12.3

- Locate the downloaded file (typically `python-3.12.3-amd64.exe`) and double-click to run the installer.
- **Setup Wizard Steps**:
    1. **Welcome Screen**:
        - Check the box "Add Python 3.12 to PATH".
        - Click "Customise installation".
    2. **Optional Features**:
        - Ensure all options are checked (pip, tcl/tk and IDLE, Documentation, etc.).
        - Click "Next".
    3. **Advanced Options**:
        - Check "Install for all users".
        - Ensure "Add Python to environment variables" is checked.
        - Optionally, customise the installation location (default is usually fine).
        - Click "Install".
    4. **Wait for Installation**:
        - The installer will download and install Python.
        - Click "Close" once the installation is complete.

3. Verify Installation

- Open Command Prompt or PowerShell.
- Type `python --version` and press Enter to verify Python is installed correctly.
- Type `pip --version` to ensure pip, the Python package installer, is also installed.

4. Configure Python (if not added to PATH during installation)

- **Manually Adding Python to PATH**:

1. **Open System Properties**:
    - Press `Windows + X`, select "System", then click "Advanced system settings".
2. **Environment Variables**:
    - In the System Properties window, click on "Environment Variables".
3. **Edit Path Variable**:
    - In the Environment Variables window, find the "Path" variable under "System variables" and select it, then click "Edit".
4. **Add Python to Path**:

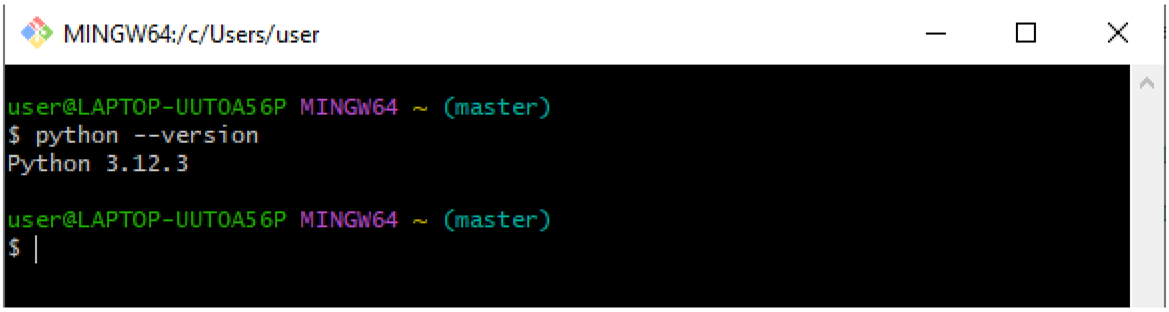
In the Edit Environment Variable window, click "New" and add the paths to the Python `Scripts` and installation directories. Typically:
plaintext
Copy code

```
C:\Users\<YourUsername>\AppData\Local\Programs\Python\Python312

C:\Users\<YourUsername>\AppData\Local\Programs\Python\Python312\Scri
pts
```

-

5. **Save and Apply**:
    - Click "OK" to close each window and apply the changes.



```
user@LAPTOP-UUTOA56P MINGW64 ~ (master)
$ python --version
Python 3.12.3

user@LAPTOP-UUTOA56P MINGW64 ~ (master)
$ |
```

Dart SDK 3.4.3 Installation and Configuration on Windows 11

1. Download Dart SDK

- Visit the Dart SDK download page.
- Under the "Get the Dart SDK" section, click on the "Download" button for Windows.
- Select the latest stable release, which is `Dart SDK version: 3.4.3`.

2. Install Dart SDK

- Locate the downloaded file (usually a ZIP archive, e.g., `dartsdk-windows-x64-release.zip`) and extract its contents.

Extract the contents to a directory of your choice, for example:
plaintext

Copy code

```
C:\dart-sdk
```

-

3. Configure Dart SDK to PATH

- **Open System Properties**:
  - Press `Windows + X`, select "System", then click "Advanced system settings".
- **Environment Variables**:
  - In the System Properties window, click on "Environment Variables".
- **Edit Path Variable**:
  - In the Environment Variables window, find the "Path" variable under "System variables" and select it, then click "Edit".
- **Add Dart SDK to Path**:

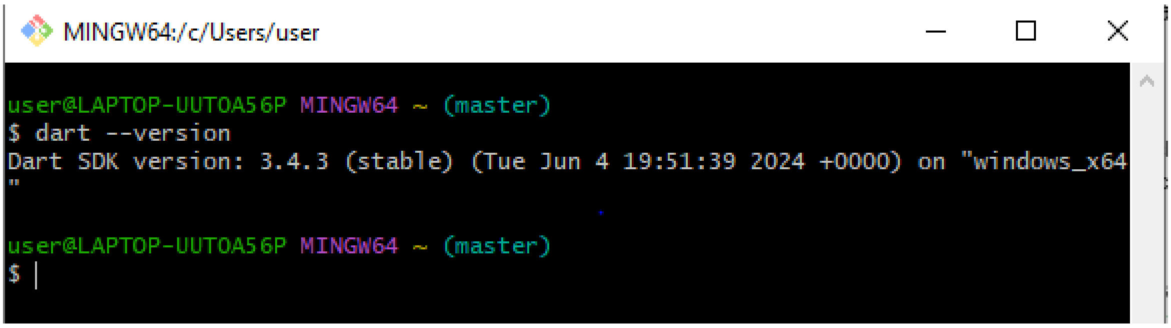
In the Edit Environment Variable window, click "New" and add the path to the `bin` directory of the Dart SDK. For example:
plaintext
Copy code

```
C:\dart-sdk\bin
```

  -
- **Save and Apply**:
  - Click "OK" to close each window and apply the changes.

4. Verify Installation

- Open Command Prompt or PowerShell.
- Type `dart --version` and press Enter to verify Dart is installed correctly.



```
user@LAPTOP-UUTOA56P MINGW64 ~ (master)
$ dart --version
Dart SDK version: 3.4.3 (stable) (Tue Jun 4 19:51:39 2024 +0000) on "windows_x64
"

user@LAPTOP-UUTOA56P MINGW64 ~ (master)
$ |
```

Flutter 3.22.2 Installation and Configuration on Windows 11

1. Download Flutter SDK

- Visit the Flutter SDK download page.
- Under the "Get the Flutter SDK" section, click on the "Download" button for Windows to download the latest stable release, which is `Flutter version: 3.22.2`.

2. Extract Flutter SDK

- Locate the downloaded file (usually a ZIP archive, e.g., `flutter_windows_3.22.2-stable.zip`) and extract its contents.

Extract the contents to a directory of your choice, for example:
plaintext
Copy code

```
C:\flutter
```

3. Configure Flutter SDK to PATH

- **Open System Properties**:
  - Press `Windows + X`, select "System", then click "Advanced system settings".
- **Environment Variables**:
  - In the System Properties window, click on "Environment Variables".
- **Edit Path Variable**:
  - In the Environment Variables window, find the "Path" variable under "System variables" and select it, then click "Edit".
- **Add Flutter SDK to Path**:

In the Edit Environment Variable window, click "New" and add the path to the `bin` directory of the Flutter SDK. For example:
plaintext
Copy code

```
C:\flutter\bin
```

- **Save and Apply**:
  - Click "OK" to close each window and apply the changes.

4. Verify Installation

- Open Command Prompt or PowerShell.

Run the following command to ensure Flutter is installed correctly:
bash
Copy code

```
flutter --version
```

-

Run the following command to verify the setup:
bash
Copy code

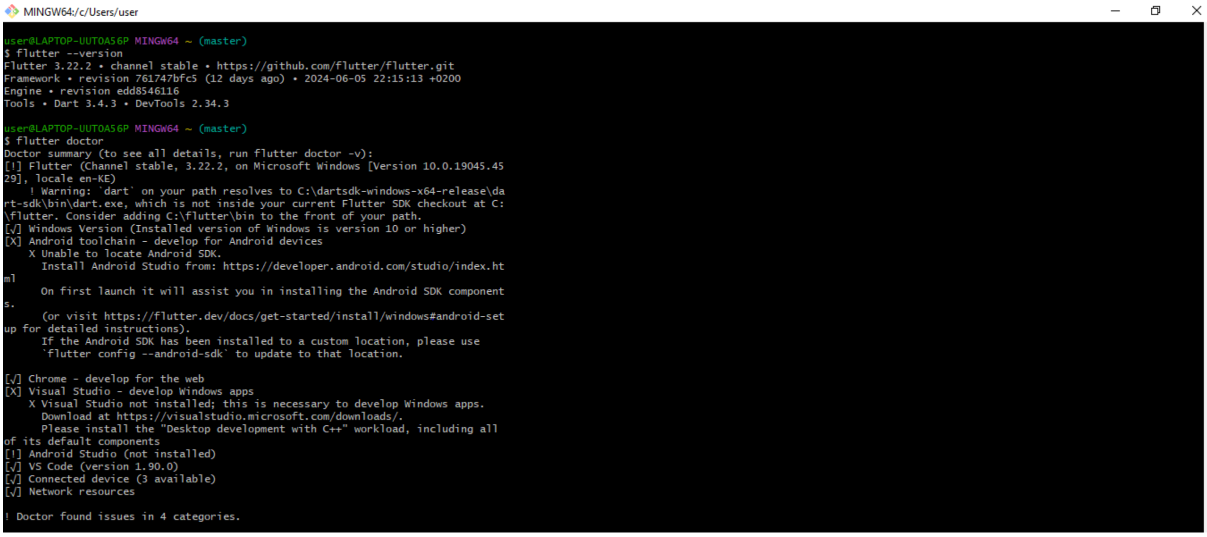
```
flutter doctor
```

-

- Follow any additional instructions provided by `flutter doctor` to complete the setup (e.g., installing additional dependencies).

5. Optional: Install Additional Tools

- **Android Studio**:
  - Install Android Studio if you plan to develop for Android.
  - Follow the [Android Studio installation guide](https://developer.android.com/studio) (https://developer.android.com/studio).
- **Visual Studio Code**:
  - Install Visual Studio Code if you prefer it as your IDE.
  - Follow the [Visual Studio Code installation guide](https://code.visualstudio.com/).(https://code.visualstudio.com/)
  - Install Flutter and Dart extensions from the Extensions Marketplace.



```
user@LAPTOP-UUTOA56P MINGW64 ~ (master)
$ flutter --version
Flutter 3.22.2 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 761747bfc5 (12 days ago) • 2024-06-05 22:15:13 +0200
Engine • revision edd8546116
Tools • Dart 3.4.3 • DevTools 2.34.3

user@LAPTOP-UUTOA56P MINGW64 ~ (master)
$ flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.22.2, on Microsoft Windows [Version 10.0.19045.45
29], locale en-KE)
    ! Warning: `dart` on your path resolves to C:\dartsdk-windows-x64-release\da
rt-sdk\bin\dart.exe, which is not inside your current Flutter SDK checkout at C:
\flutter. Consider adding C:\flutter\bin to the front of your path.
[√] Windows Version (Installed version of Windows is version 10 or higher)
[X] Android toolchain - develop for Android devices
    X Unable to locate Android SDK.
      Install Android Studio from: https://developer.android.com/studio/index.ht
ml
      On first launch it will assist you in installing the Android SDK component
s.
      (or visit https://flutter.dev/docs/get-started/install/windows#android-set
up for detailed instructions).
      If the Android SDK has been installed to a custom location, please use
      `flutter config --android-sdk` to update to that location.

[√] Chrome - develop for the web
[X] Visual Studio - develop Windows apps
    X Visual Studio not installed; this is necessary to develop Windows apps.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all
of its default components
[!] Android Studio (not installed)
[√] VS Code (version 1.90.0)
[√] Connected device (3 available)
[√] Network resources

! Doctor found issues in 4 categories.
```

Visual Studio Code Installation and Configuration on Windows 11

1. Download Visual Studio Code

- Visit the [Visual Studio Code website](https://code.visualstudio.com/)(https://code.visualstudio.com/).
- Click on the "Download for Windows" button to download the latest version of Visual Studio Code.

2. Install Visual Studio Code

- Locate the downloaded file (typically `VSCodeUserSetup-x64-<version>.exe`) and double-click to run the installer.
- **Setup Wizard Steps**:
  1. **Welcome Screen**: Click "Next".
  2. **License Agreement**: Accept the license agreement and click "Next".
  3. **Select Destination Location**: Choose the default or specify a different location, then click "Next".
  4. **Select Start Menu Folder**: Click "Next" to use the default folder.

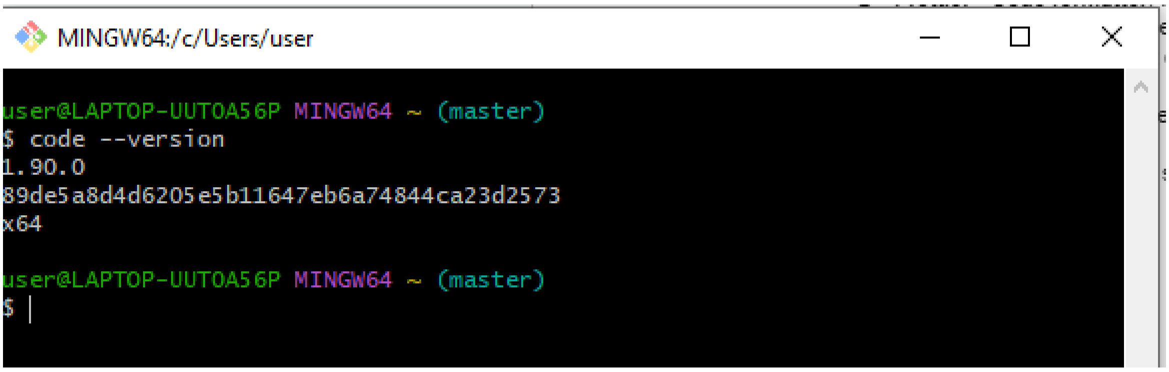
5. **Select Additional Tasks**:
    - Check the box "Add to PATH (requires shell restart)".
    - Optionally check other boxes like "Create a desktop icon" and "Add to context menu".
    - Click "Next".
6. **Ready to Install**: Click "Install".
7. **Complete Installation**: Click "Finish" to launch Visual Studio Code.

3. Verify Installation

- Open Command Prompt or PowerShell.
- Type `code --version` and press Enter to verify Visual Studio Code is installed correctly and accessible from the PATH.

4. Configure Visual Studio Code

- **Launch Visual Studio Code**:
    - Open Visual Studio Code from the Start Menu or desktop shortcut.
- **Install Recommended Extensions**:
    - Open the Extensions view by clicking on the Extensions icon in the Activity Bar on the side of the window.
    - Search for and install essential extensions such as:
        - **Python**: For Python development.
        - **Prettier - Code formatter**: For code formatting.
        - **ESLint**: For JavaScript/TypeScript linting.
        - **GitLens**: For enhanced Git capabilities.
- **Customize Settings**:
    - Open the settings by clicking on the gear icon in the lower-left corner and selecting "Settings".
    - Adjust settings to your preference, such as theme, font size, and keybindings.



```
user@LAPTOP-UUTOA56P MINGW64 ~ (master)
$ code --version
1.90.0
89de5a8d4d6205e5b11647eb6a74844ca23d2573
x64

user@LAPTOP-UUTOA56P MINGW64 ~ (master)
$ |
```

.

MySQL Installation and Configuration on Windows 11

1. Download MySQL Installer

- Visit the [MySQL Downloads page](#).

- Click on the "Download" button for the MySQL Installer for Windows (typically `mysql-installer-web-community`).

2. Install MySQL

- Locate the downloaded file (`mysql-installer-web-community-<version>.exe`) and double-click to run the installer.
- **Setup Wizard Steps**:
    1. **Welcome Screen**: Click "Next".
    2. **Choosing a Setup Type**:
        - Select "Developer Default" for a full MySQL installation, or customize based on your needs.
        - Click "Next".
    3. **Check Requirements**: The installer will check for any missing requirements. Click "Execute" to install any required software, then click "Next".
    4. **Installation**: Click "Execute" to begin installing MySQL products. Once completed, click "Next".
    5. **Product Configuration**: Click "Next" to start the configuration process.

3. Configure MySQL Server

- **Type and Networking**:
    - Choose "Standalone MySQL Server / Classic MySQL Replication".
    - Select the default TCP/IP port (3306) and ensure "Open Windows Firewall port for network access" is checked.
    - Click "Next".
- **Authentication Method**:
    - Choose "Use Strong Password Encryption for Authentication (RECOMMENDED)".
    - Click "Next".
- **Accounts and Roles**:
    - Set a strong password for the root user and create any additional user accounts as needed.
    - Click "Next".
- **Windows Service**:
    - Ensure "Configure MySQL Server as a Windows Service" is checked.
    - Set the service name (default is MySQL80) and select "Start the MySQL Server at System Startup".
    - Click "Next".
- **Apply Configuration**: Click "Execute" to apply the configuration settings. Once completed, click "Finish".

4. Add MySQL to PATH

- **Open System Properties**:
    - Press `Windows + X`, select "System", then click "Advanced system settings".

- **Environment Variables**:
  - In the System Properties window, click on "Environment Variables".
- **Edit Path Variable**:
  - In the Environment Variables window, find the "Path" variable under "System variables" and select it, then click "Edit".
- **Add MySQL to Path**:

In the Edit Environment Variable window, click "New" and add the path to the MySQL `bin` directory. For example:
plaintext
Copy code

```
C:\Program Files\MySQL\MySQL Server 8.0\bin
```

  ○
- **Save and Apply**:
  - Click "OK" to close each window and apply the changes.

5. Verify Installation

- Open Command Prompt or PowerShell.
- Type `mysql --version` and press Enter to verify MySQL is installed correctly.

Optionally, log into MySQL using the command:
bash
Copy code

```
mysql -u root -p
```

- Enter the root password set during configuration to access the MySQL shell.