Name: Mark Anthony R. Alegre Yr & Sec: BSCpE-2B

Laboratory Activity No. 1:

Topic: **Introduction to Software Design, History, and Overview**

Title: *Setting Up the Development Environment for Django Project*

Introduction: This activity will guide you through the process of setting up your development environment to start building the Library Management System (LMS) in Django. The process involves installing necessary software, setting up Python and Django, and verifying the installation.

Objectives:

- Install Python and Django on your system.
- Create a virtual environment to manage dependencies.
- Verify the installation by running a simple Django project.

Theory and Detailed Discussion: To develop the Library Management System, we will use the Django framework. Django is a high-level Python web framework that allows developers to create robust web applications quickly and efficiently. Before we can start developing, we need to set up the development environment.

Materials, Software, and Libraries:

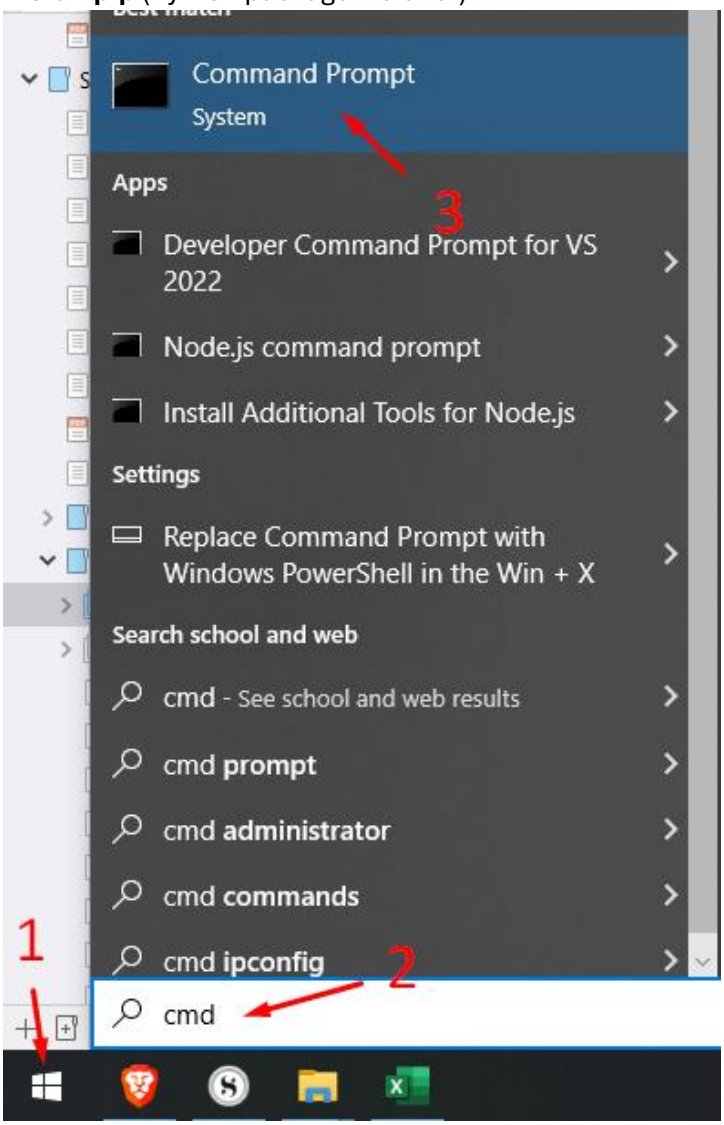
- **Python** (version 3.8 or above)
- **Django** (version 4.0 or above)
- **pip** (Python package manager)
- **Text Editor** (Visual Studio Code or PyCharm)
- **Database** (SQLite – comes with Django by default)

Time Frame: 1 Hour

Procedure:

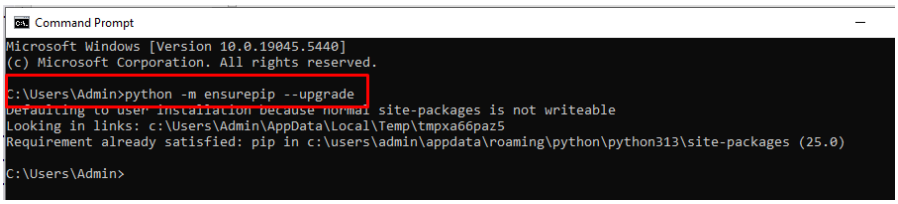
1. **Install Python**:
    - Go to python.org and download the latest version of Python.
    - Install Python by following the installation instructions for your operating system.

2. **Install pip** (Python package installer):



    - type the following command:

python -m ensurepip --upgrade



3. Create folder using your family name as the file name and Open in Visual Studio Code.

In desktop create 

(family_name) folder

Open the folder

Right Click on the folder white space then Open with Code

Or

Open Visual Studio Code

> File > Open Folder >

Select your folder

Or

In Terminal

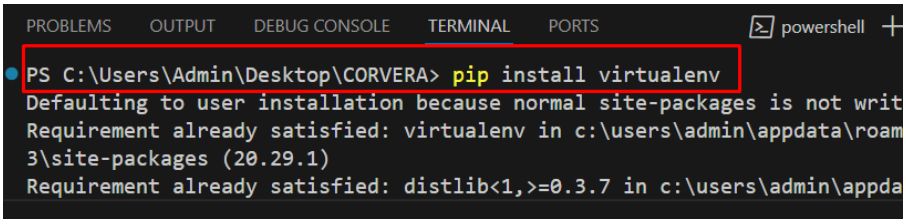
Change directory to the folder then type

Code .

3. **Install Virtual Environment**:

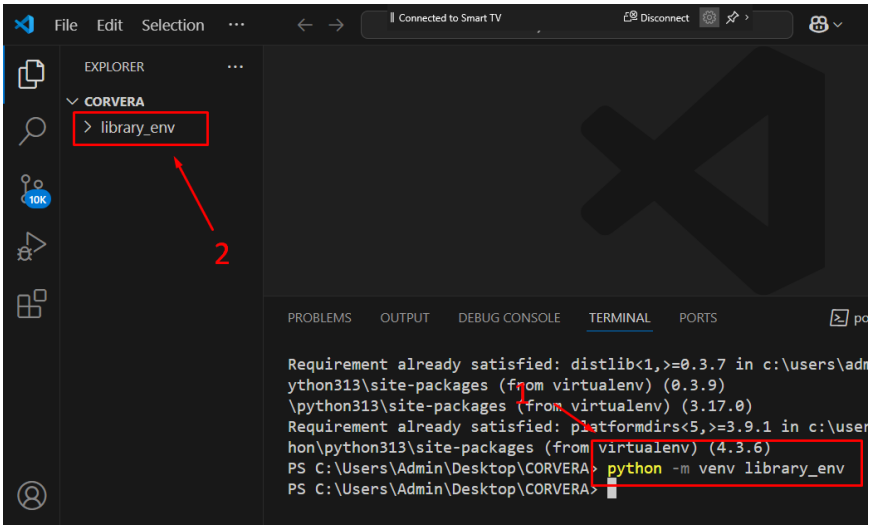
- Create a virtual environment for our project to avoid conflicts with global packages.
- If the terminal is not yet open. Click the View>Terminal in VSCode then type

pip install virtualenv



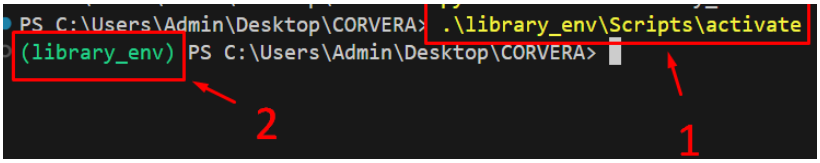
- Create a new virtual environment:

python -m venv library_env



- Activate the virtual environment:
- On Windows:

.\library_env\Scripts\activate

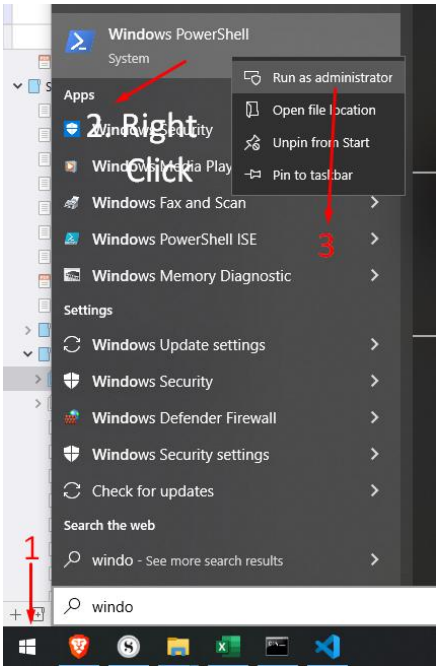


- On Mac/Linux:

source library_env/bin/activate

If **error** in virtual environment activation

Open windows Powershell by running as Administrator



Type the following code:

PS C:\Windows\system32> get-executionpolicy

Restricted

If restricted type the code:

PS C:\Windows\system32> set-executionpolicy remotesigned

Then type again the

.\library_env\Scripts\activate

1. **Install Django**:
    - After activating the virtual environment, install Django by running:

pip install django

```
(library_env) PS C:\Users\Admin\Desktop\CORVERA> pip install django
Collecting django
  Using cached Django-5.1.5-py3-none-any.whl.metadata (4.2 kB)
Collecting asgiref<4,>=3.8.1 (from django)
  Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
```

1

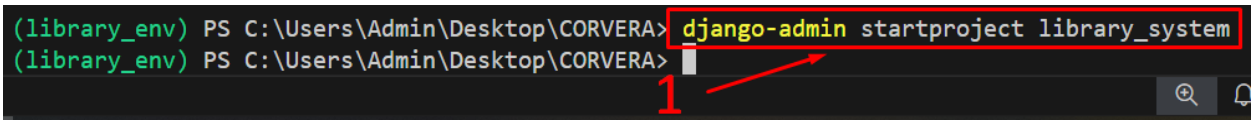
2. **Verify the Django Installation**:
    - Run the following command to verify if Django is installed:

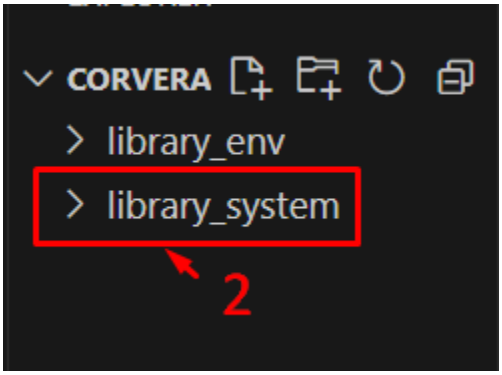
django-admin --version

```
(library_env) PS C:\Users\Admin\Desktop\CORVERA> django-admin --version
5.1.5
(library_env) PS C:\Users\Admin\Desktop\CORVERA>
```

3. **Create a New Django Project**:
    - Create a new Django project called "library_system":

django-admin startproject library_system

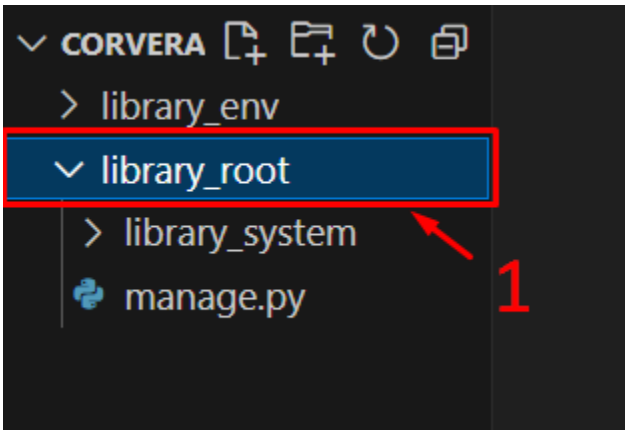




- *Rename* library_system to *library_root*

The Tree View Directory should look like this:

library_root > library_system



- Navigate into the project directory:

cd library_root

```
(library_env) PS C:\Users\Admin\Desktop\CORVERA> cd library_root
(library_env) PS C:\Users\Admin\Desktop\CORVERA\library_root>
```

1

2

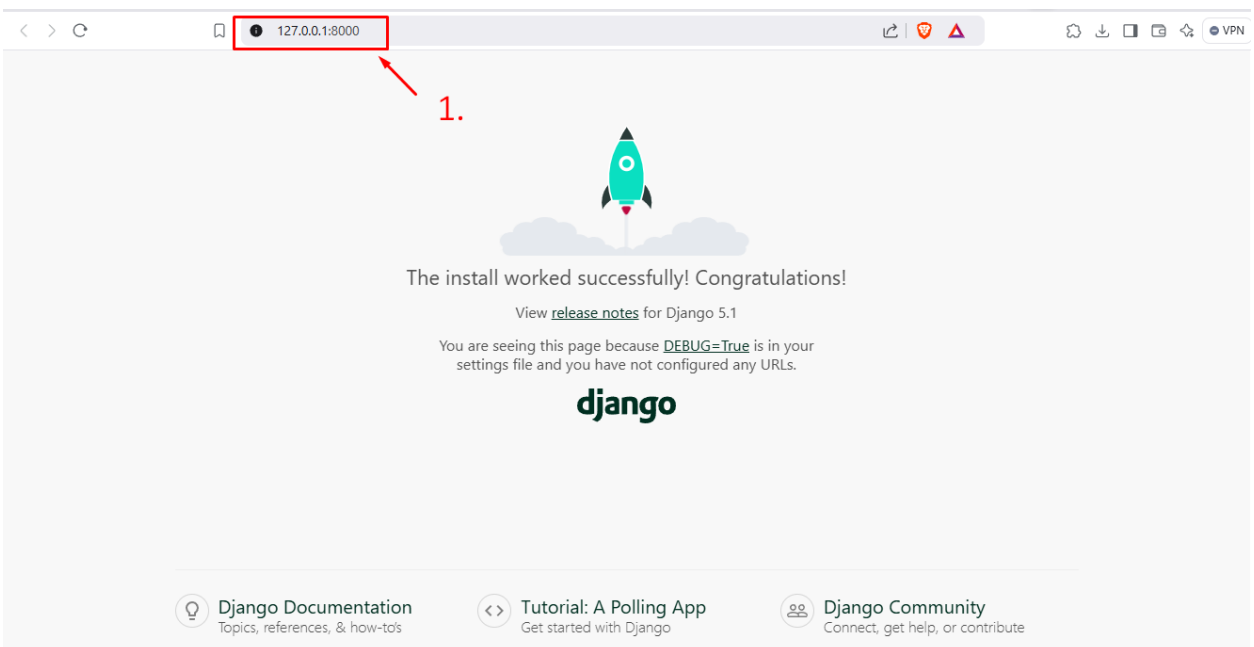
1. **Run the Django Development Server**:

- Start the development server to verify everything is working:

python manage.py runserver

```
(library_env) PS C:\Users\Admin\Desktop\CORVERA\library_root> python manage.py runserver
```

- Open a browser and go to http://127.0.0.1:8000/. You should see the Django welcome page.



Program/Code: The code here is focused on setting up the environment. The following commands should be run in the terminal:

python -m venv library_env

source library_env/bin/activate # or .\library_env\Scripts\activate on Windows

pip install django

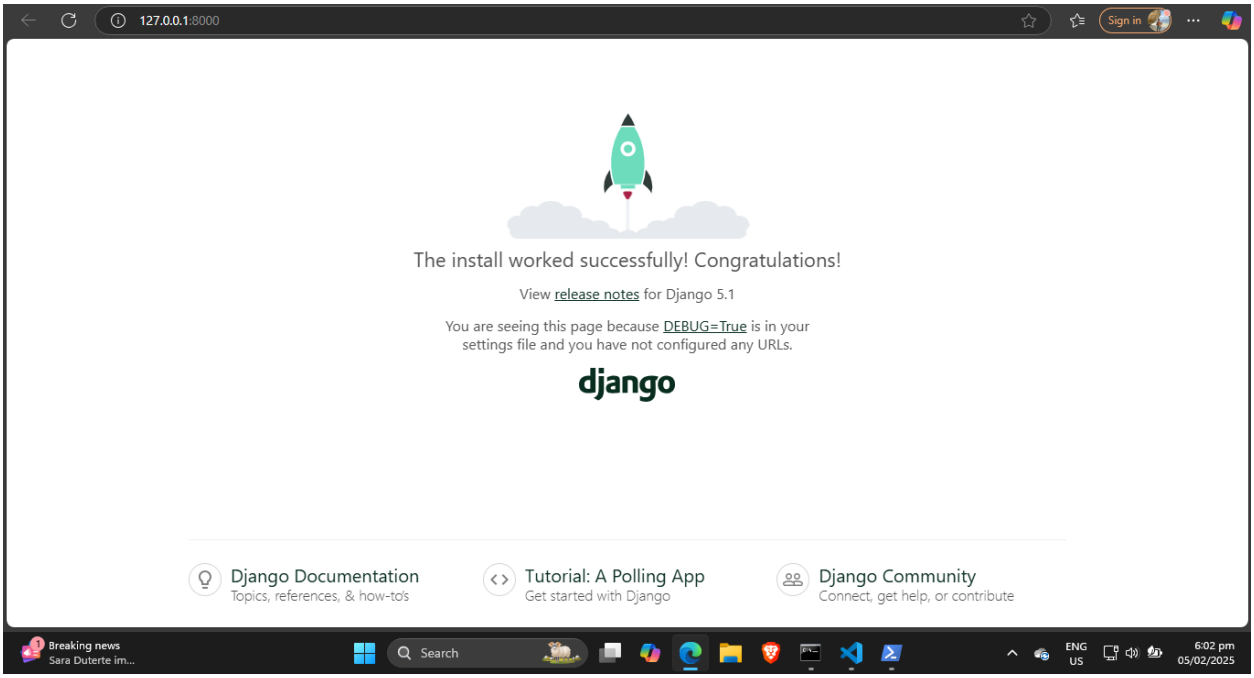
django-admin startproject library_system

cd library_system

python manage.py runserver

ss

Results: (print screen the result and provide the github link of your work)



GitHub link: [mark-alegre01/Website at master]

Follow-Up Questions:

1. What is the role of a virtual environment in Django development?
    - A **virtual environment** in Django isolates dependencies, prevents conflicts, and ensures project consistency. It keeps the project clean, manageable, and deployment ready.
    -
2. What are the advantages of using Django for web development over other frameworks?
    - Django is fast, secure, scalable, and comes with built-in features like ORM, authentication, and an admin panel. Its strong community and "batteries-included" approach make web development efficient.

Findings:

- I found that Django is a powerful, secure, and scalable web framework. It provides built-in features like ORM, authentication, and an admin panel, which streamline development. Using a virtual environment helps me manage dependencies efficiently and avoid conflicts.

Summary:

- Django's "batteries-included" approach makes web development faster and more organized. With built-in security and essential tools, I can develop robust applications while keeping my projects clean and manageable.

Conclusion:

- I see Django as an excellent choice for web development due to its efficiency, security, and built-in functionalities. It simplifies the process, making development smoother and more productive.