

Name: Nic Albert B. Seraño

Course/S.Y: 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):

- Open a terminal and type the following command:

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

3. Install Virtual Environment:

- Create a virtual environment for our project to avoid conflicts with global packages.

```
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
```

1. Install Django:

- After activating the virtual environment, install Django by running:

```
pip install django
```

2. Verify the Django Installation:

- Run the following command to verify if Django is installed:

```
django-admin --version
```

3. Create a New Django Project:

- Create a new Django project called "library_system":

```
django-admin startproject library_system
```

- Navigate into the project directory:

```
cd library_system
```

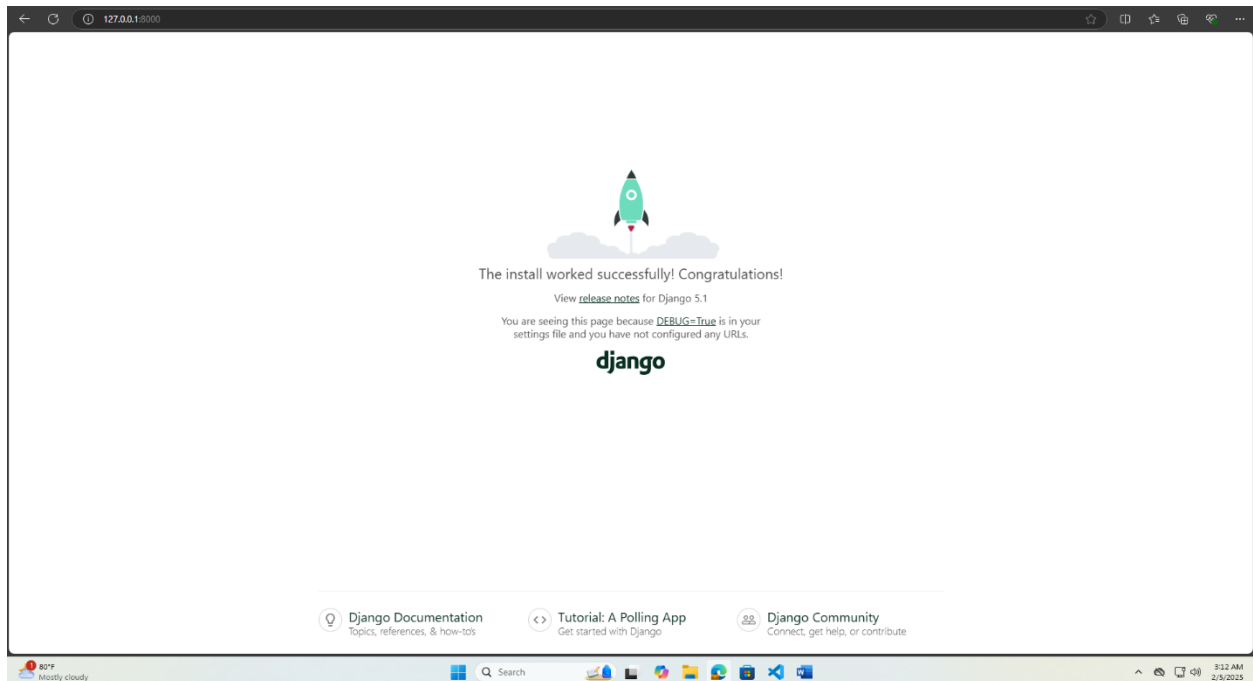
4. Run the Django Development Server:

- Start the development server to verify everything is working:

```
python manage.py runserver
```

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

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



Github link: https://github.com/NicAlbert9/library_system

Follow-Up Questions:

1. What is the role of a virtual environment in Django development?

A virtual environment isolates the project's dependencies from the global Python environment. It ensures that specific package versions are used for the project, preventing conflicts with other projects or system-wide packages.

2. What are the advantages of using Django for web development over other frameworks?

- **Rapid Development:** Django provides built-in features like authentication, admin panels, and ORM, speeding up development.
- **Security:** It includes built-in protection against common security issues like SQL injection and CSRF attacks.
- **Scalability:** Django supports large applications, with tools to manage traffic and databases efficiently.
- **Community Support:** Django has extensive documentation and a large community, offering support and reusable code.

Findings:

- **Virtual Environment** helps isolate project dependencies, ensuring no conflicts.
- **Django** offers rapid development, security features, scalability, and strong community support.

Summary:

A **virtual environment** ensures isolated dependencies, and Django stands out due to its built-in features, security, scalability, and active community, making it an excellent choice for web development.

Conclusion:

Using a **virtual environment** in Django development keeps projects organized and avoids conflicts, while Django's built-in tools and security features make it a powerful framework for rapid, secure, and scalable web development.