GUERRA, CHRISTINE J.

BSCpE – 2A

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.

Procedure:

1. **Install Python**:

o Go to python.org and download the latest version of Python.

o Install Python by following the installation instructions for your operating system.

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

o Open a terminal and type the following command:

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

3. **Install Virtual Environment**:

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

```
pip install virtualenv
```

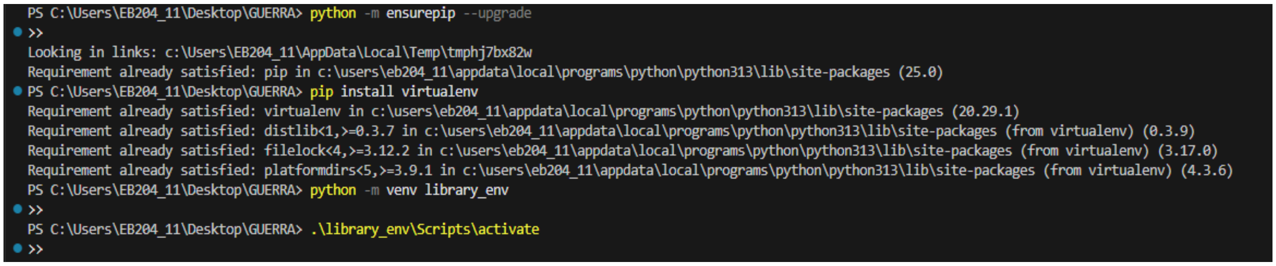
o Create a new virtual environment:

```
python -m venv library_env
```

o Activate the virtual environment:

o On Windows:

```
.\library_env\Scripts\activate
```



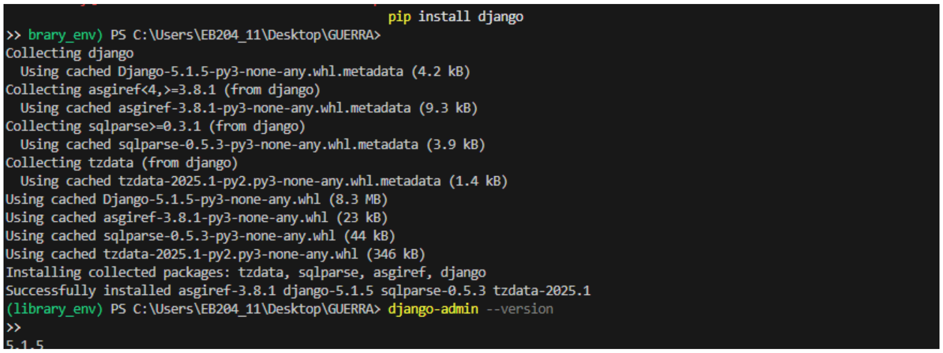
```
PS C:\Users\EB204_11\Desktop\GUERRA> python -m ensurepip --upgrade
>>
Looking in links: c:\Users\EB204_11\AppData\Local\Temp\tmphj7bx82w
Requirement already satisfied: pip in c:\users\eb204_11\appdata\local\programs\python\python313\lib\site-packages (25.0)
PS C:\Users\EB204_11\Desktop\GUERRA> pip install virtualenv
Requirement already satisfied: virtualenv in c:\users\eb204_11\appdata\local\programs\python\python313\lib\site-packages (20.29.1)
Requirement already satisfied: distlib<1,>=0.3.7 in c:\users\eb204_11\appdata\local\programs\python\python313\lib\site-packages (from virtualenv) (0.3.9)
Requirement already satisfied: filelock<4,>=3.12.2 in c:\users\eb204_11\appdata\local\programs\python\python313\lib\site-packages (from virtualenv) (3.17.0)
Requirement already satisfied: platformdirs<5,>=3.9.1 in c:\users\eb204_11\appdata\local\programs\python\python313\lib\site-packages (from virtualenv) (4.3.6)
PS C:\Users\EB204_11\Desktop\GUERRA> python -m venv library_env
>>
PS C:\Users\EB204_11\Desktop\GUERRA> .\library_env\Scripts\activate
>>
```

On Mac/Linux:

source library_env/bin/activate

1. **Install Django**:

o After activating the virtual environment, install Django by running:pip install django

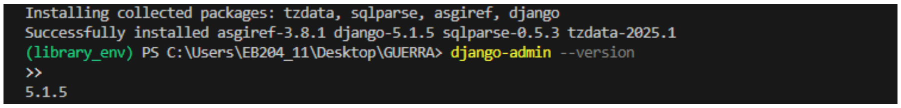


```
                                                 pip install django
>> brary_env) PS C:\Users\EB204_11\Desktop\GUERRA>
Collecting django
  Using cached Django-5.1.5-py3-none-any.whl.metadata (4.2 kB)
Collecting asgiref<4,>=3.8.1 (from django)
  Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django)
  Using cached sqlparse-0.5.3-py3-none-any.whl.metadata (3.9 kB)
Collecting tzdata (from django)
  Using cached tzdata-2025.1-py2.py3-none-any.whl.metadata (1.4 kB)
Using cached Django-5.1.5-py3-none-any.whl (8.3 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.3-py3-none-any.whl (44 kB)
Using cached tzdata-2025.1-py2.py3-none-any.whl (346 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-5.1.5 sqlparse-0.5.3 tzdata-2025.1
(library_env) PS C:\Users\EB204_11\Desktop\GUERRA> django-admin --version
>>
5.1.5
```

2. **Verify the Django Installation**:

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

django-admin --version



```
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-5.1.5 sqlparse-0.5.3 tzdata-2025.1
(library_env) PS C:\Users\EB204_11\Desktop\GUERRA> django-admin --version
>>
5.1.5
```

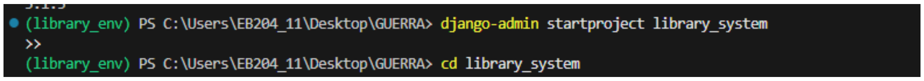
3. **Create a New Django Project**:

o Create a new Django project called "library_system":

django-admin startproject library_system

o Navigate into the project directory:

cd library_system

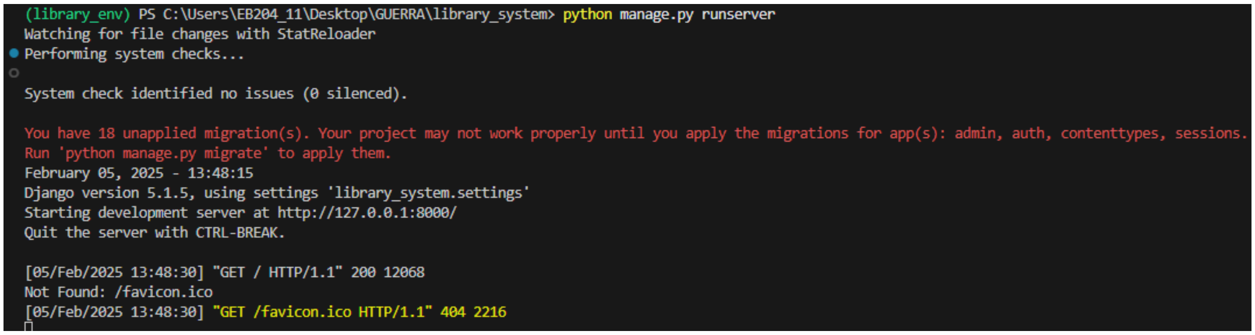


```
5.1.5
(library_env) PS C:\Users\EB204_11\Desktop\GUERRA> django-admin startproject library_system
>>
(library_env) PS C:\Users\EB204_11\Desktop\GUERRA> cd library_system
```

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

- Start the development server to verify everything is working:

python manage.py runserver

- 

```
(library_env) PS C:\Users\EB204_11\Desktop\GUERRA\library_system> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 05, 2025 - 13:48:15
Django version 5.1.5, using settings 'library_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[05/Feb/2025 13:48:30] "GET / HTTP/1.1" 200 12068
Not Found: /favicon.ico
[05/Feb/2025 13:48:30] "GET /favicon.ico HTTP/1.1" 404 2216
```

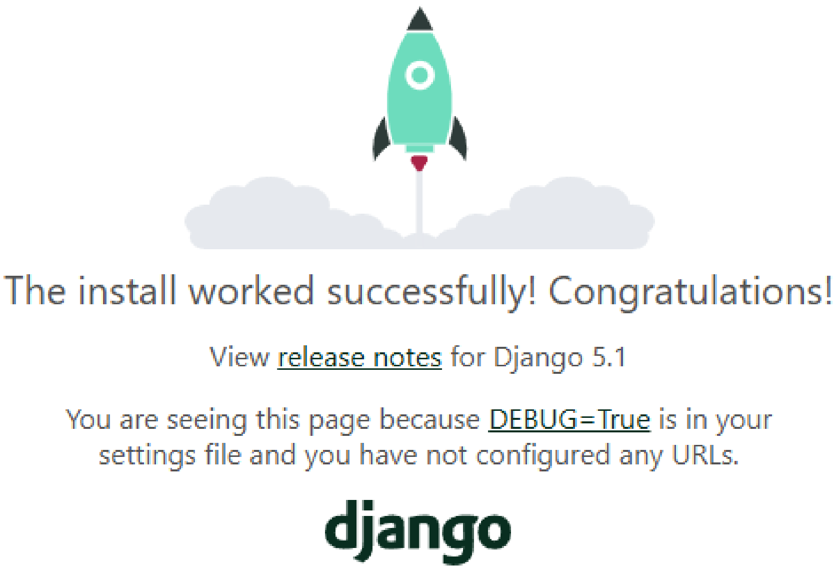
Open a browser and go to http://127.0.0.1:8000/. You should see the Django welcome page.

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
```

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



The install worked successfully! Congratulations!

View release notes for Django 5.1

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

Github Link:

Follow-Up Questions:

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

Ans. A virtual environment in Django development creates an isolated space for each project, ensuring that dependencies don't conflict. This helps keep projects organized and prevents issues when deploying applications.

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

Ans. Django is fast, secure, and scalable. It includes built-in features like authentication, an admin panel, and an ORM, reducing development time and cost. Its strong community support also makes it a reliable choice.

Findings:

Django is a powerful web framework known for its speed, security, and scalability. It simplifies development with built-in tools and a strong community, making it ideal for creating web applications efficiently.

Summary:

Django prioritizes rapid development, security, and cost-effectiveness. Virtual environments help maintain project stability by isolating dependencies and keeping everything organized

Conclusion:

Django is a great choice for web development when speed, security, and scalability are key. Using virtual environments ensures smooth project management and prevents conflicts, making development more efficient.