April Evan Jean Y. Elico

BSCpE – 2A

Laboratory Activity No. 1

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:

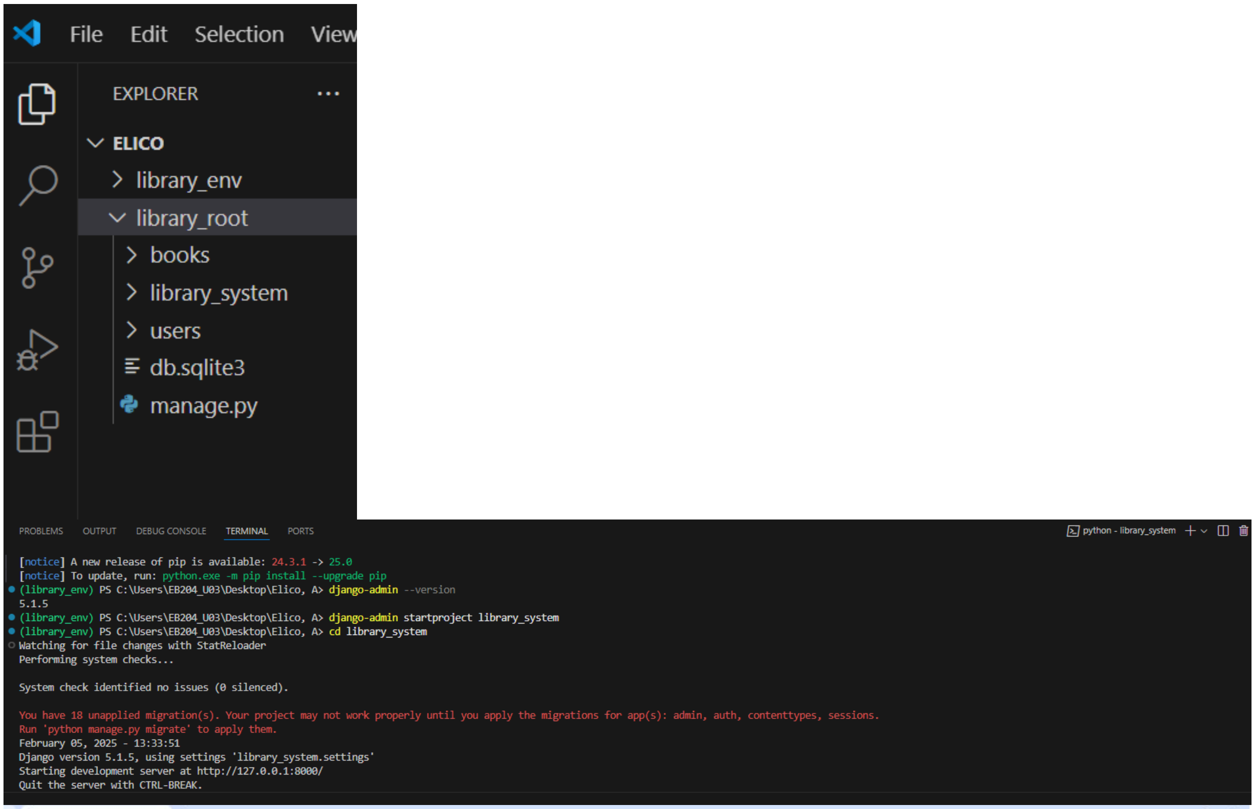
```
python manage.py runserver
```

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

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
```

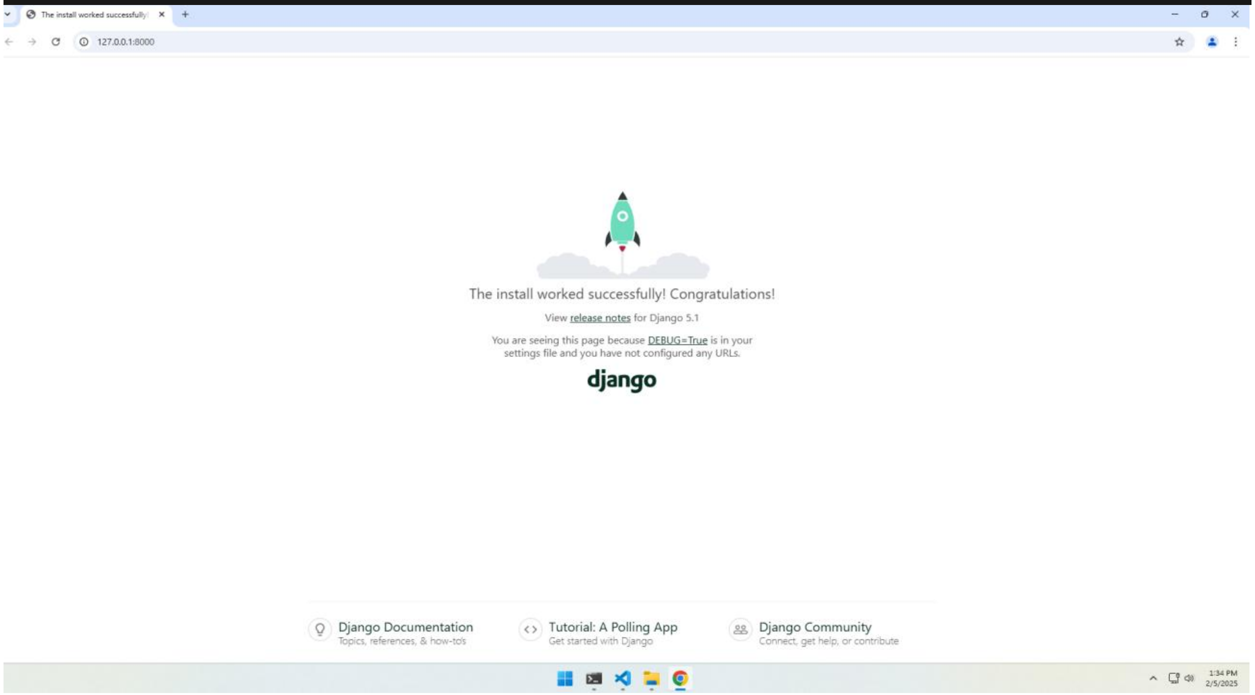
Results: (print screen the result and provide the github link of your work)



```
[notice] A new release of pip is available: 24.3.1 -> 25.0
[notice] To update, run: python.exe -m pip install --upgrade pip
(library_env) PS C:\Users\EB204_U03\Desktop\Elico, A> django-admin --version
5.1.5
(library_env) PS C:\Users\EB204_U03\Desktop\Elico, A> django-admin startproject library_system
(library_env) PS C:\Users\EB204_U03\Desktop\Elico, A> cd library_system
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 - 13:33:51
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.
```



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.

Follow-Up Questions:

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

Ans. A virtual environment in Django development is a separate, clean space for each of your Django projects. It prevents your other projects from messing with each other. This keeps things organized, prevent conflicts, and makes sure your project works the same way on your computer as it does when you put it online.

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

Ans. Django offers several advantages for web development compared to other frameworks. It is fast, secure, and cheaper, making it easier for users to build websites quickly without worrying about the expenses and web attacks.

Findings:

Django is a rapid web development framework known for being scalable, secure, and versatile. It lowers development costs and benefits from a strong community, making it a great choice for building various types of web applications quickly and efficiently.

Summary:

Django is a powerful tool that prioritizes rapid development, security, and cost-effectiveness. Its active community make it a popular choice for developers building a wide range of web applications. Virtual environments play a vital role in ensuring the stability and isolation of Django projects in order to keep things organized.

Conclusion:

Django is a good choice for web development when rapid development, security, and scalability are important considerations. Its features and community make it a reliable and productive framework for creating modern web applications, and using virtual environments alongside Django projects is important for maintaining project integrity and management.