

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
```

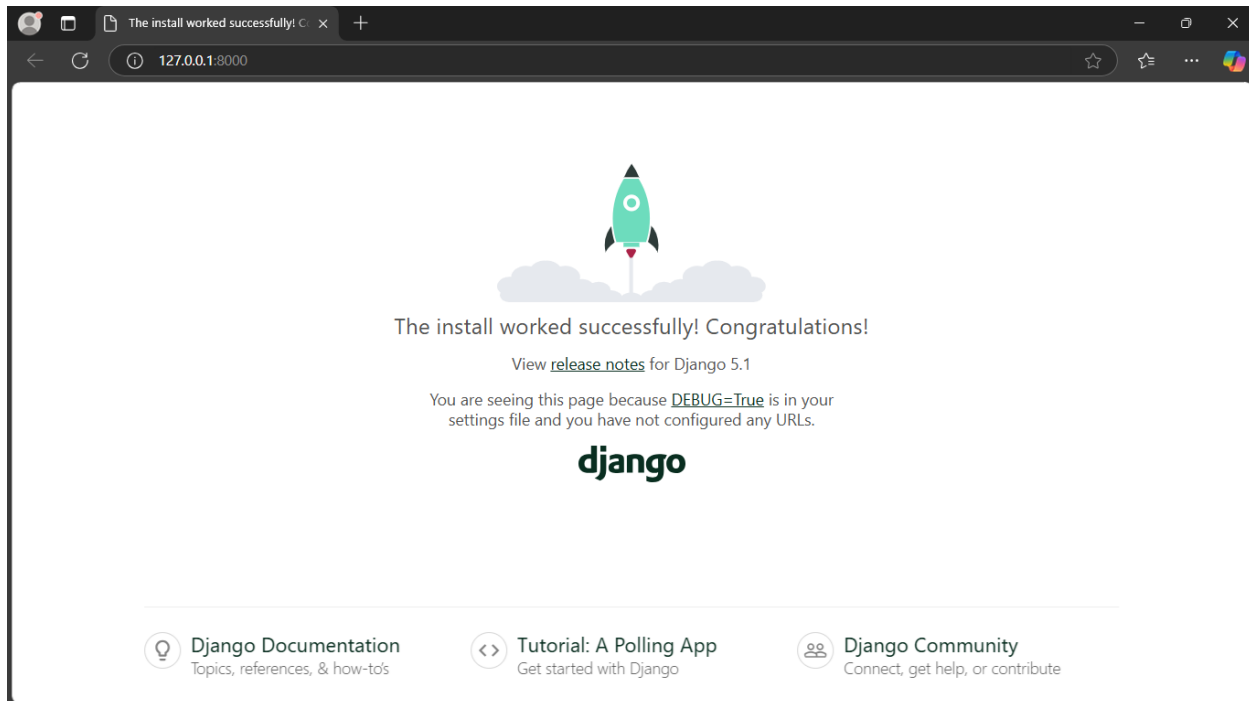
```
django-admin startproject library_system
```

```
cd library_system
```

```
python manage.py runserver
```

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

Django: <http://127.0.0.1:8000/>



My Github Link: https://github.com/sarvida20/My-Project-2025/tree/99101723ff519b2e3b5623c9ba1ca663ceab3fd7/library_system

Follow-Up Questions:

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

Ans: A virtual environment helps manage project-specific dependencies, preventing conflicts between different Python projects. It ensures that each Django project has its own isolated set of packages.

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

Ans: **Built-in Features** – Django comes with an admin panel, authentication, and security features.

Scalability – It supports large-scale applications with ease. **Security** – Protects against SQL injection, CSRF, and XSS attacks.

ORM (Object-Relational Mapping) – Allows easy database management.

Rapid Development – Django's structure allows developers to build applications quickly.

Findings:

Django installation was successful.

The development server ran without errors.

The virtual environment was correctly set up.

Summary:

The activity covered setting up the Django development environment, including Python installation, creating a virtual environment, installing Django, and verifying the installation by running a test project.

Conclusion:

Successfully setting up Django is a crucial step before starting web development. This activity provided hands-on experience in configuring a proper development environment, ensuring a smooth workflow for future projects.