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:

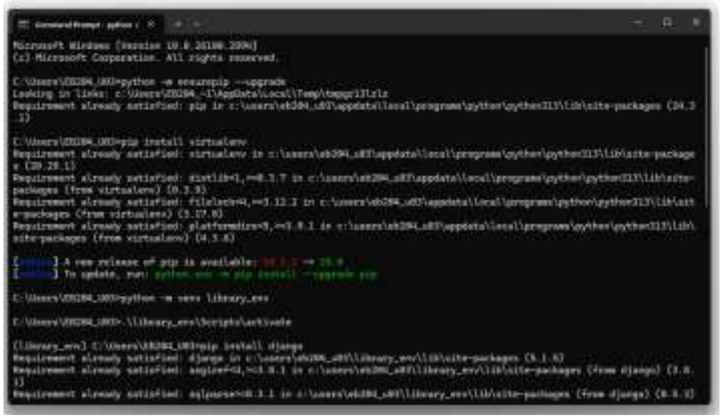
```
python manage.py runserver
```

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

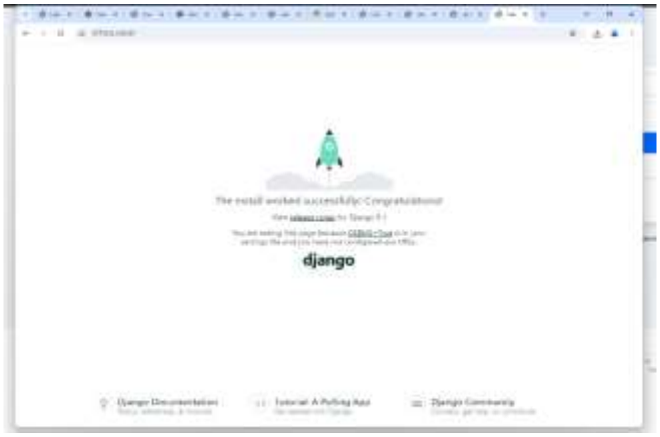
Program/Code: The code here is focused on setting up the environment. The following commands should be run in the terminal:

```
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)



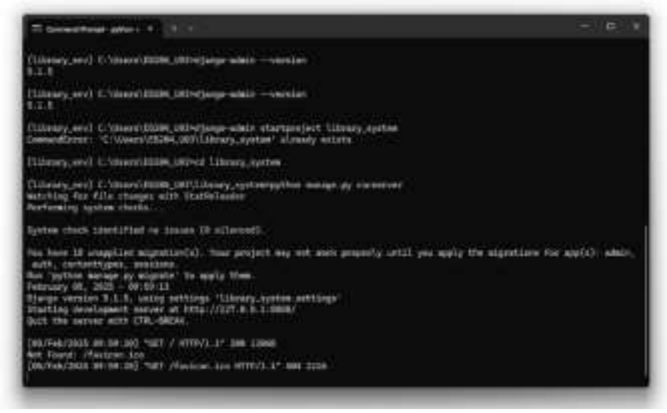
```
Microsoft Windows [Version 10.0.26100.2894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\EB204_U03>python -m ensurepip --upgrade
Looking in links: c:\Users\EB204_~1\AppData\Local\Temp\tmpgri3lzlz
Requirement already satisfied: pip in c:\users\eb204_u03\appdata\local\programs\python\python313\lib\site-packages (24.3
.1)

C:\Users\EB204_U03>pip install virtualenv
Requirement already satisfied: virtualenv in c:\users\eb204_u03\appdata\local\programs\python\python313\lib\site-package
s (20.29.1)
Requirement already satisfied: distlib<1,>=0.3.7 in c:\users\eb204_u03\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_u03\appdata\local\programs\python\python313\lib\sit
e-packages (from virtualenv) (3.17.0)
Requirement already satisfied: platformdirs<5,>=3.9.1 in c:\users\eb204_u03\appdata\local\programs\python\python313\lib\
site-packages (from virtualenv) (4.3.6)

[notice] A new release of pip is available: 24.3.1 -> 25.0
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\EB204_U03>python -m venv library_env

C:\Users\EB204_U03>.\library_env\Scripts\activate

(library_env) C:\Users\EB204_U03>pip install django
Requirement already satisfied: django in c:\users\eb204_u03\library_env\lib\site-packages (5.1.6)
Requirement already satisfied: asgiref<4,>=3.8.1 in c:\users\eb204_u03\library_env\lib\site-packages (from django) (3.8.
1)
Requirement already satisfied: sqlparse>=0.3.1 in c:\users\eb204_u03\library_env\lib\site-packages (from django) (0.5.3)
```



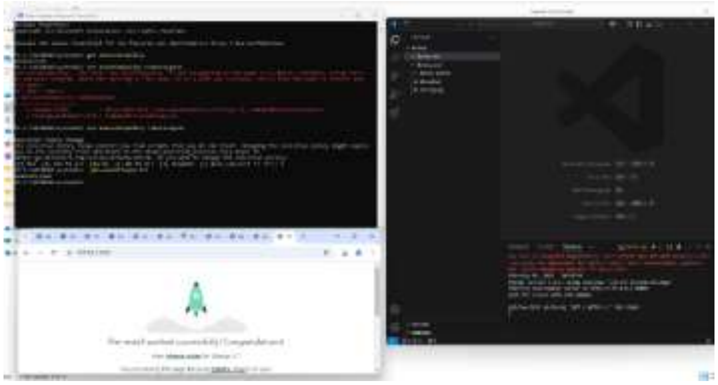
The install worked successfully! Congratulations!

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.



```
(library_env) C:\Users\EB204_U03>django-admin --version
5.1.6

(library_env) C:\Users\EB204_U03>django-admin --version
5.1.6

(library_env) C:\Users\EB204_U03>django-admin startproject library_system
CommandError: 'C:\Users\EB204_U03\library_system' already exists

(library_env) C:\Users\EB204_U03>cd library_system

(library_env) C:\Users\EB204_U03\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 08, 2025 - 09:59:13
Django version 5.1.6, using settings 'library_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[08/Feb/2025 09:59:20] "GET / HTTP/1.1" 200 12068
Not Found: /favicon.ico
[08/Feb/2025 09:59:20] "GET /favicon.ico HTTP/1.1" 404 2216
```



The install worked successfully! Congratulations!

Follow-Up Questions:

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

- A virtual setting in Django development separates project requirements, avoiding clashes with globally installe d packages   and guaranteeing a tidy, movable, and uniform development space.

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

   - Django provides quick development, the ability to scale, integrated security features, flexibility, an automatically created admin panel, and a comprehensive, well-explained environment, which makes it a favored option for building websites.

- **Findings**:
  - The results of Django showcase its advantages in quick development, safety, adaptability, and user-friendliness. It streamlines the process of web development with integrated functionalities such as an administrative dashboard, user login systems, and database transitions. The solid framework of Django promotes optimal practices, and its active community along with thorough documentation offers significant assistance. Furthermore, it is extremely adaptable, which means it can cater to both minor projects and extensive applications.

Summary:

-Django is a robust framework for web development, recognized for its ability to facilitate quick development, robust security, and adaptability. It comes with built in resources such as an administrative dashboard and user authentication, streamlining the development process. Django promotes effective practices, and its extensive community and comprehensive documentation provide outstanding assistance. It is appropriate for applications of any size, positioning it as a flexible and dependable option for building websites.

Conclusion:

In conclusion, Django is a strong and effective framework that makes web development easier by supporting best practices, providing built-in capabilities, and guaranteeing excellent security.

Its extensive documentation and vibrant community offer robust support, and its scalability and agility make it appropriate for projects of all sizes.

Django is a great option for developers creating dependable, scalable online applications because of its robust security features, which also help to speed up development and minimize manual coding.