**GUIDE**

.

**PREREQUISITES**

Before getting started, ensure that your computer meets the following requirements:

- Operating System: Windows, macOS, or Linux

- Python: Installed (Python 3.6 or higher)

- Pip: Installed (Python package manager)

- Git: Installed (optional, but useful for cloning the project)

**STEP-BY-STEP GUIDE**

1. Install Python:

- Download Python: Visit the [Python official website](https://www.python.org/downloads/) and download the latest version of Python (3.6 or higher) for your operating system.

- Run the installer and follow the installation instructions.

- Make sure to check the option that says "Add Python to PATH" during installation.

2. Install Git (Optional):

- If you plan to clone the project using Git, download and install Git from the [Git official website](https://git-scm.com/downloads) for your operating system.

3. Clone the Project (Optional):

- Open a terminal or command prompt.

- Navigate to the directory where you want to store the project.

- Run the following command to clone the project (replace `<project-url>` with the URL of your project's repository):

```bash

git clone <project-url>

```

4. Navigate to the Project Directory

- Use the terminal or command prompt to navigate to the project directory where you cloned or saved the project.

5. Create a Virtual Environment:

- To keep project dependencies isolated, create a virtual environment. Run the following command:

```bash

python -m venv myenv

```

- Activate the virtual environment:

- On Windows:

```bash

myenv\Scripts\activate

```

- On macOS and Linux:

```bash

source myenv/bin/activate

```

6. \*\*Install Project Dependencies:

- While the virtual environment is active, install the project's dependencies using pip:

```bash

pip install -r requirements.txt

``

7.Migrate the Database

- Run the following command to apply migrations and create the database:

```bash

python manage.py migrate

```

8. Create a Superuser (Admin Account):

- You can create an admin account to access the Django admin panel:

```bash

python manage.py createsuperuser

```

- Follow the prompts to create a username, email, and password.

9. Run the Development Server:

- Start the Django development server:

```bash

python manage.py runserver

```

- The server will start, and you'll see output indicating that the project is running.

10. Access the Project:

- Open a web browser and go to `http://127.0.0.1:8000/` to access the project locally.

11. Access the Admin Panel (Optional):

- To access the Django admin panel, go to `http://127.0.0.1:8000/admin/` and log in with the superuser credentials you created earlier.

12. Explore the Project:

- You can now explore and interact with the project using your web browser.