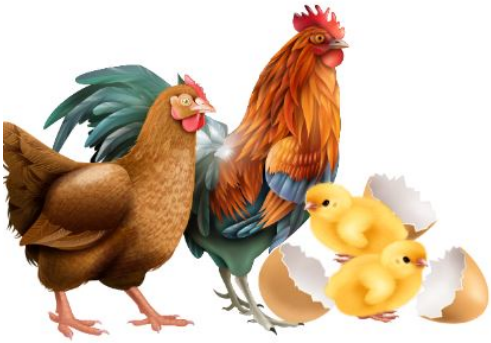
USER MANUAL



By

DUCK ON TREE

version1.1 bishwoprakshdhakal realsed on 17/10/2023

SETUP DEPLOYMENT ENVIRONMENT

This document will get you up and running with Django.

Install Python

Django is a Python web framework. See What Python version can I use with Django? for details.

Get the latest version of Python at https://www.python.org/downloads/ or with your operating system's package manager.

Python on Windows

If you are just starting with Django and using Windows, you may find How to install Django on Windows useful.

Install Apache and

mod_wsgi

If you just want to experiment with Django, skip ahead to the next section; Django includes a lightweight web server you can use for testing, so you won't need to set up Apache until you're ready to deploy Django in production.

If you want to use Django on a production site, use Apache with mod_wsgi. mod_wsgi operates in one of two modes: embedded mode or daemon mode. In embedded mode, mod_wsgi is similar to mod_perl – it embeds Python within Apache and loads Python code into memory when the server starts. Code stays in memory throughout the life of an Apache process, which leads to significant performance gains over other server arrangements. In daemon mode, mod_wsgi spawns an independent daemon process that handles requests. The daemon process can run as a different user than the web server, possibly leading to improved security. The

daemon process can be restarted without restarting the entire Apache web server, possibly making refreshing your codebase more seamless. Consult the mod_wsgi documentation to determine which mode is right for your setup. Make sure you have Apache installed with the mod_wsgi module activated. Django will work with any version of Apache that supports mod_wsgi.

See How to use Django with mod_wsgi for information on how to configure mod_wsgi once you have it installed.

If you can't use mod_wsgi for some reason, fear not: Django supports many other deployment options. One is uWSGI; it works very well with nginx. Additionally, Django follows the WSGI spec (**PEP 3333**), which allows it to run on a variety of server platforms.

Install the Django code

Installation instructions are slightly different depending on whether you're installing a distribution-specific package, downloading the latest official release, or fetching the latest development version.

This is the recommended way to install Django.

1. Install pip. The easiest is to use the standalone pip installer. If your distribution already has

2. pip

3. installed, you might need to update it if it's outdated. If it's outdated, you'll know because installation won't work.

4. Take a look at venv. This tool provides isolated Python environments, which are more practical than installing packages systemwide. It also allows installing packages without administrator privileges. The contributing tutorial walks through how to create a virtual environment.

5. After you've created and activated a virtual environment, enter the command:

6. ñ|/ñy

7.

8. ñz

```
$ python -m pip install Django
```

Check the distribution specific notes to see if your platform/distribution provides official Django packages/installers. Distribution-provided packages will typically allow for automatic installation of dependencies and supported upgrade paths; however, these packages will rarely contain the latest release of Django.

Tracking Django development

If you decide to use the latest development version of Django, you'll want to pay close attention to the development timeline, and you'll want to keep an eye on the release notes for the upcoming release. This will help you stay on top of any new features you might want to use, as well as any changes you'll need to make to your code when updating your copy of Django. (For stable releases, any necessary changes are documented in the release notes.)

If you'd like to be able to update your Django code occasionally with the latest bug fixes and improvements, follow these instructions:

1. Make sure that you have Git installed and that you can run its commands from a shell. (Enter
2. git help
3. at a shell prompt to test this.)
4. Check out Django's main development branch like so:
5. ñ|/ñy
6.
7. ñz

```
$ git clone https://github.com/django/django.git
```

1. This will create a directory
2. django
3. in your current directory.
4. Make sure that the Python interpreter can load Django's code. The most convenient way to do this is to use a virtual environment and pip. The contributing tutorial walks through how to create a virtual environment.

5. After setting up and activating the virtual environment, run the following command:
6. ñ|/ñy
7.
8. ñz

```
$ python -m pip install -e django/
```

1. This will make Django’s code importable, and will also make the
2. django-admin
3. utility command available. In other words, you’re all set!

When you want to update your copy of the Django source code, run the command

git pull

from within the

django

directory. When you do this, Git will download any changes.

Downloading Code from Github:

To download Django's code from GitHub, you can follow these steps:

1. Make sure you have Git installed and can run its commands from a shell.

2. Open a shell and navigate to the directory where you want to download Django's code.

3. Run the following command:

`git clone git@github.com:bishwopr/creative-poultry.git

This will create a directory named `creative-poultry` in your current directory.

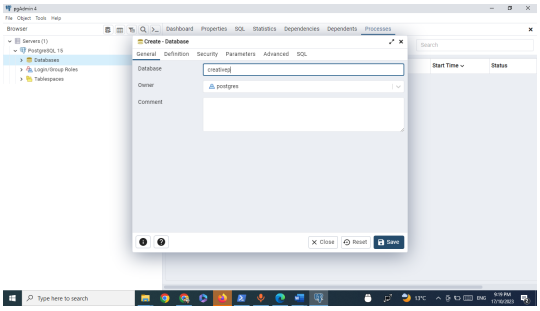
4. Once the download is complete, you can proceed with setting up Django in your Python environment as mentioned earlier.

Get your database running

If you plan to use Django’s database API functionality, you’ll need to make sure a database

server is running. Django supports many different database servers and is officially supported with PostgreSQL

Create a database on Postgres and note down name and server password:



Now Navigate to base dir of code and find cpcoultry.

open settings.py with a text editor and find the database heading edit the database name user name and password for the Postgres server and save.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'cpfg',
        'USER': 'postgres',
        'PASSWORD': '1234',
        'HOST': 'localhost',
        'PORT': '5432',
    }
```

Creating Admin account:

To create a Django superuser, follow these steps:

1. Ensure that you have Django installed by following the installation instructions mentioned earlier in the document.

2. Open a command prompt or terminal and navigate to your Django project directory.

3. Run the following command to create a superuser:

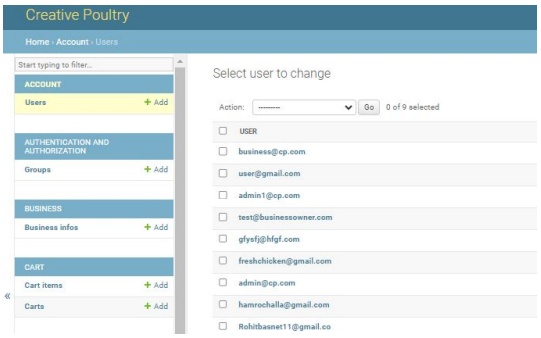
```
python manage.py createsuperuser
```

1. You will be prompted to enter a username, email (optional), and password for the superuser. Provide the required information and press Enter.

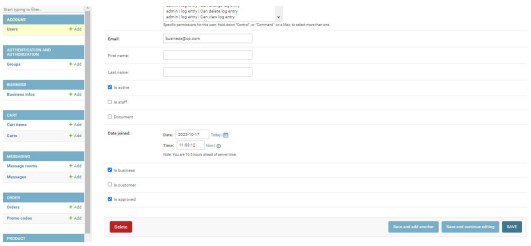
2. Django will then create the superuser account and display a success message.

You can now use this superuser account to access the Django admin interface and perform administrative tasks for your project.

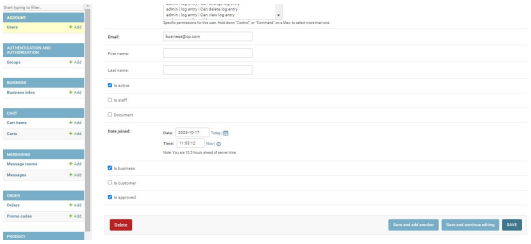
Login as admin and navigate to recently added user:

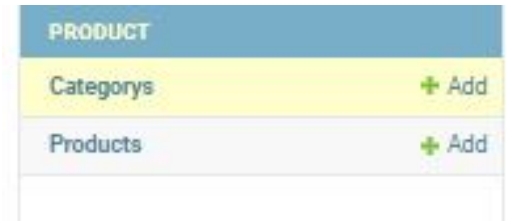


Click on User and navigate to is approve checkbox at end and check the box and save:



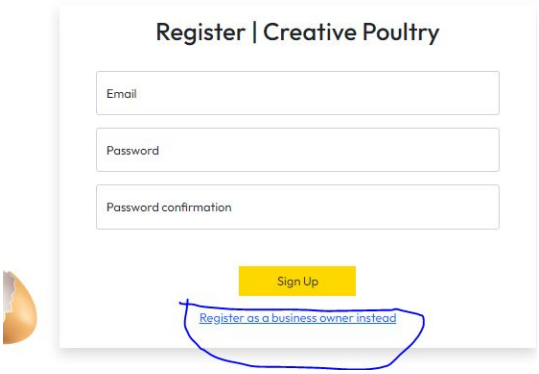
Click on User and navigate to the delete button at the end click and save:



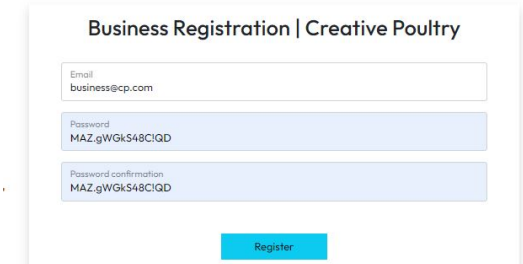


Creating Business Owner Account:

Click on register as buiness owner link on botton of landing page:



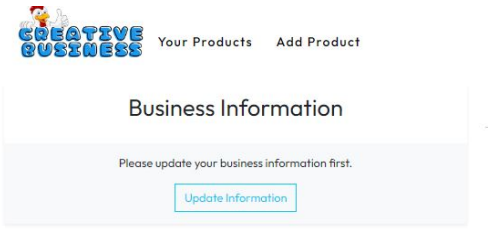
Add valid E-mail and password:



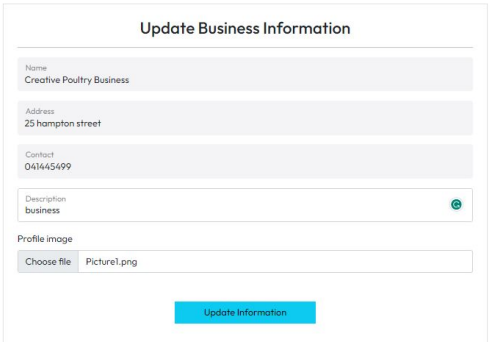
It will prompt this message:

Adding Business Information:

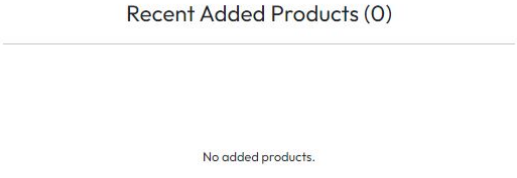
Login as a business owner and click on update information:



Provide name,address,contact,description,image and click on update information:

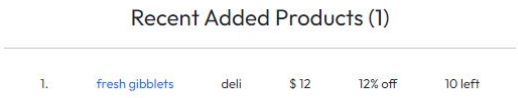


Click on Add products button on landing page:

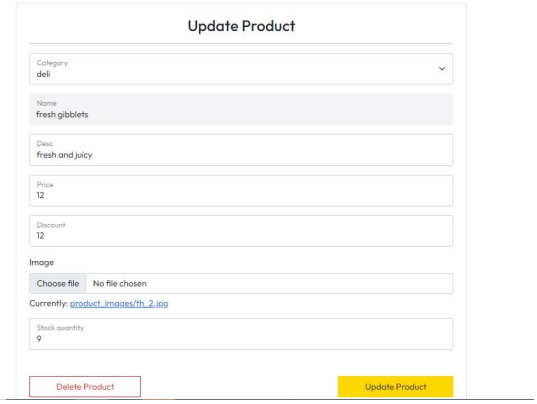


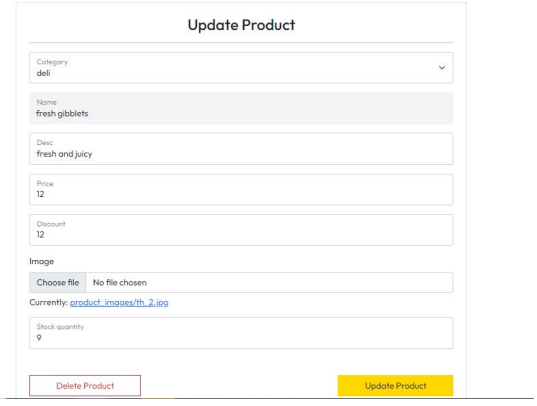
On the product page, fill in the details for each product, including name, description, price, and image. Then click on the "Add product" button to save the information.

Navigate to Recent Added products



Click on the product and make changes and click on update product:

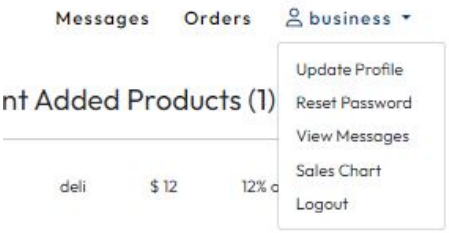




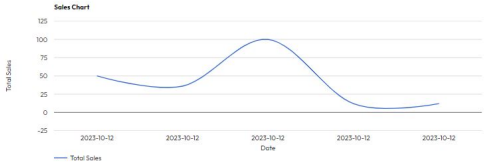
Click on delete product.

Click on drop down menu fromprofile name:

click on sales chart

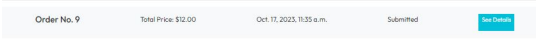


The sales vs date chart will be shown:

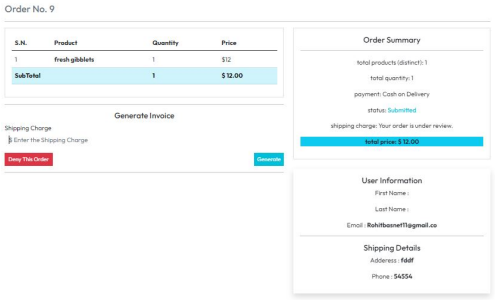


login as a business owner and navigate to the orders menu in the navigation bar:

Click on order details:

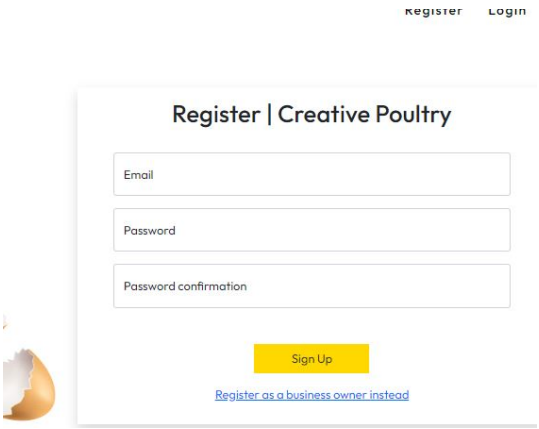


enter delivery charge to Approve or click deny this order to deny order:

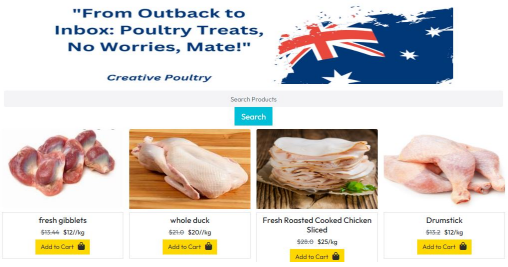


Register as a User:

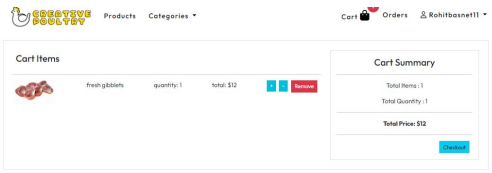
Navigate to Register page from navigation bar , fill email and password and click signup:



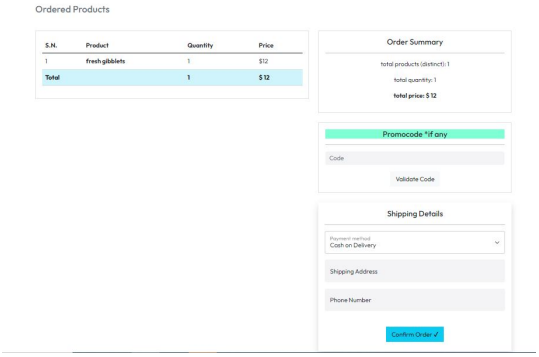
After successful login it will navigate to products landing page:



Click on image for more information or click add to cart directly:



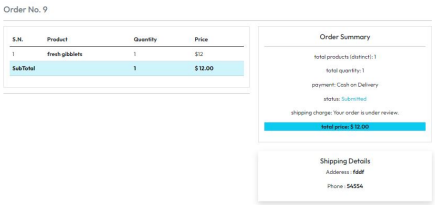
Increase or decrease the quantity and procced to checkout:



:

Fill in the required details for payment and shipping and click on "Complete Order":

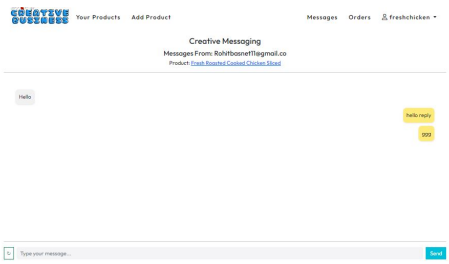
the order will now be submitted:



viewing customer message:

click on view messages fromprofile drop down menu:

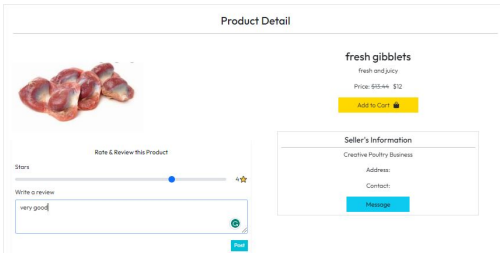
choose messages and see details:



Providing rating and review of products:

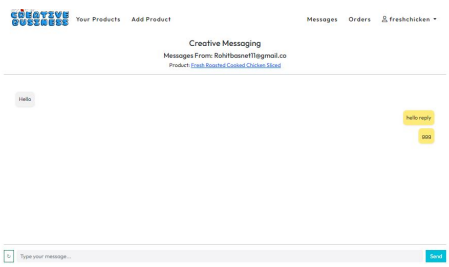
Navigate to the Purchased product page:

Drag the rating bar 1 to 5 provide a comment and click post



Sending business message:

Navigate to product page and click on send message to seller



Compose your message and click send. The seller will receive your message and respond accordingly.595e1773-16b3