

Nestegg Developer and User Manual

Casey Weed, Dax Lloyd, William McMullen, Christopher Wilson

May 7, 2015

Contents

1	Developer's Manual	1
1.1	Technical Description Preface	1
1.1.1	Front-End Design	1
1.1.2	Back-End Design	1
1.2	Setup	3
1.2.1	Installing MySQL	3
1.2.2	Installing Redis	4
1.2.3	Installing Python	4
1.2.4	Installing Nestegg Requirements	4
1.2.5	Signing up for Google reCAPTCHA	4
1.2.6	Signing up for Stripe	5
1.3	Configuration	6
1.3.1	Customizing <code>config.py</code>	6
1.3.2	Customizing <code>stripe.keys.py</code>	7
1.3.3	Establishing Static Directories and MySQL	7
1.4	Deployment	8
1.4.1	Small Scale	8
1.4.2	Large Scale	8
	Nginx	8
	uWSGI	9
	Supervisor	9
2	User's Manual	11
2.1	Signing Up	11
2.2	Pro Membership	13
2.3	Uploading a File	15
2.4	Editing & Deleting a File	18
2.5	Searching for Images	20
A	Database Schema	22

1 Developer's Manual

1.1 Technical Description Preface

Nestegg uses a variety of technology to establish its presence. The purpose and execution of each is briefly detailed below.

1.1.1 Front-End Design

We used HTML, CSS, and a minor amount of JavaScript with jQuery for the front-end graphics and development of Nestegg. We used a HTML & CSS framework called “Skeleton”¹ as a boilerplate template to structure the general design and feel for the site. The Skeleton framework allows for the site to be used on mobile devices as it is a responsive CSS design. Simplicity and ease of use were the main goals in designing Nestegg and Skeleton facilitated that.

1.1.2 Back-End Design

Python The primary language for the application was written using Python. Python is a high-level object oriented programming language.

Flask The micro web framework used to create the entirety of the application. Flask follows a more “hands off” doctrine giving the developer freedom to apply solutions how they see fit.

SQLAlchemy A Python ORM made for enterprise solutions, performs well at scale and is ideal for large scale operations. However, its API can be tricky to use at times.

MySQL DBMS to store the information acquired from users. Nestegg requires only two tables (one for user information, another for files) in a single database to function.

Redis Open source, BSD licensed, advanced key/value (dictionary) in-memory store. We used Redis to store client session information instead of using the default method of file based caching to avoid needless I/O strain.

¹<http://getskeleton.com>

Stripe Merchant² and financial information management. Stripe sports a robust API that is secure, and makes it incredibly easy to handle payments.

For each transaction a unique ID and event type is created for each user when they submit their payment information. This ID is then stored in our database and is used to cancel the subscription at another point in time.

Stripe's Webhooks allow us to automate the process of canceling user subscriptions once they expire or cannot be renewed. We manage an endpoint on the server which Stripe sends a JSON payload to, we then retrieve the unique customer ID from the payload and remove the user's Pro account status.

Google reCAPTCHA Google's CAPTCHA³ service is simple to use. The short simple bot-resistant test lets us filter humans from automated users.

²<https://stripe.com>

³<https://www.google.com/recaptcha/intro/index.html>

1.2 Setup

You should have a copy of the Nestegg application on hand and have it extracted onto the target machine which will be hosting the application before proceeding.

Note:

This guide assumes all the services will be installed to the same machine.

Nestegg deploys like any other typical web application written using Django, web2py, and so on. If you have experience deploying any applications using these web frameworks you should be familiar with the process. However, before we are able to install the application itself, several services need to be installed and configured beforehand.

1.2.1 Installing MySQL

1. You will need to install MySQL for your required platform. For Debian based platforms you can use **Apt** package manager to install MySQL server using `apt-get install mysql-server mysql-client`. Additional packages required by the client and server packages will be included automatically by Apt.
2. During the installation you will be required to enter a password for the root user. Set the password and remember it, it will be required for creating our user for Nestegg later on.
3. After the installation completes running `mysql_secure_installation` is highly recommended if in a production setting⁴
4. Login to the root user for MySQL using `mysql -u root -p`. You will then be asked to enter the root password from earlier, input it now.
5. Now we will need to execute the SQL script included with Nestegg (`create_db.sql`). Before running it is recommended that you change password for the `nestegg` user, see listing 1 for the line of interest.

⁴`mysql_secure_installation` will perform execute simple security measures ideal for a production environment. You may skip this if in a development setting.

Listing 1: create_db.sql

```
CREATE USER 'nestegg'@'localhost' IDENTIFIED BY 'nestegg';
```

6. To execute the SQL script, run `source create_db.sql`.
7. To check if the database was created successfully run `show databases;` and check if `nestegg` is included in the list of databases.

1.2.2 Installing Redis

Installing Redis is very straightforward with Debian based systems. Simply run `apt-get install redis-server`. The default options are satisfactory for Nestegg.

1.2.3 Installing Python

Similar to Redis, installing Python is an easy process. On Debian based system run `apt-get install python27 python27-dev python-pip`.

1.2.4 Installing Nestegg Requirements

Convenient for us, installing Nestegg's list of required Python packages is made easy thanks to PIP. To install these packages run `pip install -r requirements.txt` using the `requirements.txt` from the Nestegg package.

If you encounter an error while installing `python-bcrypt` you may need to install the `bcrypt` package using Apt.

1.2.5 Signing up for Google reCAPTCHA

1. Navigate to <https://www.google.com/recaptcha/> and create a new site with your domain (if using localhost for development simply enter 'localhost').
2. Keep the 'Site' and 'Secret' key on hand, we will need them later for configuring reCAPTCHA within Nestegg.

1.2.6 Signing up for Stripe

Stripe is a merchant and provides a fantastic API for all kinds of financial transactions and a handy dandy web interface from which to observe or manage all of your data. Near the end of the sign up process we will need the 'Public' and 'Private' keys. Afterwards we'll create the webhook endpoint.

1. Navigate to <https://stripe.com>.
2. If you are already a member then sign in, if not from the Sign In page choose Sign Up, fill in the appropriate information and login.
3. At the dashboard in the top-right corner, click your account name and from the resulting dropdown list choose "Account Settings".
4. From the top portion of modal dialog, choose "API Keys".
5. Copy the set of keys appropriate for your environment (e.g. if in development using testing keys, production use live).
6. Next transition to the "Webhooks" tab.
7. Press "Add endpoint".
8. Enter the URL from which you will be hosting the application and append `/user/hook`. This will be the URL which Stripe will send event information to for processing. If you are testing or developing from a local machine you will need to sign up and use the URL provided by the Ultrahook⁵ service.
9. Then select the appropriate mode.
10. Next, Choose the radio item "Select events".
11. From the list of events choose only "customer.subscription.deleted" and "customer.subscription.updated".
12. Finally, press the "Create Endpoint" button.

⁵<http://ultrahook.com> allows you to perform testing with Stripe Webhooks from local machines.

1.3 Configuration

Before we can run the application we must configure it to use the services we've installed earlier.

1.3.1 Customizing `config.py`

`config.py` contains the majority of the configurable content for Nestegg. Including the database URI, reCAPTCHA keys and more. We are primarily concerned with the first three lines under the `Config` object.

Listing 2: `config.py`

```
3 class Config(object):
4     SQLALCHEMY_DATABASE_URI = 'mysql://nestegg:
      nestegg@localhost/nestegg'
5     RECAPTCHA_PUBLIC_KEY = '6
      LfrqAETAAAAA0_BbffXgjLZJ_iEOHNGBBy5Jo2z'
6     RECAPTCHA_PRIVATE_KEY = '6LfrqAETAAAAABe7HiBUZAMy9M-
      lp7kunMwp42eB'
7     IMAGES_SET = set(['jpg', 'jpeg', 'png', 'gif'])
8     TESTING = False
9     DEBUG = False
10    UPLOAD_DIRECTORY = 'static/files'
11    IMAGE_URL_PATH = 'files/'
12    IMAGES_PATH = [UPLOAD_DIRECTORY]
13    SECRET_KEY = 'changeme'
```

- Replace the reCAPTCHA public and private keys with the keys were attained earlier (see listing 2 line № 5-6).
- Replace the database URI in `SQLALCHEMY_DATABASE_URI` on with the appropriate credentials (see listing 2 on line № 4).
- Replace the contents of `SECRET_KEY` with a long and preferably completely random string. This is used for signing sessions and other cryptographic information.

The remaining information such as `IMAGES_SET` and `UPLOAD_DIRECTORY` can also be configured, but it is recommended to leave them as is.

1.3.2 Customizing `stripe_keys.py`

The Stripe API keys must be configured within Nestegg to access your account's credentials.

1. Create a new Python file labeled `stripe_keys.py` within the root folder of Nestegg.
2. Within the file enter the following, replacing the contents of the public and secret keys with those from Stripe (see listing 3).

Listing 3: `stripe_keys.py`

```
public = 'public_key'
secret = 'secret_key'
```

3. Save the file.

1.3.3 Establishing Static Directories and MySQL

Before Nestegg can be run a the static directory (as defined in `config.py` must be created along with the appropriate tables in the MySQL database. This is incredibly easy using `run.py`. See appendix A for a diagram of the database tables.

1. Perform `python run.py setup_dirs`. The static directory for uploaded content will be created.
2. Next, execute `python run.py reset --yes`. This will use the information from `config.py` to create our tables.

1.4 Deployment

For our purposes deployment on a small scale is easy. Large scale deployment is more involved and the gist of large scale deployment will be explained.

1.4.1 Small Scale

Using `run.py` we can start a small server fit for low traffic or development purposes.

Execute `python run.py run` to start a server accessible via the local machine at port 5000. If you would like to make the server public, use the `--host` parameter to change the host IP to `0.0.0.0`, for example `python run.py run --host=0.0.0.0`. The same process can be used to change the port, just use the `--port` argument. If you need further assistance see the help dialog for more information (`python run.py run --help`).

1.4.2 Large Scale

Large scale operations need to be able to process thousands of requests per second. This is a problem for the built-in Werkzeug server. The solution is to use a combination of software (namely nginx, supervisor, uwsgi and optionally virtualenv) to provide the application.

Nginx Nginx is ideal for reverse proxies, HTTP load balancing or in general handling large amounts of concurrent requests. The configuration setup is quite simple for Nginx, simply create a new server block directive with the following in listing 4.

Listing 4: Nginx Example

```
upstream nestegg {
    server http://your.internal.server.address;
}

server {
    listen 80;
    server_name your.url.com;

    location / {
        include uwsgi_params;
```

```

uwsgi_pass nestegg;

proxy_redirect off;
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Host $server_name;
}
}

```

You would need to replace the contents of the server(s) in the upstream block with the internal address(es) the application(s) are hosted on. After your edits be sure to reload or restart Nginx using (depending on the system) `systemctl restart nginx`.

uWSGI To host the application itself without using the built-in server you will need uWSGI. Installing it is easy, just run `pip install uwsgi`. Configuring it is more involved. We will use this with Supervisor to manage the application.

Supervisor Supervisor is a Python “Process Control System”⁶. Using Apt or your preferred package manager run `apt-get install supervisor`. The directory `/etc/supervisor` should have been created. Within `conf.d/` we will create a new configuration file for our application.

Label the new configuration file whatever you wish but be sure to append `.conf` to the filename. The general filename is irrelevant as Supervisor will look for the program name within the configuration file’s contents for identification. In listing 5 you will a basic configuration for Nestegg.

Listing 5: nestegg.conf

```

[program:nestegg]
user=youruser
group=yourgroup
directory=/path/to/nestegg/
environment=NESTEGG_CONFIG=Config,HOME=/path/to/optional/
virtualenv
command=/path/to/uwsgi -H /path/to/optional/virtualenv -w
    nestegg:app -M -p 2 --http :5000
autostart=true
autorestart=true

```

⁶<http://supervisord.org/>

```
stdout_logfile=/var/log/nestegg_out.log
stderr_logfile=/var/log/nestegg_err.log
```

Note:

The arguments `HOME` and `-H /path/to/optional/virtualenv` are considering optional as the packages required for Nestegg can be installed system-wide.

Using a Python virtual environment is completely optional (using `virtualenv`), but ideal for systems that have may use many versions of the same packages for other services.

After saving the uWSGI configuration file we will need to update Supervisor with our changes. As root or a privileged user, execute `supervisorctl`. From Supervisor we need to run `reread`, then `update`. Our application should have already started, we can confirm this by using `status` and search for the Nestegg process. If it has not, try to start it using `start nestegg`. You can then exit Supervisor with `exit`.

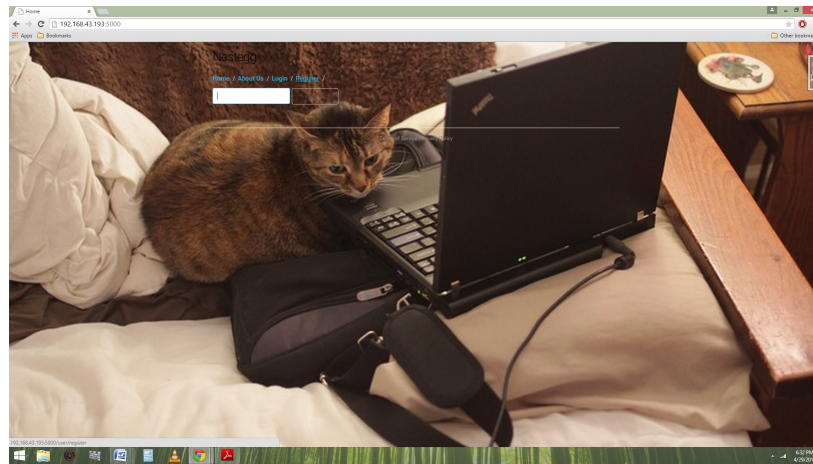
If you encounter any issues starting application via Supervisor you can review the logs we established in the configuration file or the logs for the supervisor process of Nestegg under `/var/log/supervisor/`.

2 USER'S MANUAL

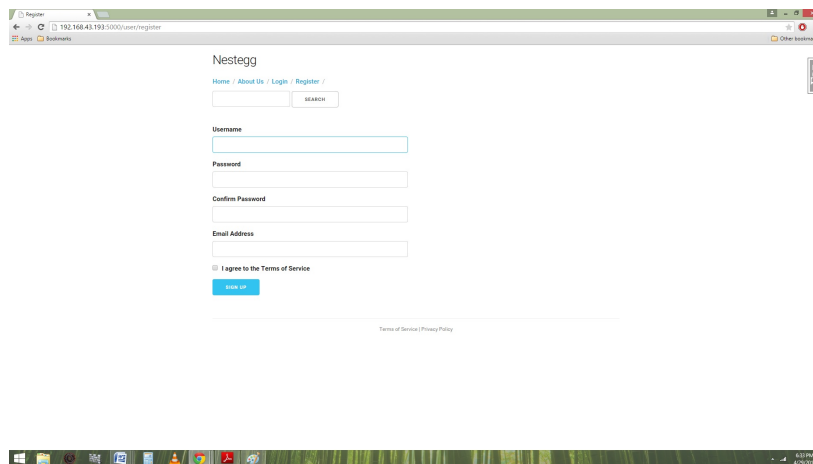
2 User's Manual

2.1 Signing Up

1. Navigate to the Nestegg homepage and click the “Register” link at the top of the page.



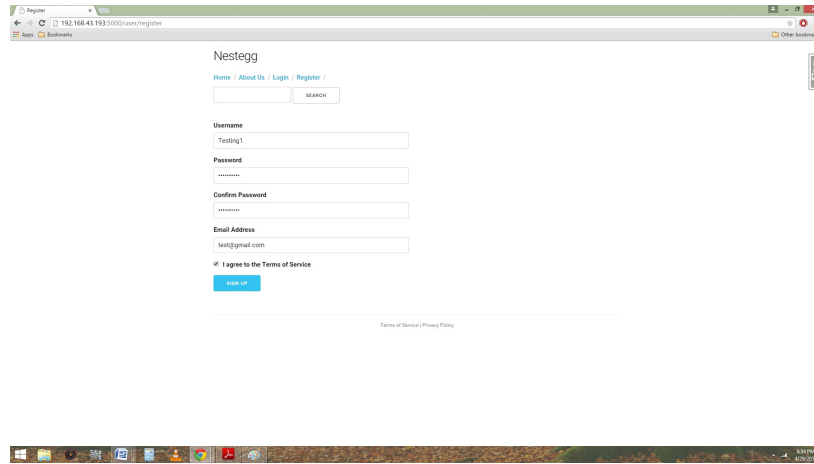
2. You will be taken to the registration page.



3. Fill in your information in the fields on the page and click the Sign Up button.

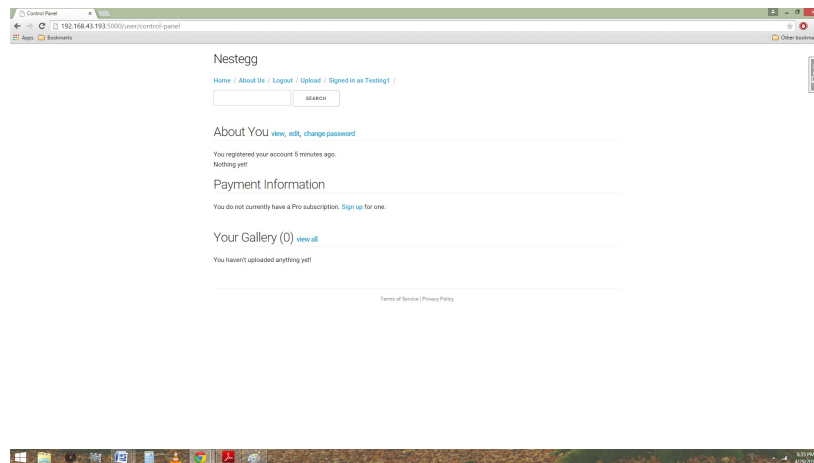
2 USER'S MANUAL

2.1 Signing Up



The screenshot shows a web browser window with the URL `192.168.43.193:5000/users/register`. The page is titled "Nestegg" and has a navigation bar with links: [Home](#) / [About Us](#) / [Login](#) / [Register](#) / . Below the navigation bar is a search bar. The main content area contains a registration form with the following fields:
 - Username:
 - Password:
 - Confirm Password:
 - Email Address:
 Below the form is a checkbox labeled "I agree to the Terms of Service" and a blue "Sign Up" button. At the bottom of the page, there is a link for "Terms of Service / Privacy Policy".

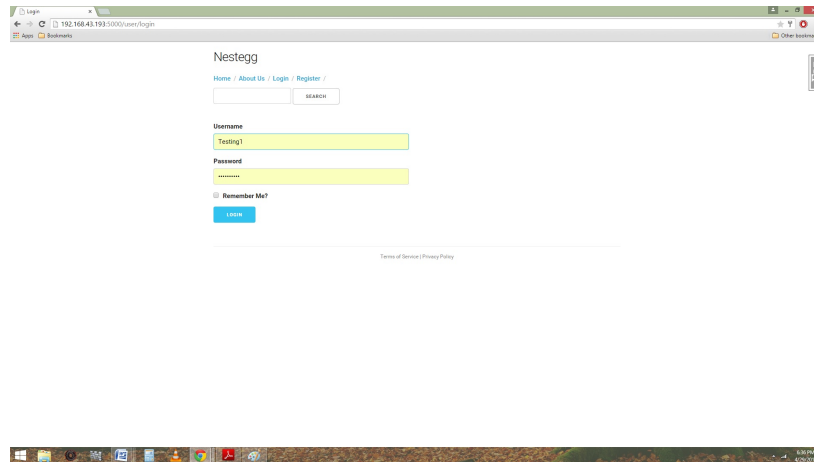
4. You will automatically be taken to your profile page. You can use the view and edit links to change the information that you have set on the site that you want other users to be able to see. You can click the change password link to change your account password. You will also be able to see anything that you uploaded by clicking the view link for the gallery.



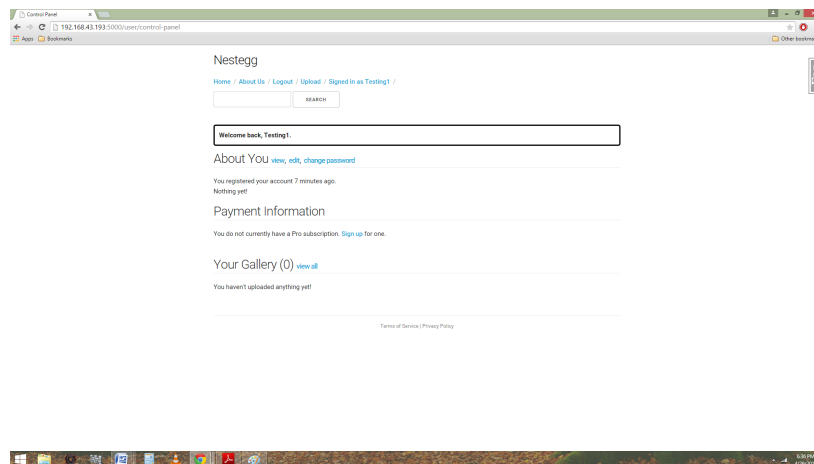
The screenshot shows a web browser window with the URL `192.168.43.193:5000/users/control-panel`. The page is titled "Nestegg" and has a navigation bar with links: [Home](#) / [About Us](#) / [Logout](#) / [Upload](#) / [Signed in as Testing1](#) / . Below the navigation bar is a search bar. The main content area contains the user's profile information:
 - About You: [view](#), [edit](#), [change password](#)
 - Payment Information: [Sign up for one](#)
 - Your Gallery (0): [view all](#)
 At the bottom of the page, there is a link for "Terms of Service / Privacy Policy".

2.2 Pro Membership

1. Click the login button on the top navigation bar of the site and enter your login information.



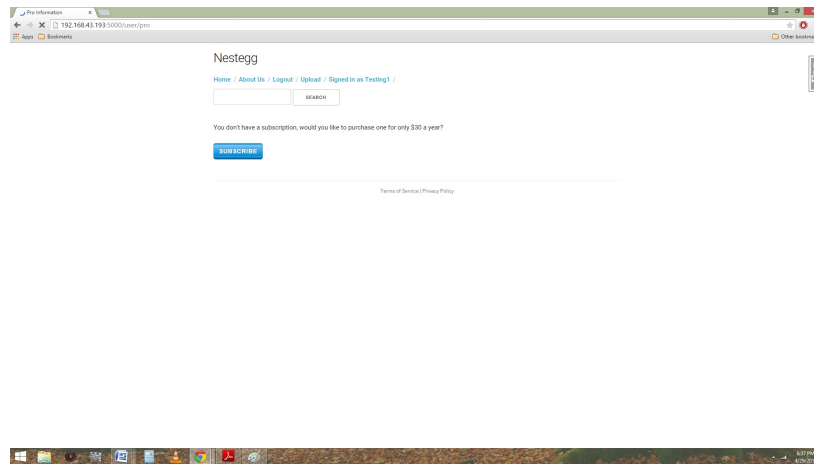
2. After signing in you will be taken to your profile page. Under payment information you will see the phrase “You do not currently have a Pro subscription. Sign up for one.” Click “Sign Up”. You will then be taken to the subscription page.



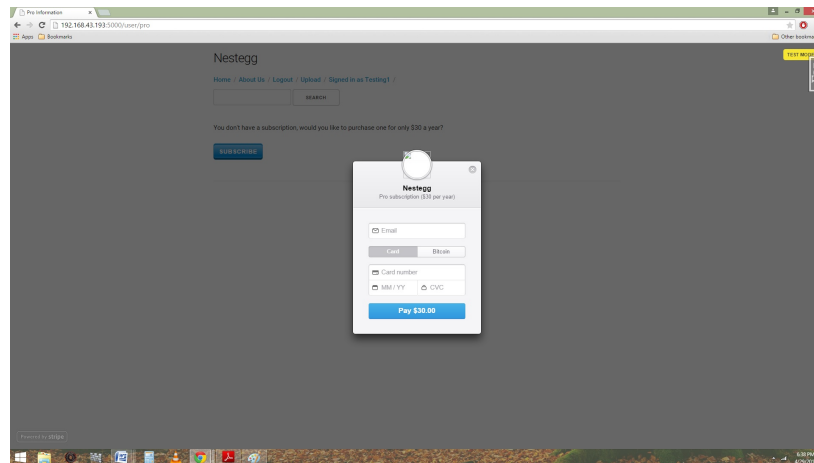
3. After you have reached the subscription page, click on the “Subscribe” button.

2 USER'S MANUAL

2.2 Pro Membership



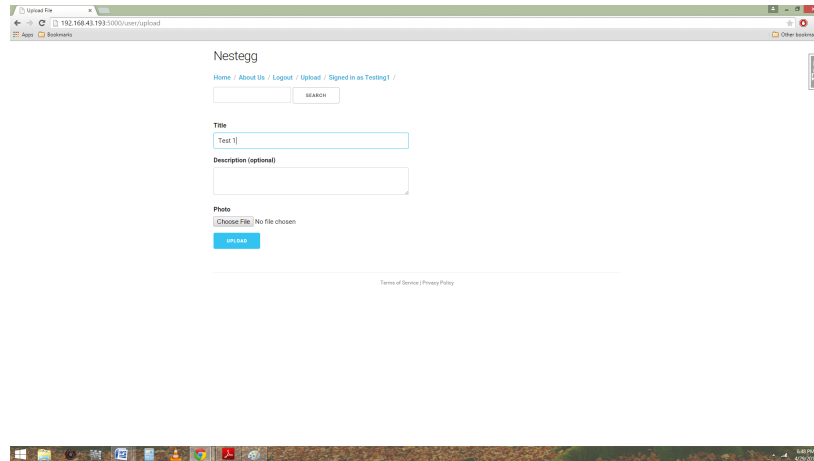
4. After clicking the subscribe button you will be prompted to enter your financial information.



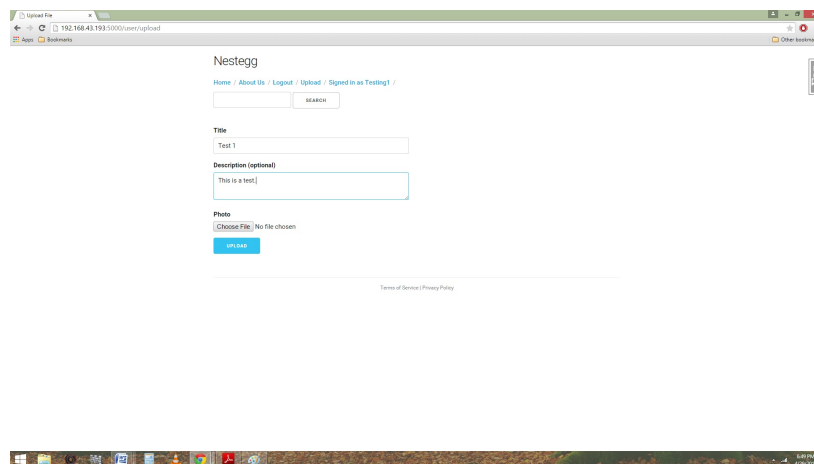
5. After you enter your payment information and click the “Pay” button you will be taken to the subscription confirmation page. This will also tell you the date that your subscription is set to renew automatically.

2.3 Uploading a File

1. While you are logged into the site click on the Upload link at the navigation bar at the top of the page and you will be taken to the upload page. Enter a name for the file that is being uploaded.



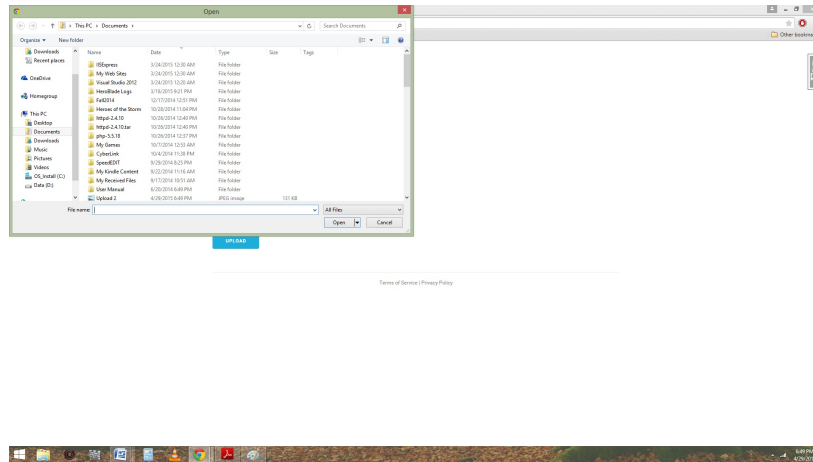
2. Enter a description of the file and any other information that you would like to be displayed when someone views the file in the description box.



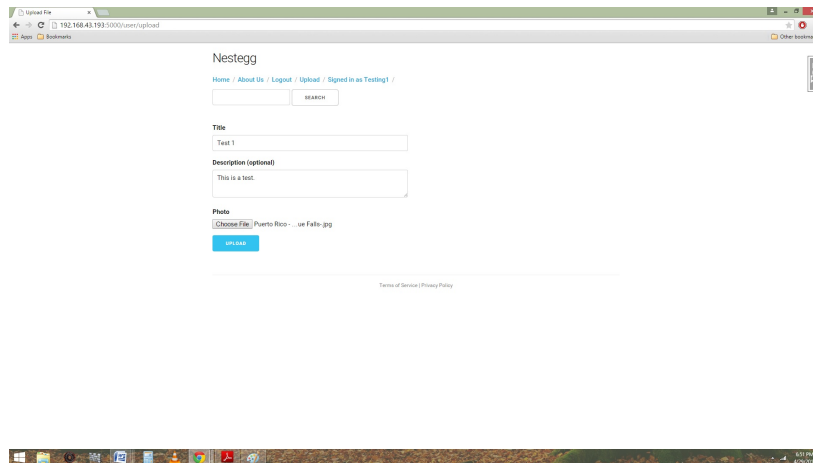
3. Click the choose file button and a window will pop up that will allow you to choose which file you would like to upload.

2 USER'S MANUAL

2.3 Uploading a File



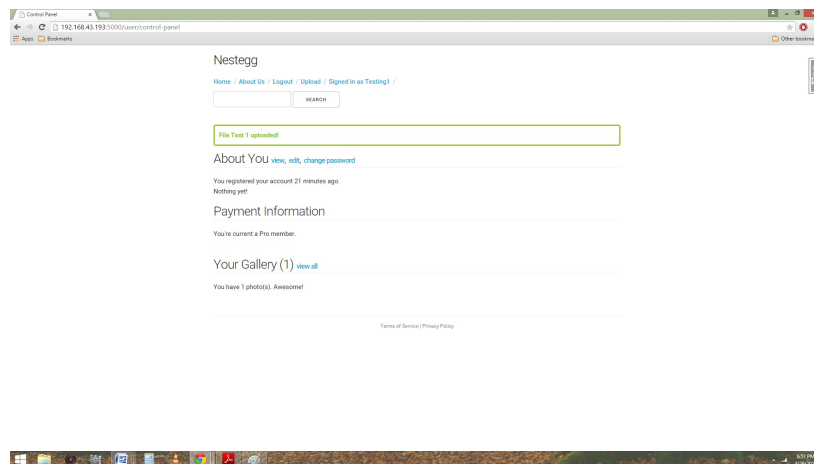
4. After choosing the file that you would like to upload click on the Upload button.



5. After uploading the file you will be taken back to your profile page and you will be able to click on the gallery link to be able to see the file and any others that you have uploaded.

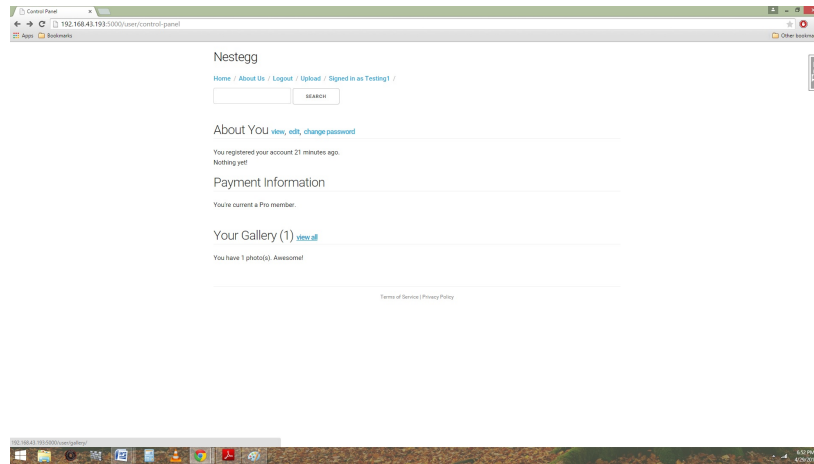
2 USER'S MANUAL

2.3 Uploading a File

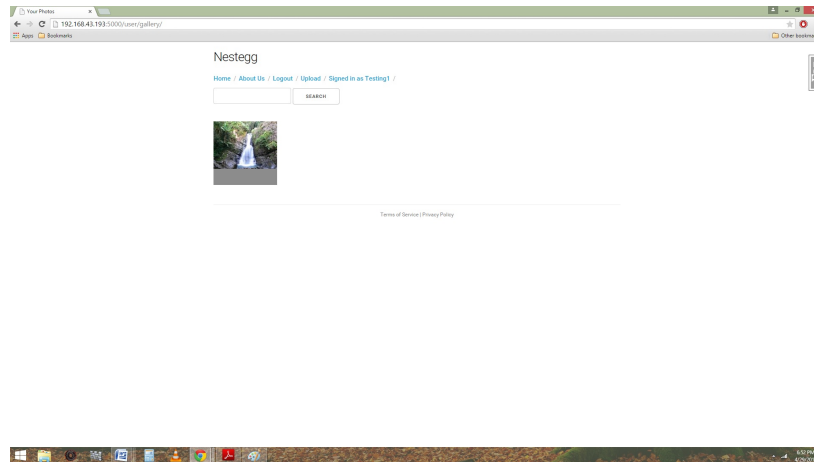


2.4 Editing & Deleting a File

1. From your profile page click on the “View All” link next to your gallery.



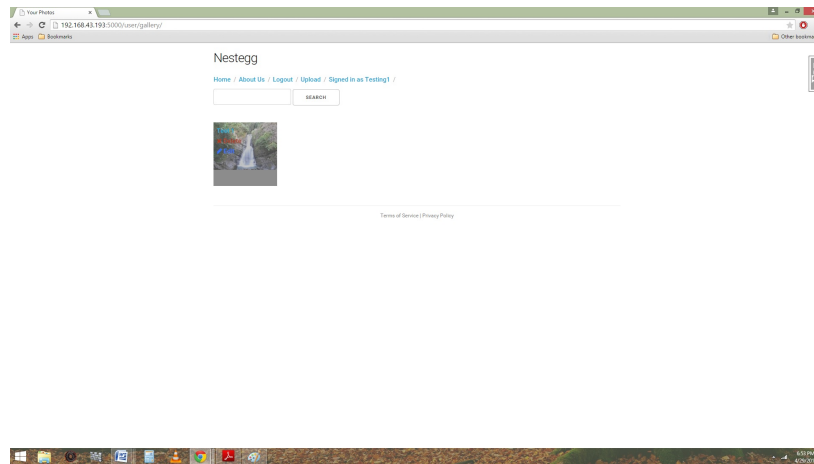
2. You will be taken to your gallery page and all of the files that you have uploaded will be displayed.



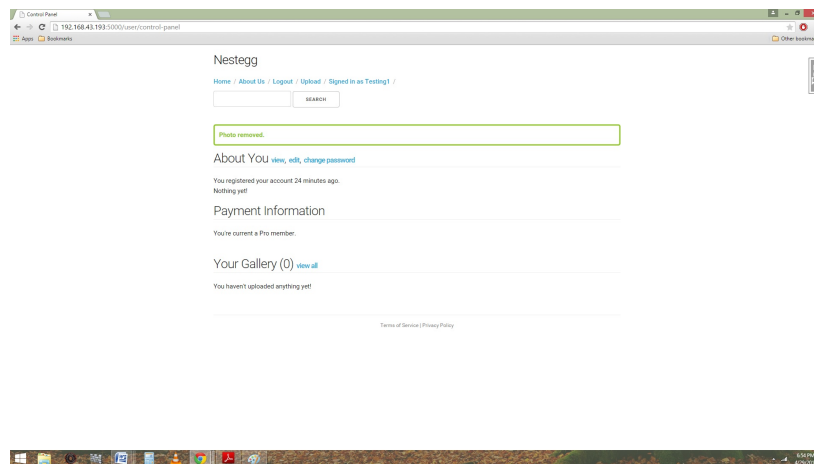
3. Mouse over the file you wish to edit and the option to either delete or edit will appear over the file (pressing delete will remove the file).
4. Pressing edit will take you to the edit page for that particular file.

2 USER'S MANUAL

2.4 Editing & Deleting a File

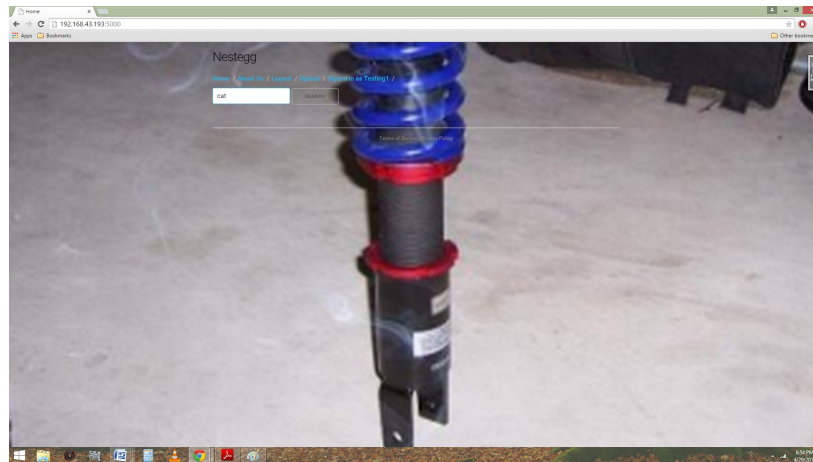


From here you can change the file's title or description. When finished press "Save Changes".

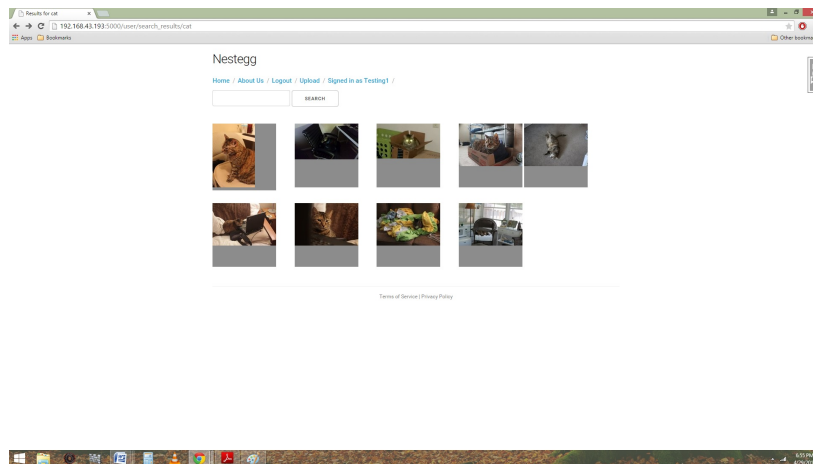


2.5 Searching for Images

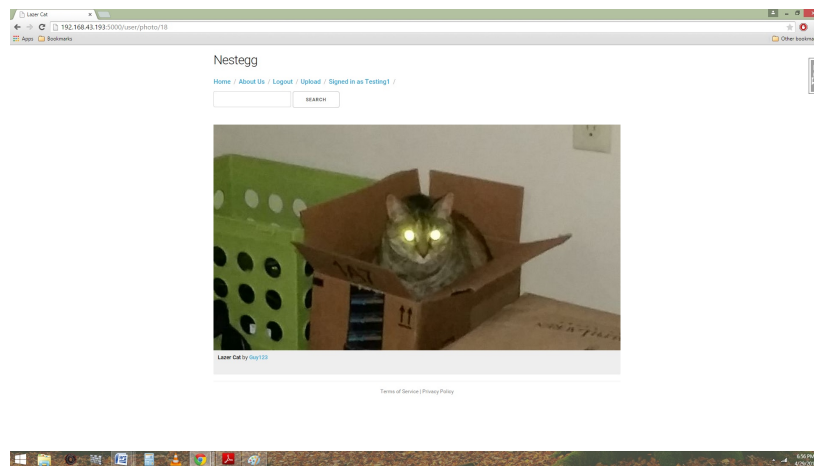
1. From any page of the site you may enter a search term in the search field at the top of the page.



2. After clicking the search button you will be shown all of the files that match the search term.



3. Click on any file to see a larger view of the file. The description, title, and owner of the file will be displayed below.



A DATABASE SCHEMA

A Database Schema

The **User** and **Files** exist in a Zero-to-Many relationship. Where the User can have none or many Files.

