

COSC349 Assignment 1

Group:

- Marcus Anderson (7049664)
- Nathan Hollows (6887316)

Nugget Watch

Nugget Watch aims to be the Wikipedia of chicken nugget prices. Nugget Watch has prices listed from the largest ready-to-eat chicken nugget suppliers in New Zealand. Nugget Watch also allows customers to rate and review their nuggets as well as upload images with their reviews.

Nugget Watch was in development before the COSC349 assignment was made public and was built to put into practice design patterns from COSC344. To demonstrate the development of an application we have implemented “featured reviews”. This required making changes to both the backend, and frontend of the website as well as the database structure and dummy data.

Build System

Configure and Build

Optional before build: Set secure database password:

Replace all instances of `insecure_db_pw` in `installs/build_database.sh` and `www/app/config/config.php` with your password of choice.

Replace all instances of `insecure_mysqlroot_pw` in `installs/build_database.sh` with a new root password for the database.

Build virtual machines:

```
vagrant up
```

Build time:

The initial build for the three virtual machines and installation of `ubuntu/bionic64` took around 16 minutes

Subsequent builds took about 8 minutes

Other notes:

There is a reported warning in which we are running a piece of software ‘Composer’ as a root super user. Composer needs to be run as a member of the vagrant user group. Considering that, this is expected behaviour.

The virtual machines

Nugget Watch is deployed over three virtual machines. Each virtual machine runs `ubuntu/bionic64`, which is downloaded once on the first build - subsequent builds use the same box file. Each webserver then downloads, compiles, and installs it’s own required software, all of which is open source.

webserver is the virtual machine responsible for serving webpages. This virtual machine runs Apache to display our web pages that are written in PHP and the Phalcon framework. **webserver** installs apache; php and useful libraries such as imagick for initial image checks; Phalcon, an PHP framework as a C-extension; and Composer, a PHP library manager. This server is preconfigured to interact with **dbserver** and is able to read and write reviews, chains, pricing, messages, etc.

dbserver runs the database using MySQL. On initialisation this executes `installs/nugget.sql`. This server communicates with **webserver** to display Nugget Watch data.

imgsever is responsible for compressing and resizing images. This virtual machine uses ImageMagick for resizing and cron for scheduling. This server can be developed further to move and resize images for a more responsive web design, and to move original full size files to longer term storage instead of deleting originals. This server works on images within `www/app/temp/` and `www/public/img/` and saves the webserver from performing these expensive operations.

How to use Nugget Watch

Register an admin user

Visit <http://192.168.2.11/register> to create an admin user then login with the same details (you will be redirected after successfully creating an account).

Administrating Nugget Watch

<http://192.168.2.11/prices> Is where you can add prices for the different chains as well as managing the existing prices. You can delete prices from Nugget Watch if they are out of date or you can confirm that the prices are correct. There is

a timestamp which will show you when you last confirmed that the prices were correct.

<http://192.168.2.11/messages> Click a name to read the message. Messages cannot be responded to from the web interface as there is no email server set up. Messages may be deleted (this sets a flag in the database to hide the message). These messages come from the contact form located on the front end <http://192.168.2.11/contact>

<http://192.168.2.11/reviews> Reviews may be deleted by admins. Other reviews submitted by the same person may also be found by clicking the **author** button.

Reviewing

<http://192.168.2.11/review> Users can review their nuggets that they had from the provided chains. Reviews require the users name and email address. Reviews also need all of the ratings to be filled in as well as comments to be a valid review. The user can upload a picture of their nuggets if they want to.

Perusing

<http://192.168.2.11> The home page of Nugget Watch is where all your dreams come true. The main chains are displayed with the deals that they offer, their overall ratings and a link to the reviews of the chain. The deals are ordered by the best value per nugget.

<http://192.168.2.11/about> Tells the origin story of Nugget Watch as well as some of the most frequently asked questions that the Nugget Watch team encounters

<http://192.168.2.11/chains> The chains page deals with the reviews for Nugget Watch. You can go explore into each chain and read the reviews from the users of Nugget Watch. The reviews are ordered by the date when the review was made, the most recent reviews are at the top.

<http://192.168.2.11/contact> This is where you can contact the Nugget Watch team with any issues that are encountered.

Useful Modifications

This project has been deployed and developed so that all components are separated. The host user is able to modify the web frontend/backend within the `www/` folder, and this folder is shared between all virtual machines. `installs/nugget.sql` is responsible for holding the database structure and inserting dummy data. `installs/crontab` contains the commands and schedule for the image server.

```
- www/
  - app/
    ;; Backend logic using MVC pattern
  - public/
    ;; Public resources such as images, favicons, css
- installs/
  - nugget.sql ;; Database structure and initial dataset
  - crontab    ;; Schedule and commands for imgserver
```

Extend the API

- GET Nearest nugget
- GET Best nugget
- GET Nugget prices
- GET / POST / PUT / PATCH / DELETE Reviews

Extending the API would enable apps to be built on top of Nugget Watch. Potentially a nugget finder and review app.

Implement map functionality

Nugget Watch could show a map on the website. A service could be built to scrape the chains for their locations (if their terms of use permit this) and build a map on top of Google Maps API, or anything similar.

Rating Reviews

Thumbs up / thumbs down for each review. Reviews could be rated either 'sweet' or 'sour' and could be sorted from best review to worst review. Once you have finished creating the sweet/sour review rating you would only need to restart the database.

Image checking

Nugget Watch should have a way to check what the contents of the images uploaded are, making sure that no profanity is uploaded to the site. This could be accomplished by setting all new images with reviews to unapproved and requiring admin approval.