Web Programming Assignment 2

Build a product review application using MVC framework

Submission by:

Adam Williams and Ryan Melim

**Content index**

**1 Site Access and general information - Page 3**

**1.1 Site Access**

**1.2 Cloud site access**

**1.3 Permissions**

**1.4 Bugs**

**1.5 General information**

**2 Part A File structure, Functionality and Cloud - Pages 4 to 7**

**2.1 Overview - Page 4**

**2.2 File structure - Page 4**

**2.3 Model - Page 4**

**2.4 Controller - Page 5**

**2.5 View - Page 6**

**2.6 Cloud storage solution - Page 7**

**3 Part B MVC Framework Comparisons - Pages 8 to 9**

**3.1 Overview - Page 8**

**3.2 Django vs Web2py - Page 8**

**3.3 Laravel vs Web2py - Page 8 to 9**

**3.4 Conclusion - Page 9**

**4 Appendix and bibliography - Pages 10 to 13**

**4.1 Test log - Page 10**

**4.2 WP Assignment 2 Self-Assessment Sheet - Pages 10 to 11**

**4.3 References and bibliography - Pages 11 to 13**

**1 Site Access and general information**

**1.1 Site Access**

When conducting tests of the web server you can use these three accounts. The first one will give you user access, the second will provide admin access and the third will give admin access along with the ability to make other users admins and admins regular users.

* User username: Rocko

User password: 123456

* Admin username: Rabbitstew

Admin password: admintest

* Super admin username: Wall hall

Super admin password: superadmin

**1.2 Cloud site access**

Port forwarding is used for cloud access so go to the following address in order to access the site:

[**http://86.27.227.119:8000/assignment/default/index**](http://86.27.227.119:8000/assignment/default/index)

If there is no response when trying to load the above page then email Adam Williams on i7648171@bournemouth.ac.uk as he may have forgotten to turn the server on.

86.27.227.119 is the external IP address of the router the server is connected to and 8000 is the port Web2py uses for communication.

**1.3 Permissions**

A regular user can view reviews, view products, search products, make reviews, edit their reviews and delete their reviews.

An admin can do everything a regular user can along with being able to: delete anyone’s review, add products, edit products and delete products.

A super admin can do everything an admin can along with being able to promote regular users to being admins and demoting admins to being regular users.

**1.4 Bugs**

For reasons unknown the Web2py login system will temporarily not work with one browser but work with another, after a time it will switch so that the accepted browser is denied and the denied browser is accepted.

This was tested with Chrome, Chromium and Firefox with no consistent cause but seemed to happen more often when using an admin account and not using firefox, if you have issues with logging in or if you are automatically logged out then please try another browser.  
  
Sometimes after making a review, deleting a review or at random the user will be logged out, this stopped happening after a while but it may happen again.

Sometimes random images will not load, refreshing resolves this.

Web2py randomly cleared the auth\_group table at one point, no apparent cause.

**1.5 General information**

Some custom CSS was used for the assignment, this can be found in static/css/assignment.css.

Minor modifications were made to the bootstrap scheme using !important CSS, an example of this is the response menu (menu bar) colour being changed to be grey instead of black.

**2 Part A File structure, Functionality and Cloud**

**2.1 Overview**

This section will provide an overview of the file structure, functionality for the Model, View and Controller sections within our Web2py site, brief descriptions on how they interact with each other and an explanation of our cloud storage solution.

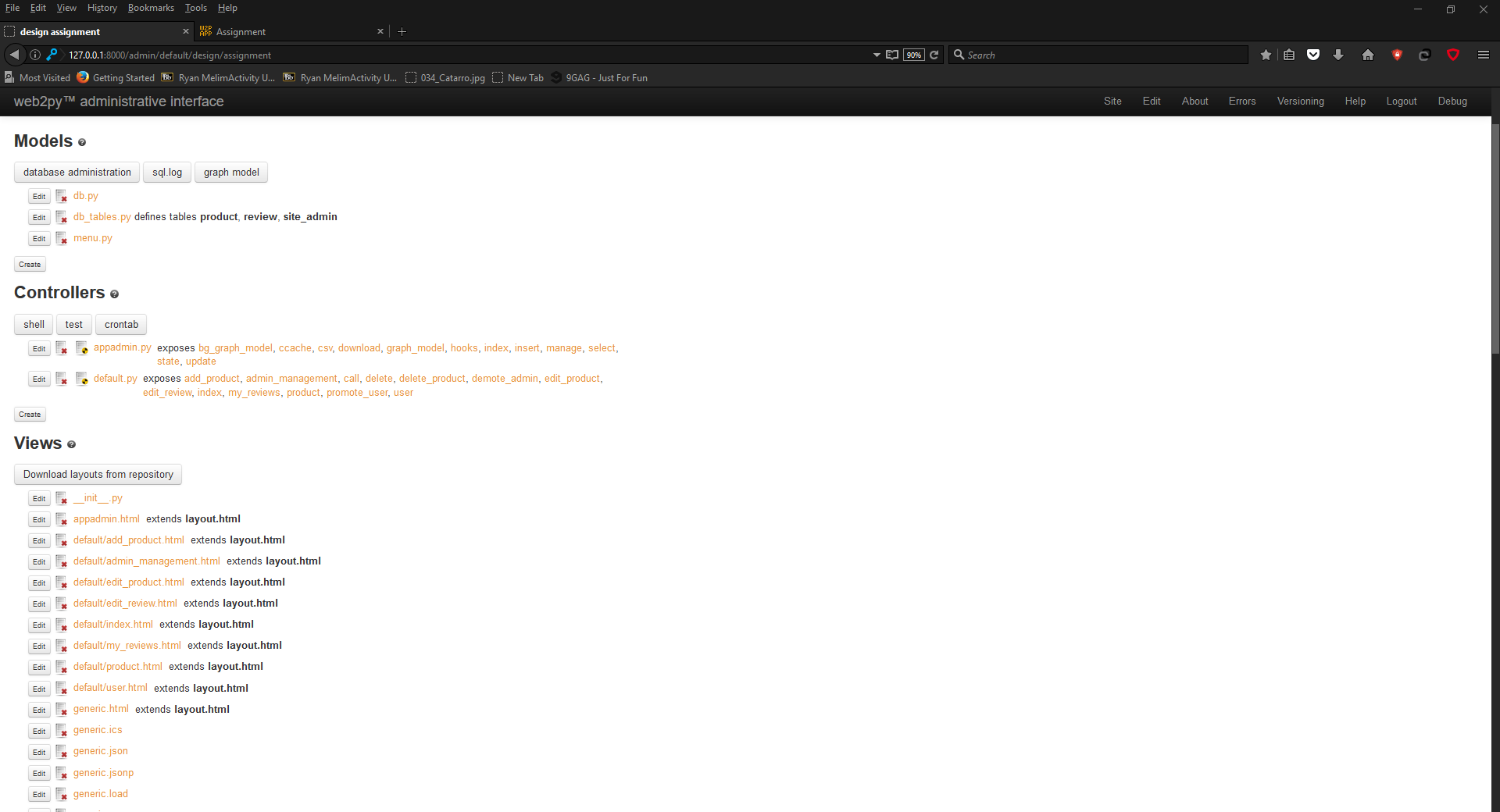
**2.2 File structure**

The file structure consists of Model, View and Controller, which allows developers to separate data into a clear and easily manageable system for the creation of a website simply and to a high standard.

**2.3 Model**

The model section of Web2py is where tables and menus are defined, in our model section we have defined product, site\_admin and review tables, along with modifications to db.py so that usernames are used and modifications to menu.py to change the response menu.

Tables are stored in db\_tables.py instead of db.py as Web2py has shown unexplained behaviour resulting in data being lost from db.py, so isolation of the tables from db.py mitigates this issue.



**Main tables**

product: Data for products.

review: Data for user reviews, foreign keys link this table to the auth\_user and product tables.

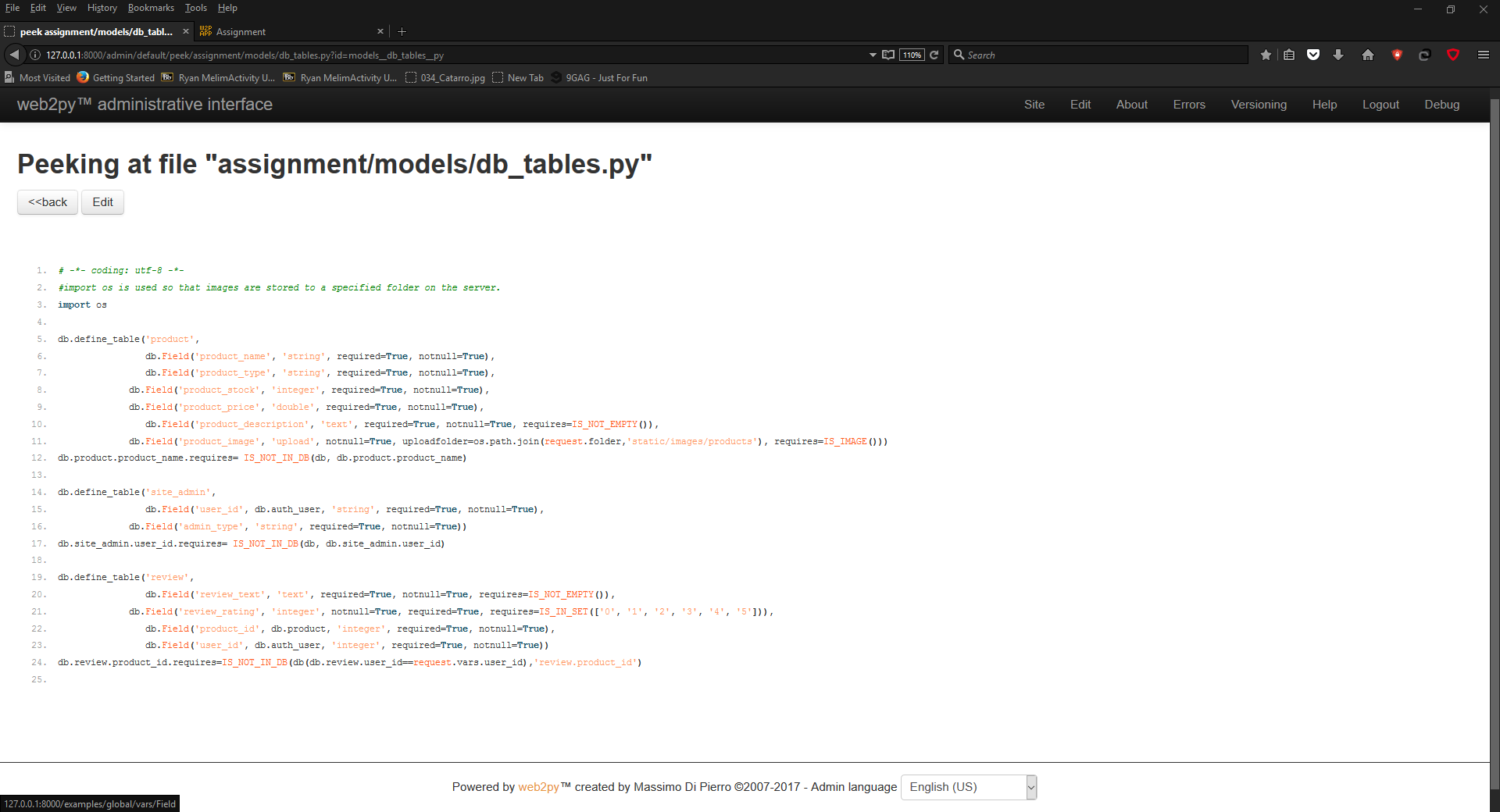
auth\_user: User data for Web2py’s user system, has been modified to allow for usernames.

site\_admin: Web2py randomly cleared the auth\_group table so we are not confident in using it for storing group information, so site\_admin was made for admin group data.

**Code explanation**

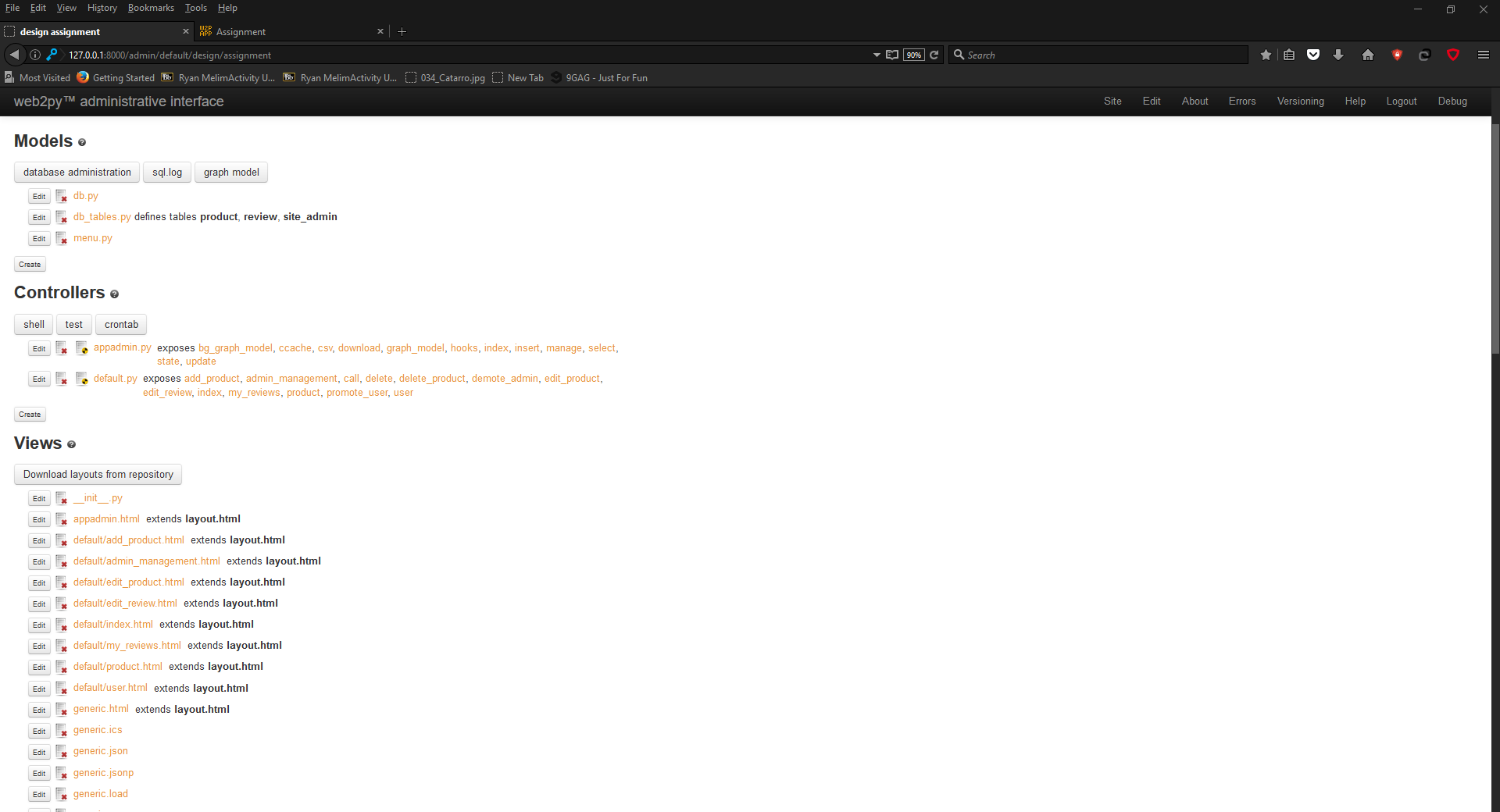
Below is code for the product table, it shows the table definitions with conditions that mean no field can be empty and so that product\_image will point to an existing location for images.

This code would be used in conjunction with controller and view code to establish the visual representation of tables on a page.



**2.4 Controllers**

The controller contains most of the functions used to correlate view and model data, i.e. constructing database queries from partial queries in functions and user input from webpages, along with providing dictionaries for views.



**Main functions:**

add\_product: Adding products to the database using an SQLFORM and providing a response based on user input, connects to add\_product.html.

admin\_management: Gets user and admin data if the user is a super admin, returns data to admin\_management.html.

demote\_admin and promote\_user: callable functions that make an admin a user or a user an admin, receives user ID by args and redirects to admin\_management.html.

edit\_product: Editing existing product using an SQLFORM, connects to edit\_product.html.

edit\_review: Editing existing review using an SQLFORM, connects to edit\_review.html.

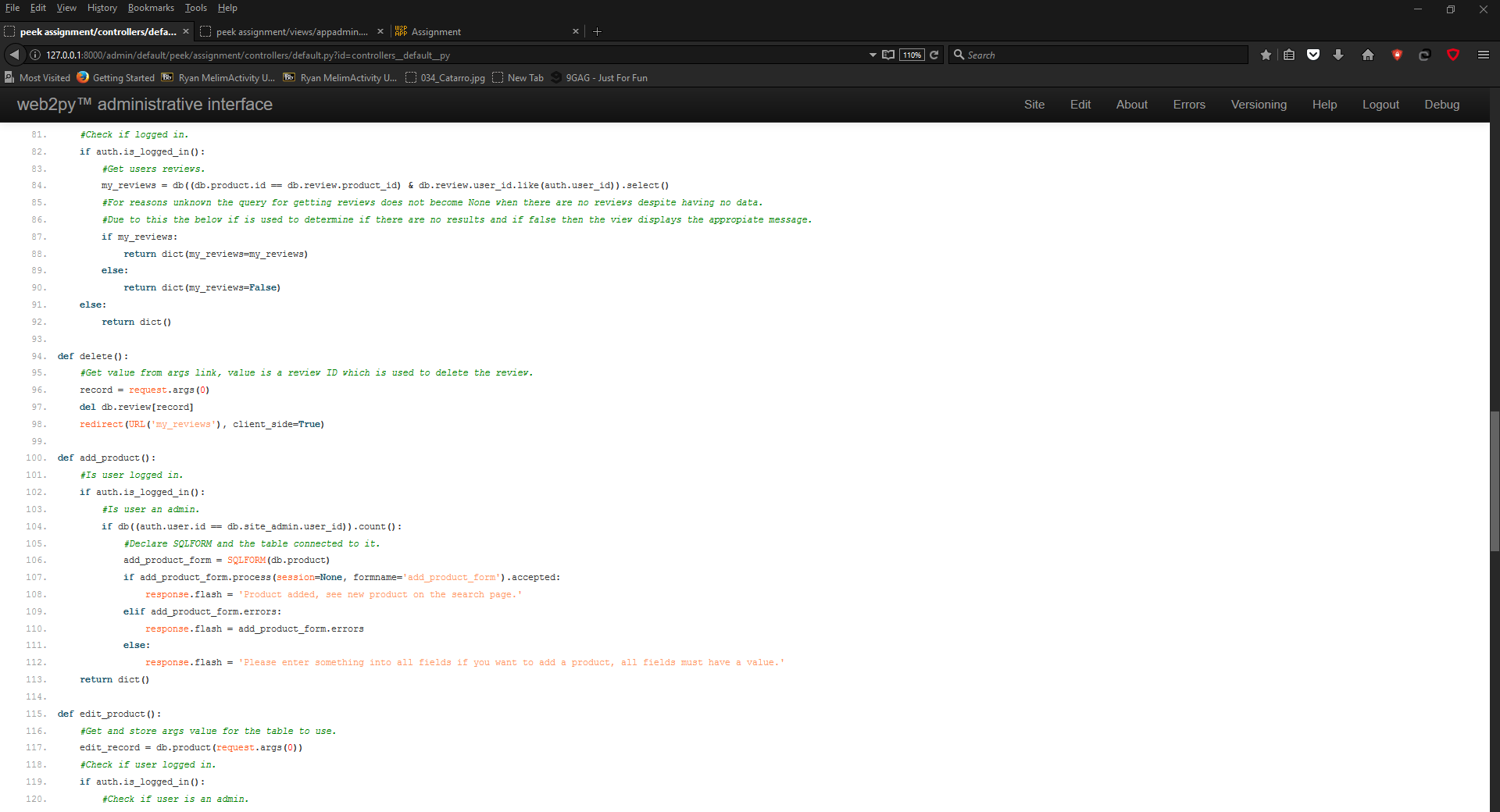
index: Search for products using an SQLFORM and gets 3 random reviews, returns to index.html.

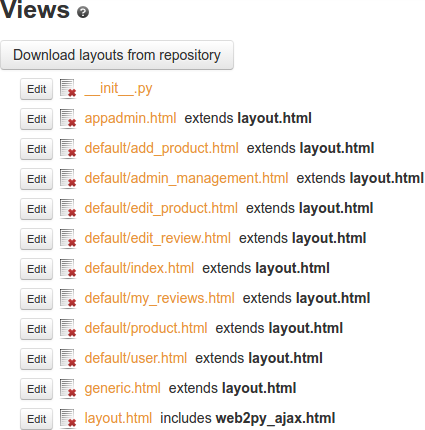
my\_reviews: Get data for reviews the user has made, returns to my\_reviews.html.

product: Get product data, product reviews and for making reviews using an SQLFORM for a product with the ID from args, returns to product.html.

**Code explanation**

Below is the controller code for adding a product, it checks if the user an admin before declaring the form for the view to use, if the add is successful the admin is notified about the submit, if not then an error is shown.



**2.5 Views**

The view component determines how data will be presented on a page, usually formatted data is returned from controller functions or is formatted in the view.

**Main pages:**

add\_product: Page with form and elements for adding a product if admin.

admin\_management: Page for promoting or demoting users if a super admin.

edit\_product: Page with form and elements for editing existing products if admin.

edit\_review: Page with form and elements for editing existing reviews if logged in as the user who made it.

index: Page with form and elements for searching for products and seeing three random reviews.

my\_reviews: Page with elements that shows the reviews the user has made.

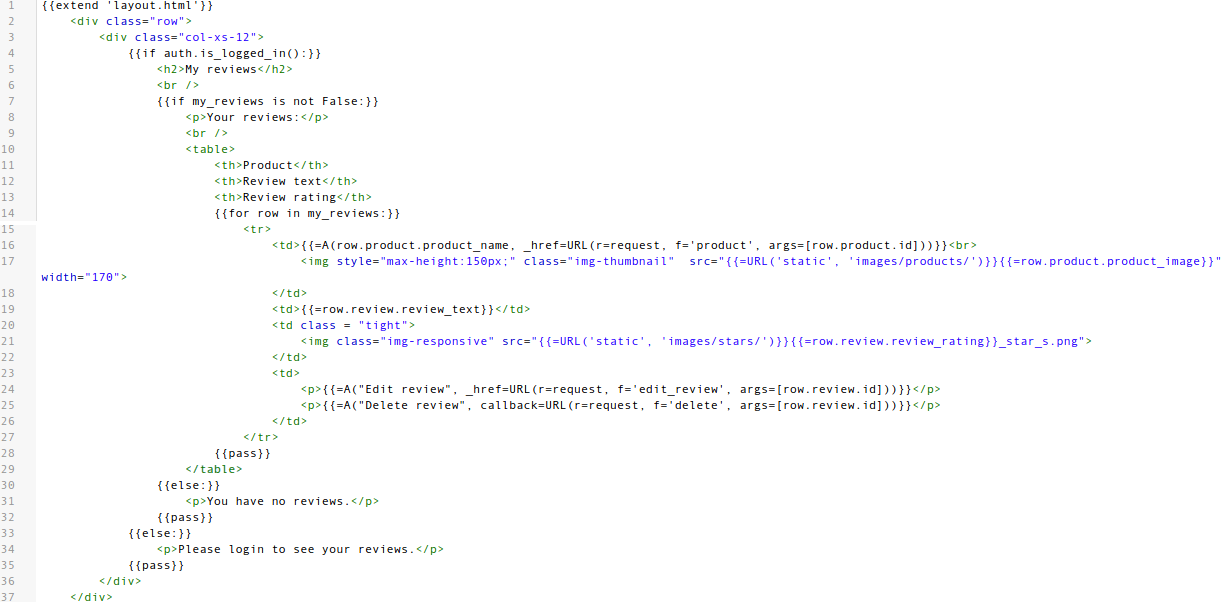
product: Page with elements and form for seeing product information, its reviews and making a review.

layout: Template page that other pages get layout and structure from.

**Code explanation**

The below is code for my\_reviews.html, which receives the user’s reviews from the controller as Rows data and iterates them as HTML table rows.

It also has links for each review to delete and edit it.



**2.6 Cloud storage solution**

For our cloud solution we have Web2py running on a Windows 2008 R2 server computer.

To connect over the internet we have port forwarded the port 8000 (Web2py communication port) so that any device with an internet connection and a web browser can access the site and admin interface if they have the password.

Additionally, the ports for remote desktop access have been forwarded for to the OS over the internet and we have all our project work in a google drive folder which is set to synchronise with a folder on the server, this allows for file transferring to and from the server along with a backup of work.

The reason we have used this could solution is so that we have complete control over the cloud operations and functionality along with 24/7 access to the server and its files and being able to upgrade server hardware and software if more processing power or features are needed.

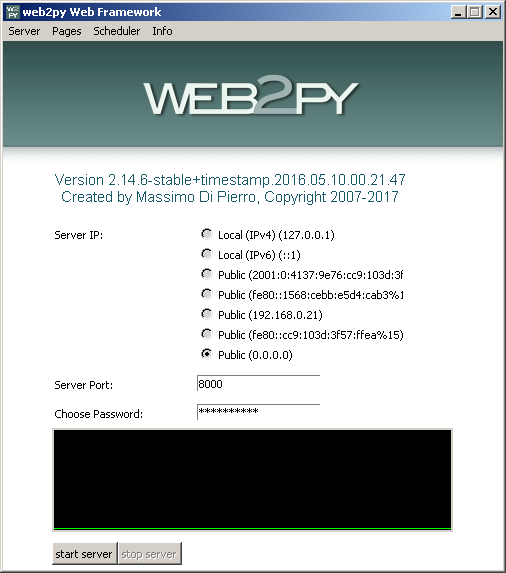
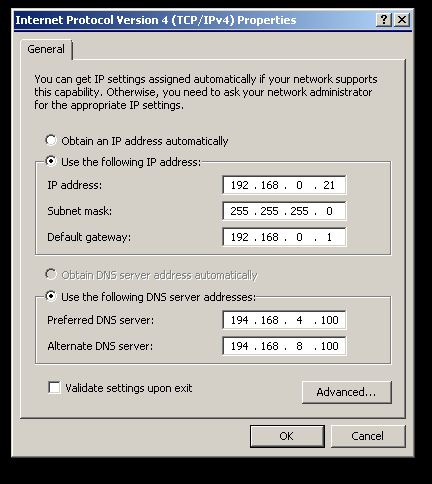
For site access go to this address:[**http://86.27.227.119:8000/assignment/default/index**](http://86.27.227.119:8000/assignment/default/index)

Shown in the images below is the port forwarding configuration on the server, router and in Web2py:

Router



Server Web2py



**3 Part B MVC Framework Comparisons**

**3.1 Overview**

With the way websites are developed nowadays a framework is almost a necessity, for our assignment we have been using Web2py but there are many alternative MVC frameworks available with both positive and negative attributes, like Django and Laravel.

The table below contains a brief overview of each MVC framework.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Django** | **Web2py** | **Laravel** |
| **Language** | Python | Python | PHP |
| **Framework base** | Django | Laravel | Laravel |
| **Description** | Free open source full-stack framework for rapid development. | A Framework For Web Artisans | The Web framework for perfectionists with deadlines |
| **Difficulty** | Advanced / Intermediate | Beginner / Advanced | Intermediate / Master |
| **Community** | Excellent | Very Good | Good |

The capabilities of the three are fairly similar in terms of design but have their differences:

Web2py and Django allow developers to be more flexibility in terms of design compared to Laravel.

Laravel allows encapsulation of its data to protect from unauthorized users whereas Web2py and Django do not.

Now if you compare Web2py to these two MVC frameworks you can see that they all have advantages and disadvantages over one another, for example, if you compare Web2py to Django you can see that they are very similar in their functionality and capabilities, but Web2py is simpler and has a quicker learning curve than Django as Web2py is designed to be compact and have good functionality available from the get go.

**3.2 Django vs Web2py**

Another advantage of using Web2py over Django is that Django support only version (1.8) and higher, while Web2py support all previous versions.This means that each variation of the Web2py system’s can interact without conflicts and that new updates won't break previous versions

A disadvantage of using Web2py over Django is that the Django community is more active than web2py and some of Web2py’s documentation is outdated, So if you have a query or need help on a specific issue you are more likely to find help and a solution from Django resources or an active Django user.

**3.3 Laravel vs Web2py**

If you compare Web2py to Laravel they have some similarities and differences, for example, the main difference is between them is Laravel is coded in PHP while Web2py is Python based.

PHP has is more accepted in terms of web development and is more widely used than Python which means that there is more likely to be information available to help solve issues that arise.

But Web2py includes frontend and backend scaffolding by default which means that developers can start designing their site’s more efficiently and start to implement functionality to the client side a lot faster.

Lastly, one other advantage of Web2py over Laravel is that Web2py from the start gives the developers the ability to immediately start writing in code so websites can be designed quickly and with a good amount of functionality.

**3.4 Conclusion**

In conclusion Web2py is a good full stack framework with a quick learning curve and a lot of potential to expand but as it is currently there are some things that other frameworks do better for example Django has better community and resources for problem solving, while Laravel has more features for advanced users such as unit-testing which allows users to thoroughly test their code for bugs and errors.

But really it comes down to user preference and the suitability of the task for a specific framework, for instance if you are familiar with python and want something quick and easy to use then Web2py could be beneficial for you whereas if you're more familiar with PHP then Laravel may be better to use.

**4 Appendix**

**4.1 Testing log**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Description** | **Expected**  **Outcome** | **Outcome** | **Solution** | **Solution achieved** |
| Login | Test to see if login to the website is successful | Login successful | Login was successful but occasionally would not work | If the login fails try on different browser |  |
| Review product | As a user or admin you should be able to review products | User review is added to product page | Review successfully added | Not needed |  |
| Image icons loads | Product Images loads when page opens | The images load correctly | Majority of the images loaded | Reload browser/ change browser |  |
| User view | Logging in as a user and gaining access to user specific stuff | When logged in they can see user specific options | Displayed user view options | Not needed |  |
| Admin view | Logging in as a admin and gaining access to admin specific stuff | When logged in they can see admin specific options | Displayed admin view options | Not needed |  |
| Edit products | When the edit product button is pressed it should allow you to input updated details and save them | The new details are saved | Details were saved | Not needed |  |
| Delete products | Delete product should remove the product from the database | Product should be removed | Product was removed | Not needed |  |
| Add products | When the add product button is pressed it should allow you to input details about a product and save them | New product is created | Product was created | Not needed |  |

**4.2 WP Assignment 2 Self-Assessment Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Yes (completed)** | **Partially completed** | **No (not submitted)** |
| **Report** | | | |
| MVC file structure |  |  |  |
| Cloud solution design |  |  |  |
| Evaluation/comparison |  |  |  |
| **System** | | | |
| Product Management functions (Search, Edit, Add, Delete) |  |  |  |
| Login functionality |  |  |  |
| Login restrict menus |  |  |  |
| Reviews |  |  |  |
| Reviews with user names |  |  |  |
| Cloud solution implementation |  |  |  |

**4.3 References and bibliography**

KillerX~enwiki. 2006. Django (web framework). [ONLINE] Available at: <https://en.wikipedia.org/wiki/Django_(web_framework)>. [Accessed 4 April 2017].

Massimo Di Pierro. Web2py Chapter 1. [ONLINE] Available at: <http://www.web2py.com/book/default/chapter/01#Model-View-Controller>. [Accessed 1 April 2017].

web2py vs. Laravel vs. Django comparison | vsChart.com. 2017. web2py vs. Laravel vs. Django comparison | vsChart.com. [ONLINE] Available at: <http://vschart.com/compare/web2py/vs/laravel/vs/django-framework>. [Accessed 30 March 2017].

Rating star images are modified from:

Crissov, (2014), ijSS5 [ONLINE]. Available at: <http://ux.stackexchange.com/questions/67785/are-stars-a-good-rating-system>, <https://i.stack.imgur.com/ijSS5.png> [Accessed 22 March 2017].

Mirror image

THE FOREST & CO, (2015), original\_thousand-leaves-mirror [ONLINE]. Available at: <https://www.notonthehighstreet.com/theforestandco/product/thousand-leaves-mirror>, <http://cdn.notonthehighstreet.com/system/product_images/images/001/643/300/normal_thousand-leaves-mirror.jpg> [Accessed 22 March 2017].

Wall image

NA, (2016), wall [ONLINE]. Available at: <http://www.allwidewallpapers.com/southern-border-fence-mexico/c291dGhlcm4tYm9yZGVyLWZlbmNlLW1leGljbw/>, <http://grantjkidney.com/wp-content/uploads/2016/09/maxresdefault-1038x576.jpg>

[Accessed 22 March 2017].

Web2py image

Christian Burprich, (2016), logo\_db [ONLINE]. Available at: <http://www.web2py.com/init/default/download>, <http://www.web2py.com/init/static/images/logo_db.png> [Accessed 22 March 2017].

Ryzen CPU image

NA, (2017), amd-ryzen-cpu-lineup\_01 [ONLINE]. Available at: <http://digiworthy.com/2017/01/29/no-amd-6-core-ryzen-cpu-but-the-8c8t-parts/>, [https://i0.wp.com/digiworthy.com/wp-content/uploads/2017/01/amd-ryzen-cpu-lineup\_01.jpg](https://i0.wp.com/digiworthy.com/wp-content/uploads/2017/01/amd-ryzen-cpu-lineup_01.jpg?resize=752%2C440)

[Accessed 23 March 2017].

Cheese and baguette image

Pret A Manger, (2015), Products480-440x420-18308 [ONLINE]. Available at: <https://www.pret.co.uk/en-gb/480-baguettes-wiltshire-cured-ham-&-greve-cheese-baguette.aspx>, <https://az714505.vo.msecnd.net/3bc76f904ca74f2f9254e0bc48b4df0b/Images/Products480-440x420-18308.jpg> [Accessed 23 March 2017].

Boeing image

Tim Ockenden / PA Wire/PA Photos, (2017), 150814-ba-777-jpo-547\_524ab530eec65d475b13f71fb8193cf0.nbcnews-ux-2880-1000 [ONLINE]. Available at: <http://www.nbcnews.com/news/world/british-airways-boeing-777-emergency-landing-after-cockpit-starved-air-n409756>, <https://media3.s-nbcnews.com/j/newscms/2015_33/1172216/150814-ba-777-jpo-547_524ab530eec65d475b13f71fb8193cf0.nbcnews-ux-2880-1000.jpg> [Accessed 23 March 2017].

OSX windows linux image

Nam Le, (2013), windows-mac-linux\_BhWrJ\_25552 [ONLINE]. Available at: <http://lenam701.blogspot.co.uk/2013/02/linux-mac-os-x-windows-and-me.html>, <http://1.bp.blogspot.com/-PewHKvcErFA/T1Oks0PGYxI/AAAAAAAACKA/Lb5fz_wt08I/s400/windows-mac-linux_BhWrJ_25552.jpg> [Accessed 23 March 2017].

Fridge image

NA, (2011), 244634\_large [ONLINE]. Available at: <http://www.ethicalconsumer.org/buyersguides/appliances/fridgefreezers.aspx>, <http://www.ethicalconsumer.org/portals/0/images/reports/244634_large.jpg> [Accessed 23 March 2017].

Keyboard image

Jason Giddings, (2012), multi-touch-keyboard-and-mouse-by-jason-giddings2 [ONLINE]. Available at: <http://www.tuvie.com/multi-touch-keyboard-and-mouse-by-jason-giddings/>, <http://futuristicnews.com/wp-content/uploads/2012/05/futuristic-Multi-Touch-Transparent-Wireless-Keyboard-Mouse-02.jpg> [Accessed 23 March 2017].

Toaster image

DELONGHI, (2016), u\_21469765 [ONLINE]. Available at: <http://www.currys.co.uk/gbuk/household-appliances/small-kitchen-appliances/toasters/delonghi-icona-vintage-cto-v4003bg-4-slice-toaster-cream-21469765-pdt.html>, <http://brain-images.cdn.dixons.com/5/6/21469765/u_21469765.jpg> [Accessed 23 March 2017].

Spitfire text description

Wikipedia. 2006. Supermarine Spitfire. [ONLINE] Available at: <https://en.wikipedia.org/wiki/Supermarine_Spitfire> [Accessed 24 March 2017].

Spitfire image

Franck Cabrol, (2006), Ray\_Flying\_Legends\_2005-1 [ONLINE]. Available at: <https://en.wikipedia.org/wiki/Supermarine_Spitfire>, <https://en.wikipedia.org/wiki/File:Ray_Flying_Legends_2005-1.jpg> [Accessed 24 March 2017].

VMWARE image

VMware Inc, (2006), VMW\_Boxshot\_WorkstationPro12-5\_213x225 [ONLINE]. Available at: <http://store.vmware.com/store/vmwde/en_GB/DisplayProductDetailsPage/ThemeID.29219600/productID.323427300>, <http://drh2.img.digitalriver.com/DRHM/Storefront/Company/vmwareglobal/images/product/detail/VMW_Boxshot_WorkstationPro12-5_213x225.png> [Accessed 24 March 2017].

VMWARE text

VMware.com. 2006. VMware Workstation 12.5 Pro. [ONLINE] Available at: <http://store.vmware.com/store/vmwde/en_GB/DisplayProductDetailsPage/ThemeID.29219600/productID.323427300> [Accessed 24 March 2017].

GTX titan text

Geforce.co.uk. 2015. NVIDIA TITAN X. [ONLINE] Available at: http://www.geforce.co.uk/hardware/10series/titan-x/. [Accessed 24 March 2017].

GTX titan image

NVIDIA Corporation, (2015), geforce\_gtx\_titan-x\_3qtr\_front\_left [ONLINE]. Available at: <http://www.geforce.co.uk/hardware/10series/titan-x/>, <http://assets.nvidia.com/nv/pascal/images/geforce_gtx_titan-x_3qtr_front_left.png> [Accessed 24 March 2017].

Croquette text description

Wikipedia. 2004. Croquette. [ONLINE] Available at: <https://en.wikipedia.org/wiki/Croquette> [Accessed 24 March 2017].

Croquette image

deramaenrama, (2006), Croquettes with salad [ONLINE]. Available at: <https://en.wikipedia.org/wiki/Croquette>, <https://upload.wikimedia.org/wikipedia/commons/5/57/Croquettes_with_salad.jpg> [Accessed 24 March 2017].