Cardiff School of Computer Science and Informatics

Coursework Assessment Pro-forma

Module Code: CM1102

Module Title: Web Applications

Lecturers: Dr Natasha Edwards, Dr Martin Caminada

Assessment Title: Online Shop

Assessment Number: 5

Date Set: 20th February 2020

Submission Date and Time: by 9:30 am on Friday, 24th April 2020 (Week 10)

Return Date: by 26th May 2020

This assignment is worth 50% of the total marks available for this module. If coursework is submitted late (and where there are no extenuating circumstances):

1. If the assessment is submitted no later than 24 hours after the deadline, the mark for the assessment will be capped at the minimum pass mark.
2. If the assessment is submitted more than 24 hours after the deadline, a mark of 0 will be given for the assessment.

Your submission must include the official Coursework Submission Cover sheet, which can be found here:
https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf

Submission Instructions

You will need to submit your coursework to THREE locations: 1. Learning Central; 2. GitLab; 3. OpenShift. More specifically:

1. Submission to Learning Central

The following files need to be submitted via Learning Central:

No.	Description		Type	Name
1.1	Cover sheet	**Compulsory**	One PDF (.pdf) file	*group_X*.pdf (*replace* 'X' *with* **your** group number)
1.2	Website files in a single zip file	**Compulsory**	One ZIP (.zip) file	*group_X*_website.zip (*replace* 'X' *with* **your** group number)
1.3	Plain README text file	**Compulsory**	One .txt file	*group_X*_README.txt (*replace* 'X' *with* **your** group number)

- For **1.2 "Website files in a single zip file"** above:
  - This must be a **single .zip archive** that includes **the complete source code** of your website, with the same structure as needed for deployment on OpenShift server, including all required assets such as images, requirements/ dependencies, etc.

- For **1.3** above **your "Plain README text file"** must include:
  i. **URL of your website** on OpenShift server;
  ii. For **EACH** group member:
    ii.a. **Student number**;
    ii.b. **E-mail address**;
    ii.c. **Responsibility**, i.e. which part/ functionality of the website this student was responsible for developing;
    ii.d. Admin user log in details, i.e. admin username and password.
  iii. **full (40 characters) CHECKSUM of your last commit** you performed on your repository on GitLab. This will be used to check the timestamp of the latest commit/ changes in your repository. **Note: no changes are allowed after the submission deadline!** Non compliance with this requirement, i.e. working on the coursework after the deadline, may be penalised and may result in capping the mark at the pass rate (for the work submitted < 24 hrs late) or an award of zero marks (> 24 hrs late submission).
    - ***Instructions on how to access the checksum*** *of your last commit on GitLab are found in Appendix A.*

✓ A template of **`README.txt`** file is available on Learning Central.

2. Submission to GitLab (https://gitlab.cs.cf.ac.uk/)

Share your GitLab repository with Natasha Edwards as follows:

- Go to: **`Settings > Members > Select members to invite`**
- search for Natasha Edwards (user name: **`scmne`**)
- Choose a role permission: **`Maintainer > Add to project`** ...

3. Submission to OpenShift (https://openshift.cs.cf.ac.uk/)

Deploy your website on OpenShift by the deadline specified on the front page.

Rules, Restrictions and Further General Comments

- Any deviations from the submission instructions above (including the number and types of files submitted) will result in the loss of marks, as specified above and *Assessment Criteria and Marks Allocation'* section of this document.
- **Staff reserve the right to invite students to a meeting to discuss coursework submission.**
- If you have extenuating circumstances that prevent you from submitting on time, you should follow the School's Extenuating Circumstances procedure (see the Student Handbook for more detail - available on Leaning Central).
- You are reminded of the need to comply with Cardiff University's Student Guide to Academic Integrity. If you use any external resources, these need to be properly referenced (e.g. as in-line comments in your code). You should also read the School's electronic coursework submission policy, available on Leaning Central.

Assignment

INSTRUCTIONS

Working in a group, you are tasked with constructing a **web-based system for an online shop**, which includes the customer-facing website and backend admin site.

- The **website** should allow the customers to browse through the available goods (or services), view their price and all other relevant information, allow the customers to add the goods to the shopping basket, and finally view the basket (display all selected items and their total price) and allow the customer to enter payment details. It is **not required** to implement the actual payment mechanism.
- The **backend admin site** should allow the website administrators to update the product catalogue, such as insert or delete a product, change a product's price; and view a list of products on sale.

Structure and Functionality of the Website

The website should be organised into at least the following sections and pages:

1. The **product pages**:
   - The **front product gallery page** should be used to present customers with a brief **gallery** of goods and their primary characteristics. This page should:
     - Allow the customer to view brief information about an item, such as the item's title and price.
     - Allow the customer to click on an item (e.g. title or image) so that the customer can view detailed information about an item.
     - Allow the customers to sort the items according to selected criteria (*e.g.* price, mass, size: depending on what properties you choose your items to have).
     - Allow the customer to add an item or items to the shopping basket.

   - The **detailed product** page should be used to display **detailed information** about a particular item (a *'single product'* web page). Just like on the front page, the user should be able to add the currently selected item to the shopping basket.

2. Yet another page should display the contents of the **shopping basket**. The shopping basket page should:
   - Be accessible only to the customer who has placed the goods there.
   - Show all the chosen items and their total price.
   - Allow the customer to delete an item (or all items) from the shopping basket.
   - Be persistent within the session.

3. Your website should provide the user with a **user account** functionality, which would allow the user to register, log in and log out, delete their account and update their information (such as email, address, password). The system should display helpful messages when the user is not able to register, log in or log out, change their details or delete the account.

4. Another page (**checkout**) should allow the user to enter their credit card payment details, and confirm the final price to be paid. As the user steps through the fields of the form, help messages should be displayed explaining what information needs to be entered in each field. There is no need to implement an actual checkout or secure payment mechanism. It is sufficient to implement only a basic validation for the payment form: when the user fills in the form incorrectly and attempts to submit it, a helpful error message should be displayed. When the user's input

is valid and the user submits the form, an order should be created and a confirmation page should be displayed. It is not necessary to process this form in any other way, e.g. process payment.

5. **Admin page**(s) should enable an admin user to update the product catalogue by adding or deleting items. The admin should be able to change an individual item information, such as an item's title, description and price. The admin user should also be able to view a list of all products sold on your website.

For groups of 6 and 7, EACH team member must choose one functionality to implement from the following list:

6. **Wishlist**, which enables a customer to save or delete an item, choose to view an item's details (on the item's individual page), and move a particular item to the shopping basket.
7. A facility to prepare **shipping labels** and **electronic invoices** in PDF format.

Website Quality

Overall, the website should be aesthetically pleasing, *e.g.* display an attractive banner with the shop title, contact phone number, lay out the items in a nice tabular form, offer intuitive navigation, have consistent 'look and feel'.

Individual Team Member's Responsibilities

This coursework consists of groupwork and individual elements - see *"Assessment Criteria and Marks Allocation"* section below. As well as contributing to the group-based activities, each team member must choose or be allocated one of the functionalities specified in the *"Structure and Functionality of the Website"* section above. For a team of 5 people, implementation of functionality No. 1, 2, 3, 4, and 5 is compulsory [1]. For those teams with 6 or 7 members, you need to add the No. 6 *'Wishlist'* and/or No. 7 *"Shipping labels and electronic invoices"*. It is up to the team to decide how to allocate the responsibilities, but the final allocation of these must be clearly listed in README.txt file. You are allowed to help each other, but each team member will receive individual mark for the functionality s/he was responsible for.

Goods for Sale

The nature of goods to be "sold" by your online shop is your choice. The site, however, must not include any content that is likely to be offensive to others. As the minimum, the items in your shop should have the following properties:

- Name
- Textual description
- Picture
- Price

and any other relevant characteristics.

The items in your shop may be imaginary, but should be plausible in order to illustrate the functionality of your website. Please populate your databases with at least five nice examples (items) prior to submission so that the examiners can adequately test the functionality of your shop.

[1] In the event, the number of team members drops down to a lower number, e.g. 4, the team should implement functionality No. 1 - 4.

Technologies to be Used

In order to demonstrate your ability to use Flask micro web framework and other various web technologies introduced in the module, your online shop should use the following:

- Navigational menus with links to the different parts of the website (*e.g.* "view basket", "go to front page" etc.)
- Lists, which could be used, for example, for menus or to itemise properties of the shop items.
- Tables, which could for instance be used to format the list of available items.
- CSS style sheets, including an external CSS file to define the appearance of your online shop.
- Links to external web resources (*i.e.* via the anchor element), for example to a manufacturer's website.
- A MySQL database that maintains the content of the shop.
- You may also use JavaScript if/ when/ where appropriate, for example to provide feedback to the customers when they update their shopping basket. (Note: you do not have to use JavaScript, only if you want to. The same functionality can be implemented in Python/ Flask.)

Important Conditions

- The software tools and technologies that are to be employed on the website are HTML (HTML5 and XHTML are allowed), CSS, MySQL, JavaScript and Python/ Flask. The use of XML is also allowed.
- Use of libraries, APIs, etc. is allowed, however, the final code must be authored by you. If you use external libraries, APIs, etc. you must provide an explanation of how and where these were used AND complete references in your code and in a separate file (e.g. README.md).

Note that there should be no need to produce a user guide as it is assumed that the website interface will be self-explanatory.

Good luck!

Appendix A: Checksum of your last commit (on GitLab)

To access the checksum of your last commit, go to the Project's main page, and click on the button next to the 8-character text - see Fig. 1. This will copy the full checksum (i.e. 40 characters). Paste the copied text into your README file.

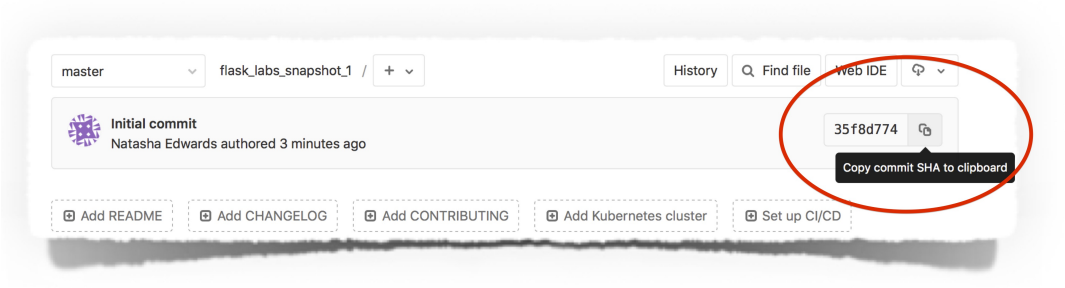

Figure 1: Accessing checksum of your last commit on GitLab

ONLINE SHOP COURSEWORK
ASSESSMENT CRITERIA AND MARKS ALLOCATION

	Max Marks
Correct Submission (group mark)	
Coursework Coversheet binary mark: *no* = 0, *yes* = 1	1 (binary)
README.txt file *not submitted* = 0, *minor issues* (e.g. some information is missing)=1	2
Website files in a single zip file binary mark: *no*=0, *yes*=1	1 (binary)
TOTAL FOR CORRECT SUBMISSION	**4**
Website Build and Deployment (group mark)	
The website has been successfully deployed on OpenShift binary mark: *no*=0, *yes*=10	10 (binary)
TOTAL FOR WEBSITE DEPLOYMENT	**10**
Database implementation (group)	
All content is generated programatically, i.e. pulled from the database, not 'hardcoded' manually Mark is binary here: *no*=0, *yes*=10	10
TOTAL FOR DATABASE IMPLEMENTATION	**10**
Functionality (individual [2])	
1. Product gallery and detailed product pages:	
The list of items is displayed to customers	
- with each item's name, price, description, and image(s) are shown	3
Customers can sort items according to selected criteria	3
An individual product page is shown when Customers click on that item	3
Customers can add selected items to the basket (from the gallery or individual product page)	3
2. Shopping basket:	
Shopping basket correctly displays the chosen items	3
The total price is displayed correctly	3
Customers can delete items from the basket	3
. . . and the shopping basket view is then updated to reflect this	3
3. User Accounts:	

[2] one per student; based on the responsibility specified in the README.txt

Customers can register to create an account	3
Customers can log in into and log out from their account	3
Customers can delete their account	3
Error messages are shown when the user in not able to register, log in or log out	3

4. Checkout (payment form):

Payment form is shown when the user proceeds to checkout	3
Help messages appear as the user steps through the input fields	3
On valid user input, an order is created and a confirmation page is displayed	3
Error messages are shown when the user incorrectly fills in the form	3

5. Admin page(s):

The admin user is able to add a new product to the product catalogue	3
The admin user is able to delete a product from the product catalogue	3
The admin user is able to change details of a product (e.g. title, description, price)	3
The admin user is able to view a list of all products	3

Functionality for groups of 6 and 7

6. Wishlist:

The customer is able to add an item (product) to their wishlist	3
The customer is able to delete an item from their wishlist	3
The customer is able to view an item located in their wishlist	3
The customer is able to move an item from their wishlist to the shopping basket	3

7. Shipping labels and electronic invoices:

Once an order is confirmed, the system generates a shipping label for that order	3
The admin user can access shipping label for each sold item on the admin page	3
Once an order is confirmed, the system generates an invoice for that order	3
The order invoice is emailed to the customer	3

TOTAL FOR EACH FUNCTIONALITY	**12**

Usability and presentation (group)

Ease of navigation (how easy it is to access every page from every other page)	5
Consistency of the design and professionalism in visual presentation	5
TOTAL FOR USABILITY AND PRESENTATION	**10**

TOTAL MARKS for the COURSWORK (max)	**46**

MARKING GUIDE

Functionality (individual mark)	
Excellent (3 marks)	Functionality described in the instructions fully works. There are no detectable bugs. The system is robust under all circumstances.
Good (2 marks)	Functionality described in the instructions mostly works. The examiner spots some inconsequential bugs that do not substantially affect the operation of the website. The system remains stable during normal usage, but fails when the examiner deliberately tries to break it.
Poor (1 mark)	Only partially functioning features. There are obvious bugs that substantially affect the operation of the website. The system behaves unpredictably during normal usage. Serious flaws in several functions. Persistent bugs affecting the functionality of the website.
None (0 marks)	Not implemented or does not work at all.

Usability and presentation	
Excellent (4-5 marks)	• Navigation is intuitive, consistent and self-explanatory, with clear navigational menus on all pages providing access to all other parts of the site without breaking the logic of operation. Navigation is personalised to a particular user (*e.g.* the website greats the user by her/ his name). • All shopping and basket pages are informative, with professionally presented, logically and consistently laid out content; effective use of HTML and other 'display' elements, with CSS control of styling.
Good (2-3 marks)	• Navigation is mostly intuitive, with a few limitations in ease of navigation or with use of menus and links, *e.g.* the user may need to re-enter some information once they left a page and came back. • Reasonable presentation of content, although not yet of professional quality. Generally well laid out using CSS where appropriate but some limitations in overall coherence of design and use of space on page.
Poor (1 mark)	• Means of navigation are confusing. Using the shop is a chore. • Inconsistent design across website with poor presentation: failure to employ CSS appropriately to control presentation. Text is often difficult to read (*e.g.* poor colour contrast, obtrusive background patterns, text too small or unnecessarily too big).
None (0 marks)	• None, absent, not implemented, not applicable.