**Cardiff School of Computer Science and Informatics**

# Coursework Assessment Pro-forma

**Module Code**: CM1102

**Module Title**: Web Applications

**Lecturer**: Dr Crispin Cooper

**Assessment Title**: Online Shop

**Assessment Number**: 3

**Hand out date**: 8th March 2023

**Submission Date and Time**: by 9:30 am on Friday, 12th May 2023

**Feedback return date:** Youwill be assigned a live marking session to attend during exam period. Feedback and provisional marks will be returned to you live during this session.

**If you have been granted an extension for Extenuating Circumstances, then the submission deadline and return date will be 1 week later than that stated above.**

**If you have been granted a deferral for Extenuating Circumstances, then you will be assessed in the summer resit period (assuming all other constraints are met).**

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.

Extensions to the coursework submission date can only be requested using the [Extenuating Circumstances procedure](https://intranet.cardiff.ac.uk/students/study/exams-and-assessment/extenuating-circumstances). Only students with approved extenuating circumstances may use the extenuating circumstances submission deadline. Any coursework submitted after the initial submission deadline without \*approved\* extenuating circumstances will be treated as late.

More information on the extenuating circumstances procedure can be found on the Intranet: <https://intranet.cardiff.ac.uk/students/study/exams-and-assessment/extenuating-circumstances>

By submitting this assignment you are accepting the terms of the following declaration:

**I hereby declare that my submission (or my contribution to it in the case of group submissions) is all my own work, that it has not previously been submitted for assessment and that I have not knowingly allowed it to be copied by another student. I understand that deceiving or attempting to deceive examiners by passing off the work of another writer, as one’s own is plagiarism. I also understand that plagiarising another’s work or knowingly allowing another student to plagiarise from my work is against the University regulations and that doing so will result in loss of marks and possible disciplinary proceedings*[[1]](#footnote-1).***

# Assignment

The aim of this assignment is to construct a **website for an online shop**. 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.

To avoid financial liability, however, you must not implement any actual payment mechanism!

## 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
* Measure of environmental impact e.g. carbon or ecological footprint

The items in your shop may be imaginary, but should be plausible in order to illustrate the functionality of your website. Environmental impact may be quantified any way you like, e.g. carbon footprint, ecological footprint, or otherwise (we will not assess whether the numbers you use are accurate). Please populate your databases with **at least four** nice examples (items) prior to submission so that the examiners can adequately test the functionality of your shop.

## Essential Features / Structure of the Site [worth 90% of the available marks]

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

1. The front page should be used to present customers with a brief image gallery of goods with their names, prices and environmental impacts. The front page should:
   1. Allow the customer to add an item or items to the shopping basket
   2. Allow the customer to view a more detailed description of an item when they choose to do so (e.g. by clicking on an item image or title).
   3. Allow the customers to sort the items according to name, price or environmental impact
   4. Include advertisements for each of your partner stores (see below)
   5. Be responsive, e.g. the website looks and works equally well on mobile devices, tablets and PCs. There is no need to test on other devices, but you should use a suitable browser plugin to ensure that content is rearranged appropriately for a mobile view.
2. Another 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.
3. Yet another page should display the contents of the shopping basket. The shopping basket page should:
   1. Show all the chosen items and their total price
   2. Allow the customer to delete an item (or all items) from the shopping basket
   3. Be persistent within the session, i.e. while the user is navigating the site, items they place in the basket should remain there
   4. Be accessible only to the customer who has placed the goods there, i.e., if while browsing the site you open a new anonymous browser window (to simulate a different customer) and visit the site, the new customer should not see the contents of the first customer’s basket.
4. Another page (checkout) should allow the user to enter their credit card payment details, and confirm the final price to be paid.
   1. As the user steps through the fields of the form, help messages must appear explaining what information needs to be entered in each field.
5. **Do not implement an actual checkout or secure payment mechanism.** Instead, implement a basic validation for the payment form to check that, when the form is submitted,
   1. the entered credit card number is a 16 digit number (you may permit dashes and spaces),
   2. that none of the other required fields are empty,
   3. plus any other validation checks you think necessary.

If the submitted form validates correctly, present a page saying that checkout was successful. If there are errors in the form (e.g. the credit card number field contains other text) you may either prevent the form from submitting, or report an error after it submits.

## Partner stores

You will be assigned to a small group with 1-2 of your peers. For the purpose of this exercise, you are to imagine you have entered a business agreement to promote one another’s stores. You must provide the other members of your group with artwork (e.g. a small banner, text, etc) which you wish them to use to promote your own store, and help them ensure this is integrated into the front page of their site. Likewise, you must integrate their advertisements into your own site. As you will be deploying your websites only locally (on your own or the lab machines) you are not expected to make hyperlinks to your partner stores, only to display the artwork.

In the event that a group member does not engage in providing artwork in reasonable time, please get in touch with your module leader to resolve this issue.

## Overall considerations

The website should be aesthetically pleasing, e.g. display an attractive banner with the shop title, lay out the items in a nice tabular form, be clearly legible and offer intuitive navigation.

Your code should be written according to principles taught on this course, e.g. Don’t Repeat Yourself (at least not unnecessarily), do use meaningful names for methods, variables, database entities etc. We will not read your code exhaustively to check this, but will decide in advance some set locations in your code we will check for compliance, and award marks for each that passes.

Your code **must** be tracked in git version control, we will use this during assessment to check you are demonstrating the same version of the code as you submitted to learning central.

## Advanced Features (worth 20% of the available marks)

The above features allow for a mark of up to 80%. A further 20% is obtainable by providing additional functionality or qualities for the website, examples of which are (but are not restricted to):

* Functionality to allow customers to leave reviews relating to particular items
* A search facility allowing the customers to find items according to various search criteria
* A facility to prepare shipping labels or electronic invoices (e.g. as a .pdf file)
* Extra features to enhance security
* Ability to create a user account, log in and log out

There are endless possibilities here and you are encouraged to think of your own extensions. **Note:** 20% is the **maximum** mark obtainable for any combination of the features in this category.

## 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:

* 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 SQLite or SQL database that maintains the content of the shop.
* 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. Python/Flask MUST be used for the server/back-end code; JavaScript may be used client-side (running in the browser) only.
* Use of libraries, APIs, etc. for a limited range of components is allowed, however, the final code must be authored by you – i.e. you cannot just use an existing “online shop” module! If you use external libraries, APIs, etc. you must add a footer to your main page with links to e.g. their homepages, github repos or documentation.

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

# Learning Outcomes Assessed

The following Learning Outcomes are being assessed through this coursework:

* Front-end website construction using HTML, CSS and JavaScript
* Server side development (e.g. Linux or Windows) using a scripting language (Python) and framework (Flask)
* Awareness of the main network protocols and Internet standards.
* Being able to debug web applications through server logs and browser analytical tools
* Making use of online resources and/or documentation
* Understanding modern web development, including appreciation of usability aspects
* Appreciation of legal and ethical issues, e.g. security, data protection
* Appreciation of commercial and economic context of online business

# Guidance on working together and plagiarism in modern web applications

***On this module we encourage you to help one another and share knowledge with peers.*** Copying large parts of code from others will be viewed as plagiarism and we reserve the right to use automated tools to check for this, however, we recognise that due to the nature of programming within a web framework, some aspects of assignments are likely to be completed by many students using similar code. For example, in places we would expect to see snippets of code similar to those we previously gave to you during lab exercises. Even outside of these, it is likely that by chance alone, a few students will write identical fragments of code 1-2 lines long. Only if automated tools indicate a much greater quantity of similar code than is normal, or if these shared fragments are longer blocks of code rather than isolated occurrences, will this be referred to a human for academic judgement on whether plagiarism is likely to have taken place.

# Criteria for assessment

Credit will be awarded against the criteria listed in the tables below, with weights attached to each area of assessment as indicated.

Linesmarked \* are assessed in binary fashion (you either get full marks for that item, or 0)

|  |  |  |
| --- | --- | --- |
| Item | Marks |  |
| at least 4 items with names, prices, images, generated from the database | 15 | \* |
| advertisement(s) appear for partner store(s) | 3 | \* |
| detailed descriptions load when items clicked | 2 | \* |
| sorting works | 3 | \* |
| can add items to basket | 3 | \* |
| another user gets a different basket | 3 | \* |
| basket shows total price | 3 | \* |
| can remove item from basket and total price is updated | 3 | \* |
| checkout shows hints as you step through form fields | 3 | \* |
| error message is shown if the user incorrectly fills out the form | 1 | \* |
| success message if they correctly fill the form | 1 | \* |
| mark for aesthetics | 20 |  |
| source code spot checks for quality (see above under ‘overall considerations’) | 20 |  |
| extra credit | Max. 20 |  |

## Criteria For Design/Visual Presentation Mark

|  |  |
| --- | --- |
| Outstanding (20 marks) | Design is above and beyond quality of the average professional shopping site. |
| Professional (15 marks) | Design as would be expected of a professional site. Navigation is intuitive, consistent and self-explanatory, with clear navigational bar providing access to other parts of the site. 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  (12 marks) | 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  (5 marks) | Means of navigation are confusing; using the site is a chore. Inconsistent design, poor presentation, failure to employ CSS appropriately to control presentation. Text difficult to read e.g. poor contrast, obtrusive background patterns, text too small/large. |

## Examples of Extra Credit Marks (max 10 in total)

|  |  |
| --- | --- |
| Functionality to allow visitors to leave reviews relating to particular items | 5 |
| A search facility allowing to find items according to specified search criteria | 5 |
| A facility to prepare shipping labels | 5 |
| A facility to prepare electronic invoices | 5 |
| Extra features for security | 5 |

# Feedback and suggestion for future learning

Feedback on your coursework will be delivered during your assigned live marking session scheduled during exam period. Feedback will address the assessment criteria. Feedback for this assessment can be used to improve your skills in professional web development, and might be useful for other modules, e.g. your Second Year Group Project and Final Year Project - if you decide to do a web-based project.

# Submission Instructions

The following files must be submitted via Learning Central:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Description | | Type | Name |
| 1 | Website source code | Compulsory | One ZIP (.zip) file | <student number>\*\_website.zip |

\*replace <student number>with **your** student number

Any deviation from the submission instructions above (including the number and types of files submitted) may result in a mark of zero for the assessment.

"Website source code" must be a **single .zip archive** that includes **the complete source code** of your website, including all required assets such as images, with the same structure as needed for local deployment.

The website **must** be tracked in git version control. **Prior to submission, you must COMMIT ALL CHANGES TO YOUR GIT REPO AND CEASE FURTHER WORK ON YOUR SITE.** During assessment we may ask you to run **git status** to demonstrate that your working directory does not differ from your repo, and **git log** to show the time of the last commit is not later than the assessment deadline. We may check that the code you demonstrate to us is the same version as the code in your submitted zip file. **If the code is found to differ, zero marks will be awarded for the assessment.**

Please note:

1. If you have used SQLite, this zip file should include your SQLite database file
2. If you have used the school SQL server, it is not necessary to include the database, however you should keep it on the school server until after the coursework has been assessed
3. The zip file should NOT include your virtual environment (venv) or git (.git) directories

Non-compliance with the submission instructions can lead to all or part of the marks being subtracted.

For assessment, we will ask you to demonstrate your website, over a Teams call, from whichever machine you have used to develop it (i.e. your own, or ours). We will also ask you to explain parts of your source code to us.

Staff reserve the right to invite students to a meeting to discuss coursework submissions

# Support for assessment

Questions about the assessment can be asked on https://stackoverflow.com/c/comsc/ and tagged with ‘cm1102’. Support for the programming elements of the assessment will be available in daily drop-in lab sessions.

Further details on support are available on the module page on Learning Central, under ‘Getting help and support’.

1. https://intranet.cardiff.ac.uk/students/study/exams-and-assessment/academic-integrity/cheating-and-academic-misconduct [↑](#footnote-ref-1)