
PART 1 (10%)

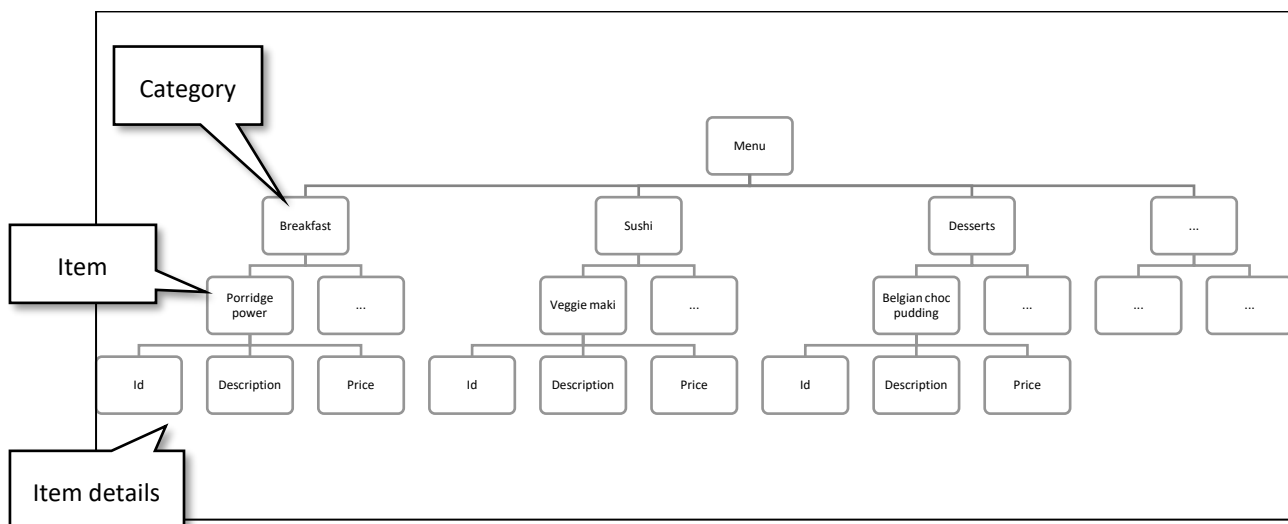
- 1) Set up an HTTP server to support PHP scripts (outline the entire process and technologies used).
- 2) Write a simple script which returns the current date and time on the server back to the client
- 3) Write a simple script which stores, for each user, the date and time of the first time the page is loaded. On subsequent interactions, the script should return something on the lines of: "You first used this page x seconds ago". Outline the techniques used to achieve this.
- 4) Write a script that (a) reads a set of parameters sent by the client and (b) stores these request parameters in session variables. Write two versions, one for GET and the other for POST requests.
- 5) Write a simple script that reads session variables stored by the scripts above and echoes them neatly back to the client. Explain how sessions work, and how PHP scripts can differentiate between different user sessions.

PART 2 (90%)

Based on the principles and best practices discussed throughout this course, you are asked to build a dynamic website for a restaurant of choice. This website should provide:

- a) generic information about the restaurant (including address and opening hours),
- b) information about the people running it,
- c) a way to allow users to contact the restaurant (to book a table, send a query or file a complaint),
- c) information about the current menu being served at the restaurant (i.e. list of dishes) and
- d) a detailed description of each item on the menu (on a separate page from the list provided in c).
- e) users should be able to add individual dishes to a “favourites” list (from within the dish details page),
- f) which list can then be accessed from a dedicated page, listing all the marked dishes (and their details), while allowing users to remove dishes from the list or send the entire list to an email address of their choice.

The menu page as well as menu-item detail pages should be dynamically generated based on a separate structured data file or a simple database containing a hierarchal representation of dishes served by the restaurant. XML, CSV as well as relational databases are all acceptable formats to represent the necessary data structure (as shown below). This strategy would allow non-technical users to modify their site content independently.



GROUP RESPONSIBILITIES FOR PART 1 AND PART 2

Each team is to submit a signed teamwork declaration, including:

- 1) An attendance log for each team member (i.e. team meetings)
- 2) Responsibilities taken up by each team member at any point in time
- 3) Detailed description of contributions made by each team member towards the project’s deliverables
- 4) An overview of group dynamics throughout the project (i.e. overall experience, issues encountered, how these were resolved etc...).

Each team member may also be asked to explain any aspect of the project during the final interview. This means that team members must be closely involved throughout the project’s various activities, from design to implementation.

OTHER DETAILS FOR PART 1 AND PART 2

- Work is to be carried out in group (2 to 3 students per group).
- You are to build **everything from the ground up** (*no boilerplate or copied PHP code, CSS themes or front-end frameworks*). Patterns, best-practices and technologies necessary to complete this assignment will be discussed in depth in-class.
- Don't worry if the result is not polished. This is a learning exercise and as such plagiarism will not be tolerated.
- Client-side code must be tested on at least two major browsers for appearance and behaviour.
- You may be called in for a 15-minute clarification meeting (in group) at the end of the course.
- The write up should include the teamwork and plagiarism declaration form. This document should be split in two parts according (Part 1 and Part 2). For Part 2 you are expected to provide a technical overview of the project including the site's architecture and how it works (files/scripts used, data structure, site behaviour, validation and so forth). This document (accompanied with the code-base) should be submitted in the respective VLE area (which will be available closer to submission date). A Word template for the document is available from the Departmental website.
- Free PHP reference manual: <http://php.net/manual/en/index.php>

MARKING FOR PART 1 AND PART 2

- 1) Style – readability of the code, including comments, indentation and proper application of naming conventions.
- 2) Requirements coverage – the extent of your solution with regards to the given requirements (does your solution cover all the requirements)
- 3) Architectural elegance – what design decisions were taken: Is the solution maintainable? Is rigorous validation implemented? Has the group adopted DRY principles? Can new functionality be added effortlessly? Is the solution efficient in terms of bandwidth consumption, client and server-side resource consumption?
- 4) Theoretical understanding – questions will be asked about your understanding of protocols used, performance considerations, efficiency, standards as well as best practices – as discussed in class.