CE01084-4 Introduction to Web Programming

# Introduction to Web Programming Assessment Part 4

## Preliminary

Part 4 of the assessment is worth 60% of the overall marks for the coursework. The coursework comprises 100% of the marks available in the module.

All web pages should use, and be compliant with, the following doctype:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

All functionality should take place on the server and be written in PHP.

## Task – Car Sales

### Scenario

You are to create a small web site which allows a user to register as a member and, once a member (and logged in), to conduct searches against a database of cars for sale. The user should also be able to edit their registration details.

### Required web site pages

Your site should include at least the following pages:

* Welcome page (open to anyone)
* Registration page (open to anyone)
* Car search page (requires user to be logged in)
* Change registration details page (requires user to be logged in)

### Minimum Database Requirements – user

Surname, forenames, date of birth, address, email and telephone number

### Minimum Database Requirements – vehicle

Make, model, year, CC, colour, picture (a picture can be entered in the database as a URL to an image).

### User Functionality – minimum

* Register
* Edit registration details
* Log in
* Log out
* Search for a car against a single criteria (for example search against model or search against year)
* The ability to do a partial string search.
* Validation of registration details to check a value has been entered

### User Functionality – additional

* Personalised message on log in
* Search on multiple fields
* Save completed searches for future use
* Use a saved search
* Be presented with their favourite search when they log in.
* Validation of registration details to check using a pattern match and data type.

### You are Required To

* Build a database to store all the data necessary for the task.
* Create a web application using HTML, CSS and PHP to carry out the task.
* Produce documentation for both the design (pseudo code or Hierarchical Task Analysis tree’s) and testing of your web application (test cases and results of testing).

# Marking Criteria Assessment Part 4

## Minimal Requirement (guaranteed 40% of the marks available for Part 4)

|  |  |
| --- | --- |
| **Function/Requirement** | **Marks (out of)** |
| Database and table(s) (at least 10 car records) | 1 |
| Welcome page | 1 |
| Working Log in | 1 |
| Working Log Out | 1 |
| Working registration page | 1 |
| Validation of registration details (whether value is entered) | 1 |
| Working change registration details page (logged in only) | 1 |
| Working search page (search on any single field – logged in only) | 1 |
| Working partial string search | 1 |
| Standards compliant XHTML (validated) | 1 |
| Semantically structured XHTML | 1 |
| Standards compliant CSS (validated) | 1 |
| Design documentation | 1 |
| Testing documentation | 1 |

## Additional Requirements (worth an additional 40% of the marks available for Part 4)

|  |  |
| --- | --- |
| **Function/Requirement** | **Marks (out of)** |
| Appropriate database tables | 1 |
| Appropriate database field types | 1 |
| Personalised log in (not favourite search) | 1 |
| Authorised access to secure pages (requires log in) | 1 |
| Search on multiple fields at the same time | 1 |
| Two or more search fields dependant on each other (eg. manufacturer and type of car) | 1 |
| Save multiple searches | 1 |
| Use saved search | 1 |
| Create favourite search | 1 |
| Present results of favourite search on log in | 1 |
| Registration details validation (pattern match and type verification) | 1 |
| Appropriate and detailed design documentation | 1 |
| Appropriate and exhaustive testing | 1 |

## Discretionary Mark (worth an additional 20% of the marks available for Part 4)

There will be a further mark available as the discretion of the assessor to be awarded for the adoption of best practices as evidenced in the PHP code.

|  |  |
| --- | --- |
| **Function/Requirement** | **Marks (out of)** |
| Adoption of ‘best practice’ PHP development | 100 |

70+

Exceptional code of a professional standard that is clearly written (indented, variables and functions well named, follows a naming convention) using a consistent style with little or no duplication and is well documented (code commenting).

60-69

Very good code that is clearly written (indented, variables and functions well named, follows a naming convention) using a consistent style with little or no duplication and has some documentation (code commenting).

50-59

Very good code that is clearly written (indented with good naming convention) some documentation (code commenting).

40-49

Good code that is well written (indented, descriptive names used) using a consisted style and or some documentation (code commenting).