# Solution

The Occupational Specialism is marked in a variety of Categories:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Functionality** | **Code organisation** | **User experience** | **Legal and regulatory guidelines and standards** | **Testing - suitability of testing** | **Testing - Use of testing to inform the iterative development process** | **Documentation - Quality of iterative development process** |
| Marks | 8 | 8 | 8 | 6 | 6 | 6 | 6 |

## Functionality

* The Solution must include two different languages, in my case I will be using Python and SQL.
* Implement a complex database model which interlinks multiple tables, Primary and Foreign Key
* The use of request and response from objects within the server and client
* Login and Signup Form
* Structure the Solution Logically and implement the use of Functions and procedures
* Use of SQL
* The Techniques used within the solution must demonstrate understanding of technical Skill.
* The Features and Assets within my Solution must work together and complement each other
* Make sure I make use and display my understanding of features I’ve Implemented
* Show Why and how I’ve used these features

The Solution must take these features into account and demonstrate how I’ve Implemented them and how can I prove I’ve implemented them.

This Section of the OS has 8 Marks, breaking it down into these parts and Implementing it well Will Make it easier.

You can Plan some of this in your design document to make it easier to develop the solution.

## Code Organisation

* Avoid Unorganised and overuse of nested if clauses
* Clear, Meaningful and readable Indentation
* The Code should Implement Functions, Classes, Objects with Proper Structure
* Use Local Variables whenever possible to minimize the use of Global Variables
* Clear and Structured use of comments to explain the logic and function of sections
* Consistent Style Throughout, Pep 8 naming Conventions
* Exception Handling to Prevent ANY crashes, if an error does occur have a system to call the error without a crash.

This Section Is also 8 Marks.

## User Experience Considerations

* Does the functionality of the solution meet the needs of the user?
* Are the Outputs correct
* Is the Solution usable and easy to use in the perspective of the user
* Is the Solution Aesthetically Pleasing?
* Consistent throughout the solution (Task 1)
* Colour Theme’s Simplistic yet good looking
* Has an Accessibility Feature Been included?

This Section is also 8 Marks, you can plan this section out in the Task 1 to make it easier to implement features to benefit the solution in the perspective of the user

## Legal and Regulatory

* Accessibility
* Accessibility is a big feature; it’s a Legal and Regulatory requirement to implement it into a website.
* Does the Accessibility provide enough features to aid most people who need it.
* Compatibility
* Does the website function on other devices. (Provide Consideration, Phone Wireframe of phone)
* Does the website function on lower end devices, provide some consideration and acknowledgement for those whos devices aren’t high end.

This section is smaller than the others but its important to go into detail and cover all these things within the development and design of the solution.

Implementing these ideas in the Design will make it easier to check it off in development.

## Suitability Of Testing Data

* The Testing should implement a variety of different types of testing to make sure its tested effectively:
* Invalid Data
* Valid Data
* Boundary Testing
* Presence Testing (No Data Implemented)
* Functionality Testing (Does the System Tested work?)
* Performance Testing (How well does the website work under load), At least Cover it don’t have to implement.
* Black Box Testing: The Tester has no knowledge or concern of the internal functionality of the solution. (Task 3)

The Testing is quite a big section, the database should be tested (Every Table) to check each of these methods and prove the database is working as it should.

## Iterative Development

|  |
| --- |
| **First. Preamble**  As above state what you are planning on doing, who's it for etc. Your assets, etc. Most will always be the same except a few images (ensure they are free, open for use) |
| **Second: Title it iteration 1.**  Show the plan for the database (from your design doc).  Show your actual database.  Show your code to connect to the database  Test your database using some basic coding, and test data.  **Record these tests with reference numbers on your testing table**  Has all the coding you've done got comments? |
| **Third: Title iteration 2.**  The design of the front end (from your design docs).  Show screenshot of your actual screens.  Show and comment on your actual HTML/CSS.  Accessible features here, alt text etc.  **Record how you tested each page goes to where it should do.**Screenshots on your test plan are good, and reference these.  (There my be no actual text on your pages, just lipsum at this point) |
| **Forth: Title iteration 3.**  The log in system. This is the whole point of the system.  Show your page for log in/ and lots of tests of how you tested this (**record these again).**  Give really good explanation of what your code is doing and how it's performing security checks etc. If a library is doing this. What is the library doing.  Show that you cannot access certain things if you cannot log in. **This should all be recorded on your testing document. Remember the range of tests needed.** |
| **Fifth: Title iteration 4:**  Anything else you have on the pages. If you are using an API, this is a good opportunity to demonstrate it. |

## 

## Documentation

The Documentation will include details on what you implemented and how to prove you met all the criteria of each section.

The Documentation You can implement how you used the design and features from the Design Documentation.