**Acceptance testing**

(basically, testing all the features)

|  |  |  |  |
| --- | --- | --- | --- |
| Input sequence | Expected result | Current output | Comment |
| Requirement 1: The user should be able to log in (normal user and admin) | | | |
| Correct login for normal user:  Staff no. = 1234  Password = 1234 | User logs in successfully | A screenshot of a computer  Description automatically generated | As the user has logged in successfully, they have been directed to the dashboard |
| Incorrect login  Staff no. = 5643  Password = wrong | User does not log in successfully |  | As the details are incorrect the page does not re-direct anywhere and the search bar shows “invalid\_credentials”. |

**Code inspection**

|  |  |  |  |
| --- | --- | --- | --- |
| Inspecting | Expected result | Current output | Comment |
| Correctly naming elements (Example 1)  Task complete button | taskCompleted, completeTask, uponCompletion |  | Input fields and buttons have been correctly named so they can be identified easily. |
| Correctly naming elements  (Example 2)  Navigation bar | menuItems, navigationBar, navBar | A computer screen shot of text  Description automatically generated | Classes for the navigation bar have been correctly named. |
| Using comments | Systematic use of comments throughout code | A screen shot of a computer code  Description automatically generated  A computer screen with text and images  Description automatically generated | Screenshots from three different files, all showing systematic and consistent use of comments throughout. |
| Using correct coding conventions such as camel casing | Correct use of camel casing when naming elements | A computer code with text  Description automatically generated with medium confidence | Systematic and correct use of camel casing throughout code. |
| No harsh use of colours. | Colours are expected to flow well and are easy on the eyes. |  | Appropriate use of colours throughout e.g., green for submit/success and red for failure/delete. Other colours flow well and do not clash with each other or other elements such as font. |
| Appropriate use of font styles. | No elaborate use of styles such as handwriting. |  | Clear and easy to read font styles have been used. |
| Appropriate use of font sizes. | The font sizes should not be too large or too small. |  | Font sizes have been used appropriately, headings large, sub-headings slightly smaller and so on. |
| User experience and useability. | The site should be easy to navigate and self-explanatory. |  | Menu items in the navigation bar are self-explanatory. All elements/buttons have been labelled clearly. |
| Are login details being sent securely to the server. | Login details are not visible in the search bar and are not showing in inspector tools. |  | As we can see only the invalid credential shows in the navbar. When the user enters correct details, they are automatically redirected. |
| Error handling mechanism (Example 1) | Trying to sign up a user which has a staff number that already exists in the database. The user should not be created. |  | Appropriate error message appears when trying to sign up a user which already exists in the database. |
| Error handling mechanism (Example 2) | Trying to add a task which already exists in the database. The task should not be added. |  | Appropriate error message appears when trying to create a task which already exists. |
| Missing input handling | Leaving signup/login fields blank. |  | The user is prompted to enter the missing details. |
| Clearing cache after the user signs out so content is not accessible | Content should not visible after the user signs out as this would violate security. |  | Screenshots show cache is being cleared and session variables being reset and destroyed. This provides security to ensure logged in user cannot see previous users data. |

Unit testing