|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Student Name |  | | Student Number | |  |
| Unit Code/s & Name/s | ICTPRG535 Build advanced user interfaces  ICTPRG556 Implement and use a model view controller framework | | | | |
| Cluster Name  *If applicable* | Web Interface Cluster | | | | |
| Assessment Type | Assignment  Project  Case Study  Portfolio  Third Party Report (Workplace)  Third Party Report (Peer)  Other | | | | |
| Assessment Name | UI Design and Develop Portfolio | | Assessment Task No. | | 2 of 2 |
| Assessment Due Date |  | | Date Submitted | | / / |
| **Assessor Feedback:** | | | | | |
| **Attempt 1** | Satisfactory | Unsatisfactory | | Date | / / |
| Assessor Name |  | | Assessor Signature | |  |
| **Student provided with feedback and reassessment arrangements**  *(check box when completed)* | | | Date scheduled for reassessment | | / / |
| **Attempt 2** | Satisfactory | Unsatisfactory | | Date | / / |
| Assessor Name |  | | Assessor Signature | |  |
| Note to Assessor: Please record below any reasonable adjustment that has occurred during this assessment e.g. written assessment given orally. | | | | | |
|  | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assessment Criteria / Benchmarks  ***The evidence submitted demonstrates that the student has satisfactorily:*** | Attempt 1 | | Attempt 2 | |
| **Date**  \_\_/\_\_/\_\_ | | **Date**  \_\_/\_\_/\_\_ | |
| Y | N | Y | N |
| Part 1 - Program and Design Specification | | | | |
| 1. Developed a program specifications report that outlines each of the following points and provides justification for each choice |  |  |  |  |
| 1. The programming language/s chosen for the project |  |  |  |  |
| 1. The chosen Integrated Development Environment and testing tools |  |  |  |  |
| 1. Frameworks chosen for MVC and UI development |  |  |  |  |
| 1. Target platform and operating system |  |  |  |  |
| 1. Application development templates to be used for MVC or user interface development |  |  |  |  |
| 1. Provided a list of all application functionality requirements including a diagram of the data models required to support the developed application |  |  |  |  |
| 1. Created and documented conceptual designs of the following: |  |  |  |  |
| 1. User interface navigation methods |  |  |  |  |
| 1. 4 different User Interface pages or forms |  |  |  |  |
| 1. A colour scheme for the application that aligns with the personalisation requirements |  |  |  |  |
| Part 2 - User Interface Design Plan | | | | |
| 1. Prepared a mock email to the client containing:   The conceptual designs or sketches created in Part 1.4  a request for feedback on the proposed designs and schemes |  |  |  |  |
| 1. Recorded all feedback provided by the teacher as required adjustments |  |  |  |  |
| 1. Developed a user interface plan that contained the following detail: |  |  |  |  |
| 1. A sketch or wireframe for each planned page |  |  |  |  |
| 1. A diagram describing the connection and navigation between each page |  |  |  |  |
| 1. A description of the planned interaction design pattern techniques that will be implemented in the application |  |  |  |  |
| 1. At least 2 Client-Side validation techniques |  |  |  |  |
| 1. Detail of the planned user interface customisability |  |  |  |  |
| 1. A description of the graphics to be created, and the multimedia to be included in the project |  |  |  |  |
| Part 3 - Application Development | | | | |
| 1. Created a new project using an IDE, Language, Framework and Template outlined in Part 1.2 |  |  |  |  |
| 1. Developed and Implemented data models required for the application ensuring: |  |  |  |  |
| 1. All data models align with the plan created in Part 1.3 |  |  |  |  |
| 1. All data models are compatible and align with the selected IDE, Language, Framework and Template |  |  |  |  |
| 1. Created MVC controllers to provide the functionality outlined in Part 1.3 ensuring at least one controller provided the following Endpoints accessible via client-side scripts: |  |  |  |  |
| 1. HTTP GET |  |  |  |  |
| 1. HTTP POST |  |  |  |  |
| 1. HTTP PUT |  |  |  |  |
| 1. HTTP DELETE |  |  |  |  |
| 1. Created views to provide the functionality outlined in Part 1.3 that align with the user interface plan created in Part 2 including the following views/pages (in any combination): |  |  |  |  |
| 1. 2 views/pages that demonstrate data being received and rendered dynamically, using model binding or Client-Side scripting |  |  |  |  |
| 1. 1 view/page providing the functionality required to allow for the user customisation options identified in Part 2.3.e |  |  |  |  |
| 1. 1 view/page that displays created and included graphics and included multimedia content |  |  |  |  |
| 1. 1 view/page that aligns with the user interface personalisation requirements identified in Part 1.4.c |  |  |  |  |
| 1. Views/pages that demonstrate 2 different interaction design patterns |  |  |  |  |
| 1. View/pages implementing client-side validation for all user input scenarios and as per outlined in the user interface plan created in Part 2 |  |  |  |  |
| 1. Views/Pages that demonstrate the usage of CSS concepts to mathematically manipulate the position of items in a view |  |  |  |  |
| 1. Included screenshots of the developed models, controllers and views/pages in a technical report |  |  |  |  |
| 1. Connected models to controllers and controllers to views to provide the application functionality outlined in Part 1.3 |  |  |  |  |
| Part 4 - Testing + Validation | | | | |
| 1. Created a testing table that covers all the applications required functionality |  |  |  |  |
| 1. Ran and tested the application ensuring: |  |  |  |  |
| 1. The steps or processes used to test the functionality is included in the testing table |  |  |  |  |
| 1. The outcome of each test is recorded in the testing table |  |  |  |  |
| 1. A summary of the testing conducted is provided, including detail around the tools used to test the user interface and application functionality |  |  |  |  |
| 1. 2 screenshots of usage of testing tools are included, 1 from the development environment and another from a browser |  |  |  |  |
| 1. Tested communication between controllers and views, providing screenshots of the following: |  |  |  |  |
| 1. Successfully navigating from a page within the application to another page that renders dynamic data |  |  |  |  |
| 1. The code inside a controller that handles a request from a view paused in a debugger (or equivalent) |  |  |  |  |
| 1. The code inside a controller that handles retrieving model data and returning the data to a view paused in a debugger (or equivalent) |  |  |  |  |
| 1. The rendered view displaying dynamic data retrieved form the model by the controller |  |  |  |  |
| 1. Created a mock email to the client to provide details and images of the developed user interface and functionality including: |  |  |  |  |
| 1. Screenshots of the developed user interface |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. A description of the personalisation and customisation applied to the developed application |  |  |  |  |
| 1. A request for feedback on the user interface and user experience |  |  |  |  |
| 1. Recorded all feedback provided by the teacher on the content within the email in the Technical Report |  |  |  |  |
| 1. Implemented any changes required as per the feedback given by the teacher |  |  |  |  |
| Part 5 - Handoff | | | | |
| 1. Prepared an email to the client to confirm updates and improvements as a result of the communication in Part 4.4 and Part 4.5 |  |  |  |  |
| 1. Created appropriate internal documentation throughout the application in line with the provided Programming and Documentation specifications |  |  |  |  |
| 1. Completed a Sign-Off sheet to indicate the project is complete, listing the items to be provided to the client |  |  |  |  |
| 1. Saved the source code and relevant documentation in line with the provided Programming and Documentation specifications |  |  |  |  |