|  |
| --- |
| **Stimulus: *Client server architecture*** |
|  |

Digital Technologies project: folio

**Instructions:**

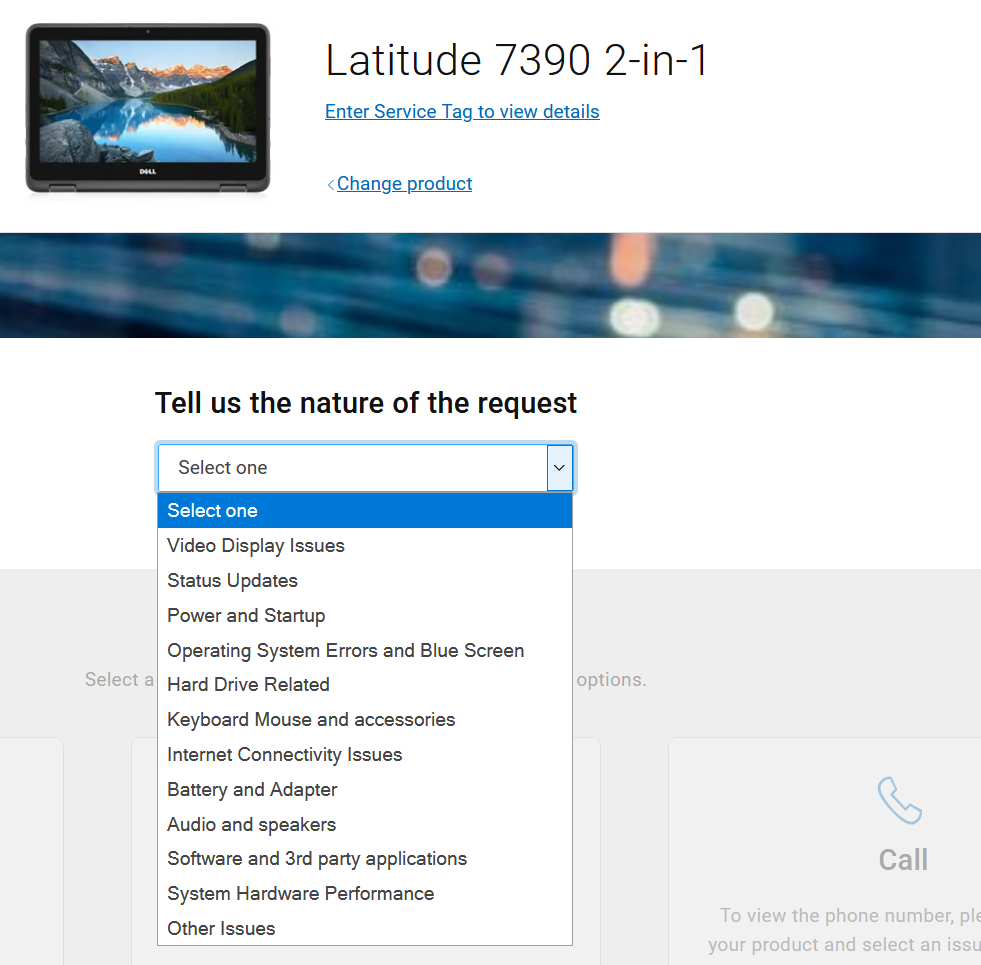
Complete a selection of tasks depending on your skill level, interest, and time available.

**There is no set number of tasks you must complete**. There is a criteria sheet at the end of this which will be used to make a holistic assessment of your production skills.

Your grade will be determined by your ability to be **purposeful**, **proficient,** or **effective** with your coding skill. Quantities of spaghetti code or [copy paste programming](https://en.wikipedia.org/wiki/Copy-and-paste_programming) will not achieve a high award for this folio.

You are encouraged to go 'above and beyond’ and adapt your creative ideas into each task to 'show off' your ability. Please ensure you stay task specific and manage your time effectively. Remember, this is not an assessment of quantity – it is an assessment of your mastery of digital production skill (i.e., coding ability).

There is no written component with this folio, however you must **comment all your written code** to evidence authenticity. Failure to do so will compromise your final award.



**Task 1: Report a fault online for Dell products.**

Dell is seeking an alternative to their fault reporting systems.

Using HTML form elements such as input text fields, text areas, select options, checkboxes, radio buttons or range sliders to design an online fault report form for customers of Dell products. Important information includes customer email, product type (laptop, desktop, or accessory), description of fault, severity (on a scale of 1 to 5), and available contact times.

|  |  |
| --- | --- |
| **Extension** | **Description** |
| **Styling** | Style the form professionally using CSS. |
| **Validation** | Use JavaScript to check the form has been correctly completed before a request is submitted to the server. |
| **Server** | Generate the server-side functionality to capture the report and save it in a variable. |
| **Display** | Generate the server-side functionality to display the report (i.e., the contents of the variable saving the reports), either via console or web interface. |

**Task 2: Online purchase page for super Nvidia graphics card.**

Nvidia has released an RTX 3090 with faster memory and clock speeds. These graphics cards are in high demand and there are only 20 available in Australia.



Develop a webpage that advertises the RTX3090 Limited Edition for sale for $9999 AUD.

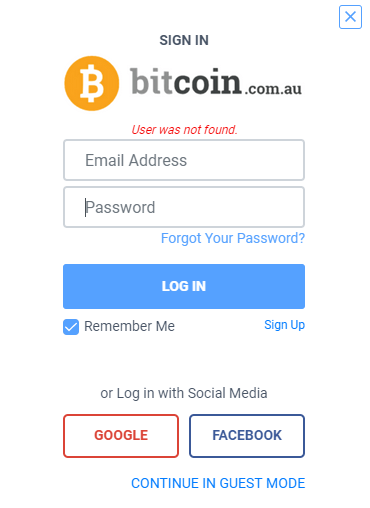
Because of the demand, there is a limit of 1 per customer.

|  |  |
| --- | --- |
| **Extension** | **Description** |
| **Styling** | Style the page layout and aesthetic using CSS. |
| **Purchase** | Capture and save the purchase request server side in a variable. |
| **Administrator** | Allow Nvidia to see all the orders (i.e., the variable used to save the orders) |
| **Stock** | Count the number of purchases. If the number of purchases exceeds the stock available, redirect the customer to a sold-out message. |

**Task 3: Crypto exchange authentication for Bitcoin.com.au.**

Cold (offline) wallets offer the highest level of secure crypto storage, but with exchanges offering the convenience to trade, sell, and buy as well as access crypto wallets, more and more people are willing to take the risk on exchanges. Bitcoin.com.au are seeking prototype front end UI for authenticating users prior to authorising their access to their crypto wallets.

Develop an authentication page user interface that requires email and password at minimum:



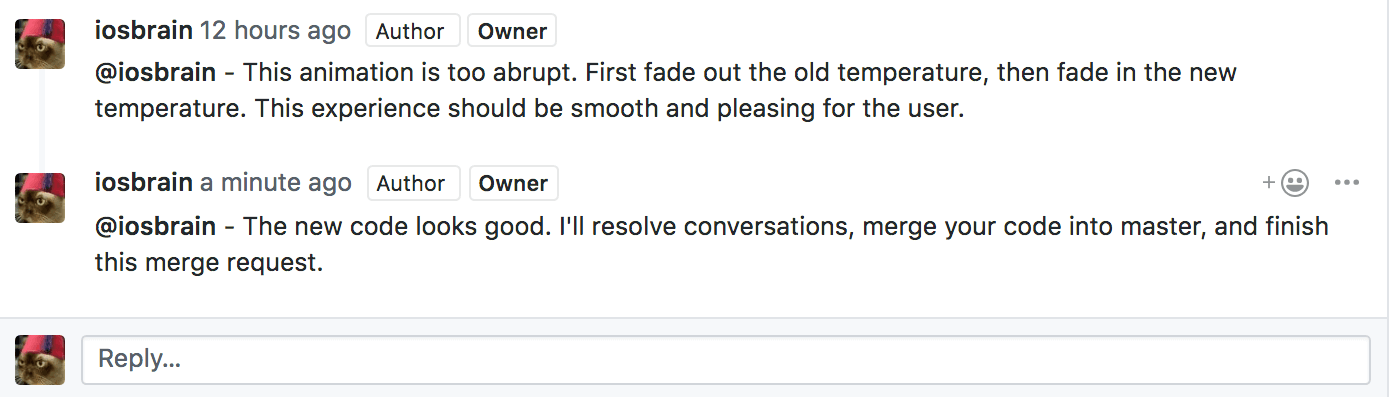
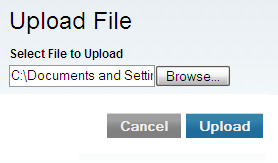
|  |  |
| --- | --- |
| **Extension** | **Description** |
| **Styling** | Style the page minimally using CSS. |
| **Authentication** | Develop the server capacity to check the email and password against a stored email and password in memory.  If the email is incorrect, the response should be akin to "email not found."  If the email is correct, but the password is incorrect, the response should be akin to "incorrect password".  If both email and password are correct, the response should return an arbitrary number of Bitcoin in a user’s account. This is all that needs to be displayed after the user has logged in. *An option to then log out would be useful*.  The value of Bitcoin should be a random number somewhere between **0.00000001 and 1 million**. |

**Task 4: WordPress want to try social media.**

WordPress was originally created as a blog-publishing system but has evolved to support other web content types including more traditional mailing lists and forums, media galleries, membership sites, learning management systems (LMS) and online stores.

WordPress is inviting submissions of a prototype comment or image sharing web page, resembling some of the functionality that would be seen on Facebook or Instagram. WordPress is not investigating video sharing platforms yet (so no Tik Tok or YouTube clones).

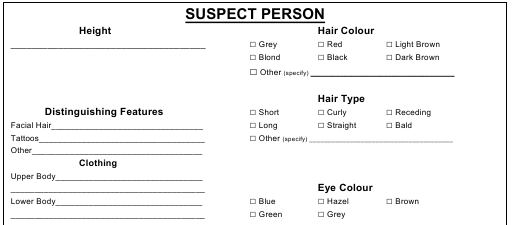
Develop a sample web page that illustrates what a simple comment (blog) style or image sharing style page would look like. Some ideas are shown here:



|  |  |
| --- | --- |
| **Extension** | **Description** |
| **Styling** | Style the page layout and aesthetic using CSS. |
| **Functionality** | Develop the server-side functionality for this image or comment sharing web app. |
| **Moderation** | Enable content moderation features for administrators. |

**Task 5: Criminal composite tool for Queensland Police Service.**

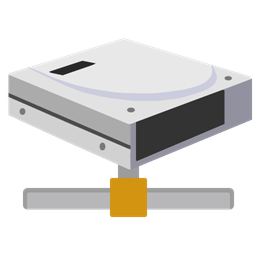
Queensland police in conjunction with Crime Stoppers encourage the public to report details on suspects of criminal activity. To encourage school students and to promote their own Detective work, they have developed some samples targeting younger aged students, that encourages these younger students to develop their recognition and memory retention skills. These samples are shown:



Queensland police have tasked you to develop an online composite tool, for simulated use within schools, populated with fake 'suspects', that enables users to submit details of a suspect and run a hypothetical "search" to see if there is a suspect matching their description.

|  |  |
| --- | --- |
| **Extension** | **Description** |
| **Styling** | Style the page using CSS, remembering the target audience of school students. |
| **Capture** | Develop the server-side functionality to accept and save the suspect details. |
| **Search** | Enable the search of a server-side data structure (variable), using the details submitted to return one or many potential "matches".  You will have to pre-populate this variable with random, fake suspects. |
| **Gamify** | The inspiration for this application has come from the 1985 video game "[Where in the world is Carmen San Diego](https://en.wikipedia.org/wiki/Where_in_the_World_Is_Carmen_Sandiego%3F_(1985_video_game))." A user had to gather clues about the suspects identity.  Gamify your current application, enabling a "treasure hunt" or "web quest" style clue finding system.  The game needs only be prototype or concept only (limited to a single element, find or play-through). |

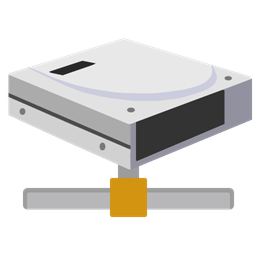
**Bonus task: OneDrive alternative: Network Drive**



This task is *purely for enjoyment* if you finish the previous tasks. It is a task that is far more difficult than the previous tasks, and it is not expected that you will finish this to a high or adequate functioning level.

***You do not have to complete this task*.**

Any attempts to this task, like all tasks, will be evaluated as per the criteria for this assignment.



|  |
| --- |
| Product: **Network Drive** |
| Unique selling point: Just like OneDrive, but only available on *Local Area Network*.   * Production: 127.0.0.1:5000 (local) * Live: 192.168.X.X (LAN address, yet to be determined) |
| Features available via the web page interface:   * Browse available files on a hard disk. * File management:   + Download   + Upload   + *Delete* * Extended file management functionality:   + Rename   + *Move* * File restrictions:   + By type   + By size * Folders:   + create, rename, remove.   + traverse up and down folder structures. * Specific file handle controls, such as:   + Plain text: edit and save (e.g., via textarea)   + Images: preview (resized thumbnail) * Sort – by alphabetical name, file type or size * Search for a file (or part of a file name string) * Generate statistics on file count or hard disk usage. * User help |
| Template file available: [Network Drive template](https://digisoln.com/var/www/work/static/recyclebin/assessment/Digital%20Solutions%20Unit%201%20FIA1%20Network%20Drive.zip) |

END OF ASSESSEMENT

|  |
| --- |
| **Submission Requirements** |
| * Complete each test in a new file. |
| * Submit all commented source code files digitally. |
| **Important Notes** |
| * Use comments in code to explain understanding of programming structures, as well as pointing out refinements and on-going testing of code. |
| * Keep backups of your files. Save every 10-15 minutes of work. |
| * Testing, refinements, or recommendations should be neatly commented in your source code files. |
| **Getting Started** |
| * Look at the examples from class to get an idea how to tackle these challenges. |
| * Look through resources from the website for ideas. |
| * Brainstorm some ideas with your friends or teacher if you cannot figure out how to start. |
| **Authentication Strategies** |
| * Acknowledge all code snippets, tutorials, advice, information, or help given. |
| * Students may be asked to explain their solution, or parts there-of, to determine authenticity. |
| * Please do not share your solutions but rather help students with their own line of work – **you may not necessarily be right or efficient**. |

#### Appendix A: QCAA Years 9 and 10 Digital Technologies standard elaborations (contextualised)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **A** | **B** | **C** | **D** | **E** |
| **Processes and production skills** | *Generating and designing; producing and implementing* | **purposeful** design and **proficient** implementation of modular programs | **effective** design and **effective** implementation of modular programs | design and implementation of modular programs | **partial** design and implementation of modular programs | **fragmented** design and implementation of modular programs |

*This will be marked digitally via the submission platform.*

#### Appendix B: Australian Curriculum content descriptions

This assessment instrument is used to allow students to formally demonstrate the following Australian Curriculum Digital Technologies Years 9 and 10 Content Descriptions:

|  |  |
| --- | --- |
| **Explicitly measured** | |
| P&PS | Implement modular programs, applying selected algorithms and data structures including using an object-oriented programming language |
| **Implicit to the task** (not formally measured) | |
| K&U | Investigate the role of hardware and software in managing, controlling, and securing the movement of and access to data in networked digital systems |
| K&U | Analyse simple compression of data and how content data are separated from presentation |
| P&PS | Develop techniques for acquiring, storing, and validating quantitative and qualitative data from a range of sources, considering privacy and security requirements |
| P&PS | Analyse and visualise data to create information and address complex problems, and model processes, entities and their relationships using structured data |
| P&PS | Define and decompose real-world problems precisely, taking into account functional and non-functional requirements and including interviewing stakeholders to identify needs |
| P&PS | Design the user experience of a digital system by evaluating alternative designs against criteria including functionality, accessibility, usability, and aesthetics |
| P&PS | Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases |
| P&PS | Evaluate critically how student solutions and existing information systems and policies, take account of future risks and sustainability, and provide opportunities for innovation and enterprise |
| P&PS | Create interactive solutions for sharing ideas and information online, taking into account safety, social contexts, and legal responsibilities |
| P&PS | Plan and manage projects using an iterative and collaborative approach, identifying risks, and considering safety and sustainability |

**Key**:

K&U: Knowledge and Understanding

P&PS: Processes and Production Skills