

Assignment 1: Innovation & Project Management.

Student Name: Jacob Moseley

Student Number: 2317034

Course: BSc Software Engineering

Module: Changemakers: Creativity and Value Creation

Module Code: ACCA5033

E-Portfolio URL: https://uowtsd-my.sharepoint.com/:o:/g/personal/2317034_student_uwtsd_ac_uk/IgDOUy6TpHqdQ5RVk3UEK86tAf-MC8jlG4_GXC1uSUNIV1g?e=XJOkJT

Task 1: Ideation & Innovation

Innovation:

Innovation plays an essential role within modern society, enhancing and improving the way we interact with technology. Innovation can be defined as both the introduction and implementation of new or improved ideas, products and processes, solving existing problems or improving current solutions [1]. Over the span of centuries, technology has transformed society, from early industrial machinery to modern technologies spanning the globe.

This rapid advancement of technology is a product of innovation, leading to what is known as the Fourth Industrial Revolution. The 4th Industrial Revolution can be described as the advent of cyber-physical systems, introducing entirely new capabilities for both humans and machines. Within this new age, digital technologies, artificial intelligence and interconnected systems shape how people interact, work and live [2].

Ideation:

When considering potential product ideas in relation to both personal interests and computing, several concepts were explored. One area of interest is online gaming and the associated digital economies. Counter-Strike 2 (CS2), a competitive multiplayer first-person shooter contains one of the largest digital markets within a video game, with a market cap of over six billion dollars [3]. Within CS2, cosmetic skins are randomly obtained through gameplay with varied rarities, patterns and physical wear. As no two skins are ever identical, players are often willing to pay extremely high prices for certain items.

Several potential product ideas in relation to the CS2 marketplace included:

1. **A CS2 Market Analysis Tool:** Enabling users to view all tradable in-game items, track prices over time, and identify the third-party marketplace with the lowest prices.
2. **A CS2 Skin Viewer:** Enabling users to customise skins and view them within a game server, allowing users to view a crafted weapon before permanently altering it within their inventory.
3. **A CS2 Trading Platform:** A platform providing real-time analytics of prices across multiple third-party marketplaces while enabling peer-to-peer (P2P) trading between users.
4. **A CS2 Inventory Valuation Tool:** Enabling users to view their inventory and its value along with individual item prices, taking into consideration applied cosmetics such as stickers and charms.

Of these ideas, the fourth concept was selected as it solved a real problem within the CS2 market while borrowing aspects from the other entries within the list. The marketplace provided by Valve, the developer of CS2, lacks many essential features, including the valuation of a player's inventory and the accurate pricing of items with applied cosmetics. These limitations have resulted in many users unintentionally selling items below their true market value. This proposed web application aims to address this issue by analysing a player's inventory and estimating the value of individual items while taking into account factors such as applied stickers, patterns and wear. In addition, the web application will also include real-time price trends and inventory analytics with the aim of helping users better understand the value of their digital assets.

Project Title: Counter-Strike 2 Inventory Valuation Tool

Aim: Develop a web-based application that analyses user inventories and estimates the market value of digital items using external marketplace data.

Objectives:

- Design a web interface capable of displaying a user's inventory data.
- Fetch a user's inventory data using the Steam Web API.
- Implement a caching system for user inventories in order to avoid rate limiting.
- Develop an algorithm capable of estimating item value considering attributes such as float value,

pattern variation and applied cosmetics.

- Research and implement historical pricing data.
- Implement inventory analytics and visualisation in order to track price trends.

Research:

In order to support the development of the selected project, several key courses and sources have been identified:

- **Steam Web API Documentation:** Essential to the implementation of the project, outlining how developers can retrieve player inventory data.
- **LinkedIn Learning:** A course provider covering Node.js and web development. Courses such as Node.js Essential Training provide guidance on building a backend capable of processing API requests and communicating with potential databases.
- **Figma Learn:** Providing tutorials and documentation regarding the design of user interfaces. These resources will support the design phase of the project by helping to develop an intuitive interface for the web application.
- **Skinport API Documentation:** Skinport provides accurate pricing data for CS2 items via its marketplace API. The documentation will be used as guidance when retrieving price data and implementing the estimation algorithm.

Task 2: Project Management

Project Scope:

The proposed concept is a web-based application aiming to provide an accurate valuation of a user's Counter-Strike 2 Inventory. Users will first enter their unique Steam64 ID, which will be used to retrieve inventory data via the Steam Web API. The retrieved items will then be compared against marketplace pricing data containing the real-world value of each item. These prices will then be adjusted by considering attributes such as float value, pattern ID and applied cosmetics (e.g. stickers or charms). The calculated results will then be displayed within the web browser.

In addition, the application will also provide analytics, including the weekly change of overall inventory value and visualised graphs displaying price trends for individual items. Filtering options will be implemented, enabling users with larger inventories to locate individual items more effectively. Automated item preview screenshots using in-game bots may also be explored, however this feature is considered outside the scope of the initial prototype. Additionally, Steam OAuth authentication may be implemented, allowing users to securely log in and retrieve their inventory data without manually entering their Steam64 ID.

Several technologies and resources will be used to achieve this implementation, these include:

Software:

- **HTML:** For structuring the web application.
- **CSS:** For styling the web application.
- **JavaScript:** For frontend development, e.g. generating cards for individual items.
- **Node.js:** For backend development, e.g. retrieving inventory data via the Steam Web API.
- **JSON:** The primary data format when communicating between APIs and the application [4].
- **Steam Web API:** For the retrieval of both Steam user and inventory data.
- **Skinport Market API:** For the retrieval of accurate current and historical pricing for individual items.
- **Figma:** Used to design the web application's user interface.
- **GitHub:** Used for version control and project organisation during development.
- **SQL Database:** Used to cache inventory, pricing and user data in order to avoid API rate limits.

Documentation:

- **Steam Web API documentation:** Used as guidance when implementing user and inventory retrieval.
- **Skinport API documentation:** Used as guidance when implementing item value retrieval.

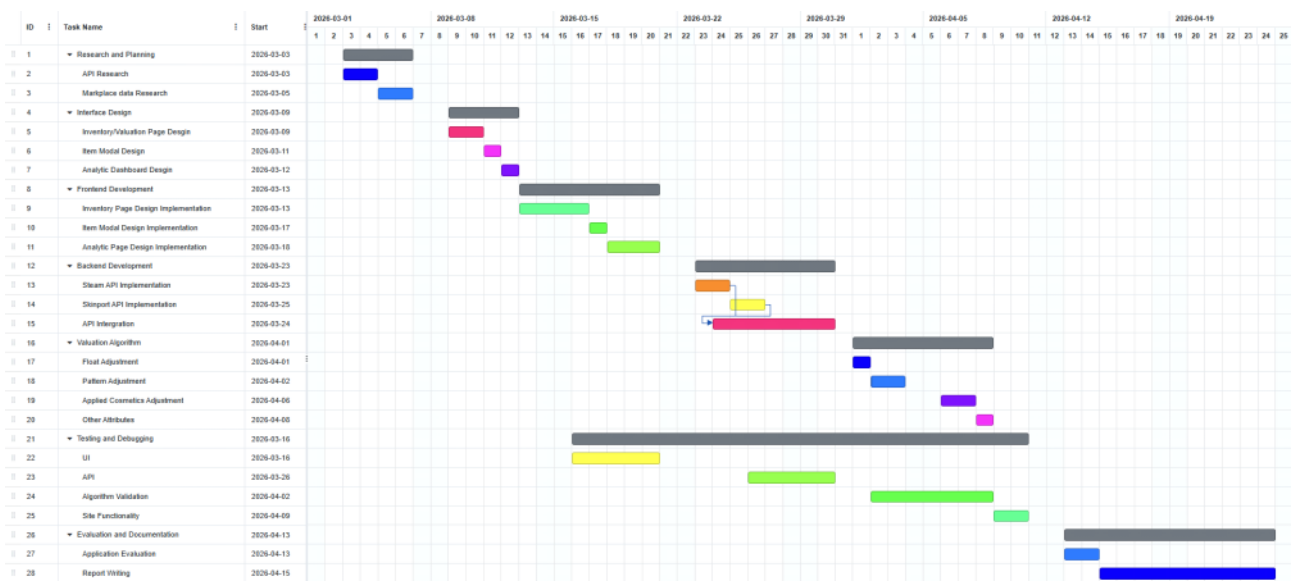
The implementation of the CS2 Inventory Valuation Tool is expected to run from the beginning of March till mid-April. The early stages will be focused on research and interface design. Development and testing will occur within the middle of this timespan, with a focus on evaluation and documentation in the final weeks.

The key stages of the projects implementation include:

1. Research and planning
2. Interface design (Figma)
3. Frontend development
4. Backend development
5. Pricing estimation algorithm implementation
6. Testing and debugging
7. Evaluation and documentation

Gantt Chart:

Using the project scope as guidance, a Gantt chart was designed as illustrated in Figure 1. Gantt Charts are a popular project management tool, providing effective means of visualising task scheduling [6]. Within this timeline, subtasks were included in order to provide a detailed breakdown of the individual stages.



[Figure 1: Gantt Chart]

Risks and Challenges:

There are several risks and challenges facing the implementation of the CS2 Inventory Valuation Tool. This area aims to identify them while discussing potential mitigation strategies.

Steam Web API Rate Limiting: While Steam's official API is stated to have a rate limit of 100,000 requests per day [7], some endpoints, such as the inventory call, are not part of the official API, and therefore are heavily rate limited. This not only slows development and implementation, but may also affect the performance of the final application. To mitigate this, a caching system will be implemented in order to reduce the number of API requests.

Pricing Data Accuracy: The prices of items within the CS2 market fluctuate frequently. If inaccurate pricing data is used, the estimated inventory value may not accurately reflect the market value. To reduce this risk, price data will be updated at regular intervals throughout the day. This approach also reduces the risk of Skinport API rate limiting, as a single request can be retrieved and used for multiple valuation calculations.

Scope Management:

There is a risk that additional features, such as automated in-game item screenshots, will expand the scope of the project beyond what is possible within the given timeframe. To manage this risk, development will focus on producing a minimum viable product (MVP), prioritising the applications core functionality.

Task 3: Project Development Method and Design Work

Development method:

For the implementation of the CS2 Inventory Valuation Tool, prototyping has been chosen as the development methodology. Prototyping is a software development life cycle model where an initial prototype is built, tested and reworked. This approach creates a base, that is built upon in order to achieve the final system or software [8].

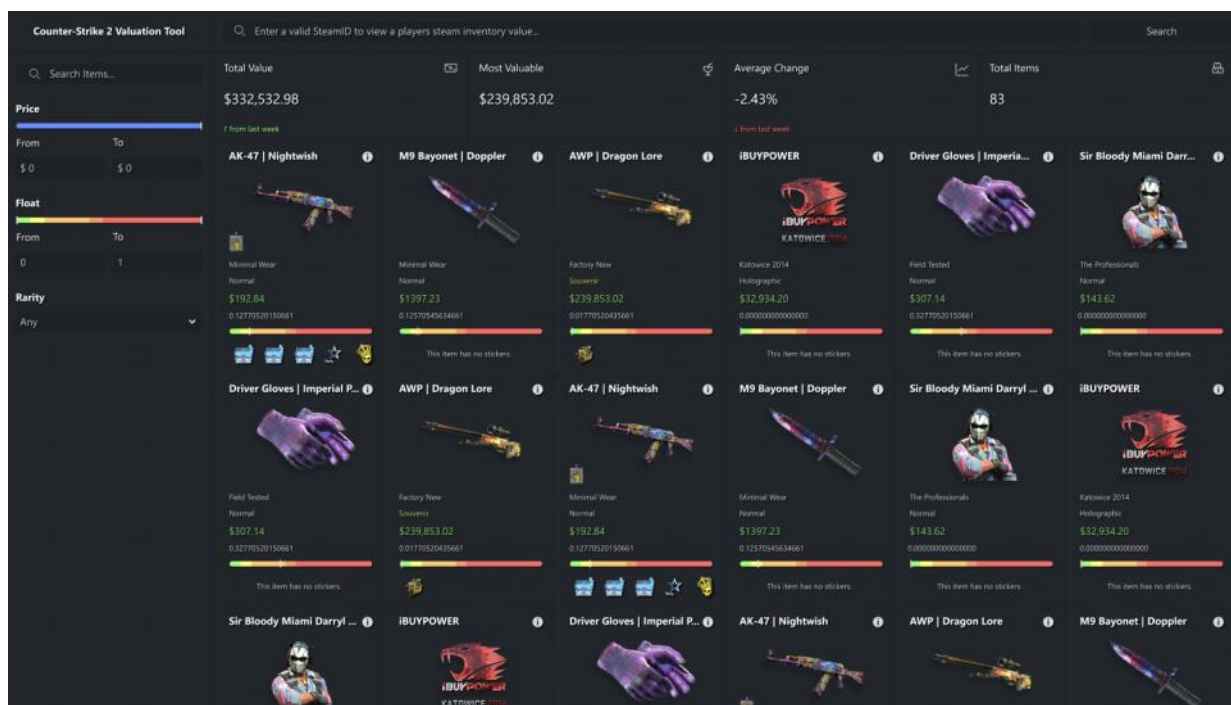
This methodology is well suited to the selected project, allowing the core functionality of the application to be built in stages. Initial development will focus on producing a minimum viable product (MVP), including the design of essential user interface components, inventory retrieval, price data retrieval and calculating the value of items. Once these core features are developed, focus can then shift to implementing additional features such as filtering, analytics and automated in-game screenshots.

By utilising prototyping, the initial implementation can be developed quickly, reviewed regularly and revised based on feedback from the end user [9]. This ensures that the final application meets and improves upon the original project objectives.

Design Work:

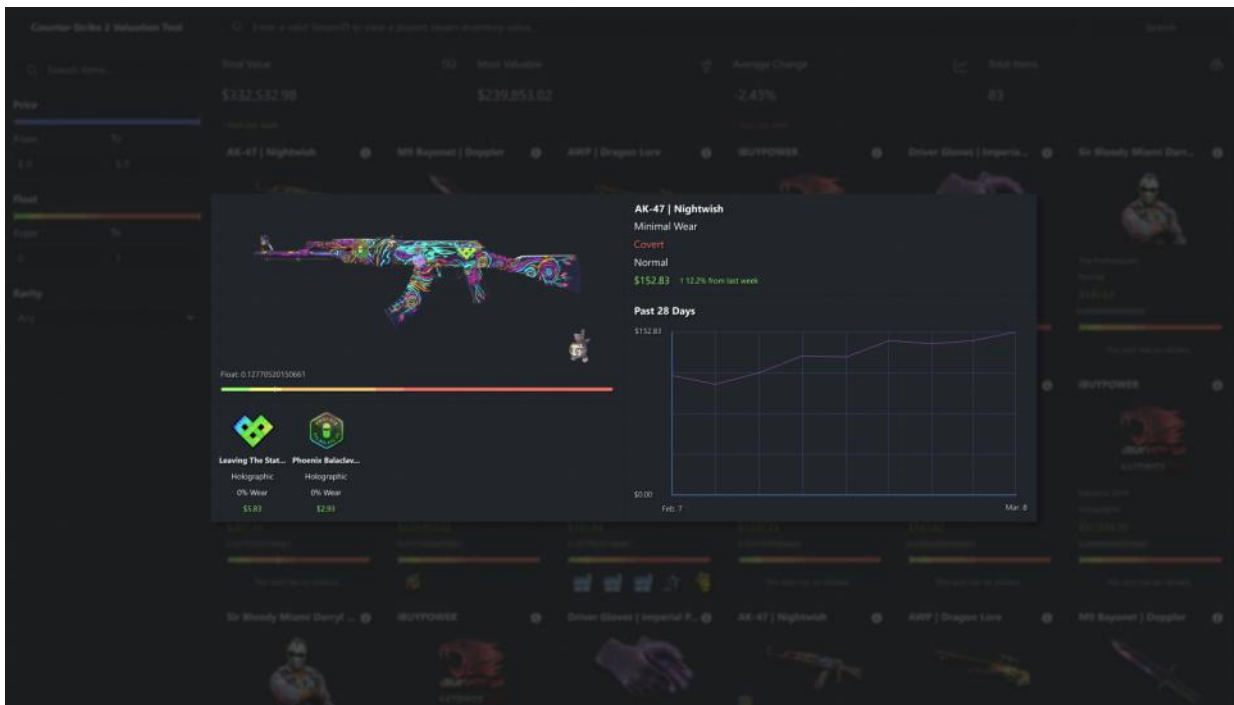
To support the development of the CS2 Inventory Valuation Tool, Figma was used to create designs for the user interface. These designs represent the intended visual layout of the application prior to implementation and are subject to change.

The first design as shown in Figure 2, illustrates the inventory overview webpage, displaying a user's items within a card-based layout. Each card is designed to display key information pertaining to an individual item. This includes attributes such as name, rarity, float and estimated value. Additional features such as filtering and inventory analytics were also considered and placed above these cards, with a focus on creating a user-friendly interface.



[Figure 2: Inventory Page UI Design]

A design for a modal containing further item information was also created as shown in Figure 3. This modal displays additional information, including sticker value and a graph visualising an item's price history.



[Figure 3: Item Modal UI Design]

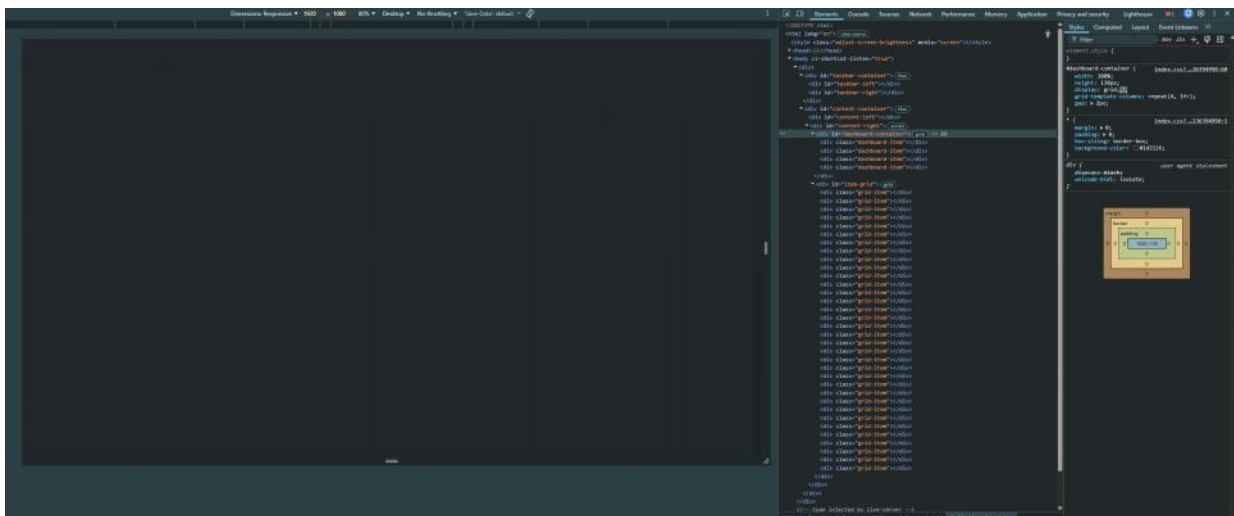
A requirement specification was also created in order to clearly define what the application must do and the constraints under which it must operate before development begins [10]. As shown in Figure 4, each requirement was assigned a priority level, helping to identify which features are important to the initial implementation. This is particularly useful when producing the minimum viable product (MVP).

ID	Title	Priority	Description	Category
FR-01	Retrieve Steam Inventory	High	Must retrieve user inventories via the Steam Web API.	Functional
FR-02	Display Inventory Items	High	Inventory items must be displayed within a card layout.	Functional
FR-03	Inventory Valuation	High	Application must calculate and display the value of an inventory and its items.	Functional
FR-04	Item Modal	Medium	Users should be able view detailed item information within a modal.	Functional
FR-05	Inventory Filtering	Medium	Users should be able to search and filter items within their inventory.	Functional
FR-06	In-game Screenshots	Low	In-game screenshots may be implemented, provided high resolution images of items.	Functional
FR-07	Caching User Data	High	User data must be cached to avoid rate limits.	Functional
NFR-01	Preformance	Medium	A user's inventory should load in a reasonable amount of time	Non-Functional
NFR-02	Usability	High	The UI must be intuitive.	Non-Functional
SEC-01	Data Protection	Low	All user data is publically available and does not require encryption.	Security
SEC-02	Authentication	Low	Users may be able to authenticate using Steam Oauth.	Security
SEC-03	API Keys	High	API keys must be stored securely.	Security
TST-03	System Testing	High	Application should be tested to ensure it is functioning as intended.	Testing

[Figure 4: Requirement Specification]

Initial Practical Implementation:

An initial prototype of the inventory page was developed using HTML and CSS, translating the original Figma design into a working implementation. As shown in Figure 5, a basic frontend layout was developed, with cards for displaying inventory analytics and individual items.



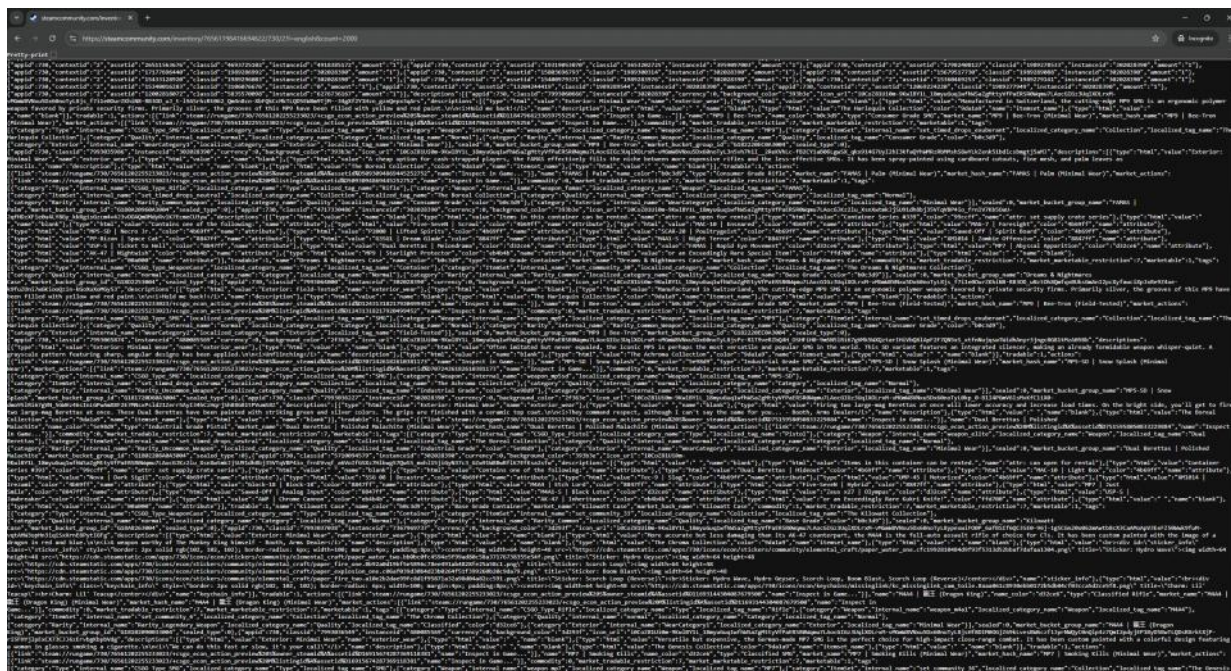
[Figure 5: Initial UI Prototype]

Cards were implemented using a CSS grid as shown in Figure 6, enabling the implementation of a function and user-friendly design. This grid will later contain user items and their associated attributes.

```
74 #item-grid {
75   width: 100%;
76   height: calc(100% - 140px);
77   display: grid;
78   grid-template-columns: repeat(8, 1fr);
79   gap: 2px;
80   margin: 2px 0 0 0;
81 }
82
83 .grid-item {
84   height: 358px;
85   background-color: #21252b;
86 }
87
```

[Figure 6: CSS Grid Code]

The Steam inventory endpoint was also tested within the browser in order to understand the structure of the returned data. As shown in Figure 7, the endpoint returns inventory data in JSON format. This data includes float values, item names, descriptions and applied cosmetics such as stickers. This test provides important context for when Node.js backend development begins.



[Figure 7: Steam Inventory Endpoint Call]

Price data for individual items was also retrieved and saved to JSON within the project file using the Skinport API as shown in Figure 8. This data will initially be used to structure the applications UI and test Steam inventory integration. Further into the valuation tool's development, this data will be moved to a SQL database and updated periodically to ensure calculated prices are both current and accurate.



[Figure 8: Saved Skinport Price Data]

References:

- [1] OECD, "Oslo Manual 2018," OECD, 2018. https://www.oecd.org/en/publications/2018/10/oslo-manual-2018_g1g9373b.html (Accessed: Mar. 4)
- [2] "How Innovation Shapes the World," Unit 1: What's the Problem?, course learning material, University of Wales Trinity Saint David (Accessed: Mar. 4)
- [3] "Counter-Strike Market Cap | Live CS2 Skin Prices," Pricempire.com, 2025. <https://pricempire.com/app/trending> (Accessed: Mar. 4)
- [4] "When would you use JSON? - Tencent Cloud," Tencentcloud.com, 2025. <https://www.tencentcloud.com/techpedia/101698> (accessed Mar. 10, 2026) (Accessed: Mar. 10)
- [5] GitHub, "About GitHub and Git," GitHub Docs, 2024. <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>
- [6] J. M. Wilson, "Gantt charts: a Centenary Appreciation," European Journal of Operational Research, vol. 149, no. 2, pp. 430–437, Sep. 2003, doi: [https://doi.org/10.1016/S0377-2217\(02\)00769-5](https://doi.org/10.1016/S0377-2217(02)00769-5) (Accessed: Mar. 7)
- [7] "Steam Community :: Steam Web API Terms of Use," steamcommunity.com. <https://steamcommunity.com/dev/apiterms> (Accessed: Mar. 7)
- [8] U. Senarath, "Waterfall Methodology, Prototyping and Agile Development," Waterfall Methodology, Prototyping and Agile Development, Jun. 2021, doi: <https://doi.org/10.13140/RG.2.2.17918.72001> (Accessed: Mar. 9)
- [9] University of Wales Trinity Saint David, "Changemakers: Creativity & Value Creation – Assignment 1: Task 3 Project Development Method & Design Options," lecture slides, University of Wales Trinity Saint David, 2026 (Accessed: Mar. 9)
- [10] OpenAI's ChatGPT, private communication, Feb. 2023 (Accessed: Mar. 9)