

Project Management Project - Documentation Task 2

Requirements

For this JavaScript project, I will design and build a web browser interactive game based on the Dungeon and Dragons's concept. The user will be able to create a character from a selection of multiple class roles. They will also have the opportunity to power up their characters by collecting items into their inventory and by doing so will increase their success in defeating the final dungeon boss. Upon entering combat, the user will roll a dice to be randomly assigned a starting power level which will determine the damage which can be inflicted per turn.

Multi-media will be integrated throughout the program to provide a more interactive and sensational role-playing experience.

System Requirements

Hardware: MS Windows

Browser: Supports JavaScript and multi-media (e.g. Chrome, Firefox, Edge)

Memory: Minimum 2 GB RAM

Software: Node.js environment with MongoDB installed (VS Code)

Functional Requirements

Character Creation Page:

- Allow users to create a character with custom class roles.
- Users can select a character's name and role.
- Integrate animations and sound effects during the character creation process.

Inventory Management System:

- Display a backpack where users can store items they found.
- Enable users to pick hidden objects through click events.
- Store inventory data in MongoDB.

Turn-Based Combat System:

- Provide a turn-based battle between the user and a boss monster.
- User can select an ability for combat.
- Allow the player and monster to take turns attacking.
- Include animations, sound effects, and real-time health updates.

Non-Functional Requirements

Performance:

- Load times for any page should not exceed 3 seconds.
- The application should support up to 1 concurrent user.
- The media is shown/played smoothly with each triggered event.

Scalability:

- The system should be scalable to accommodate future game expansions.

Security:

- User's character name is only stored in the computer's temporary memory and is not used or accessible for anything else.

- Protect inventory data in MongoDB with encryption.
- Prevent unauthorised database access.

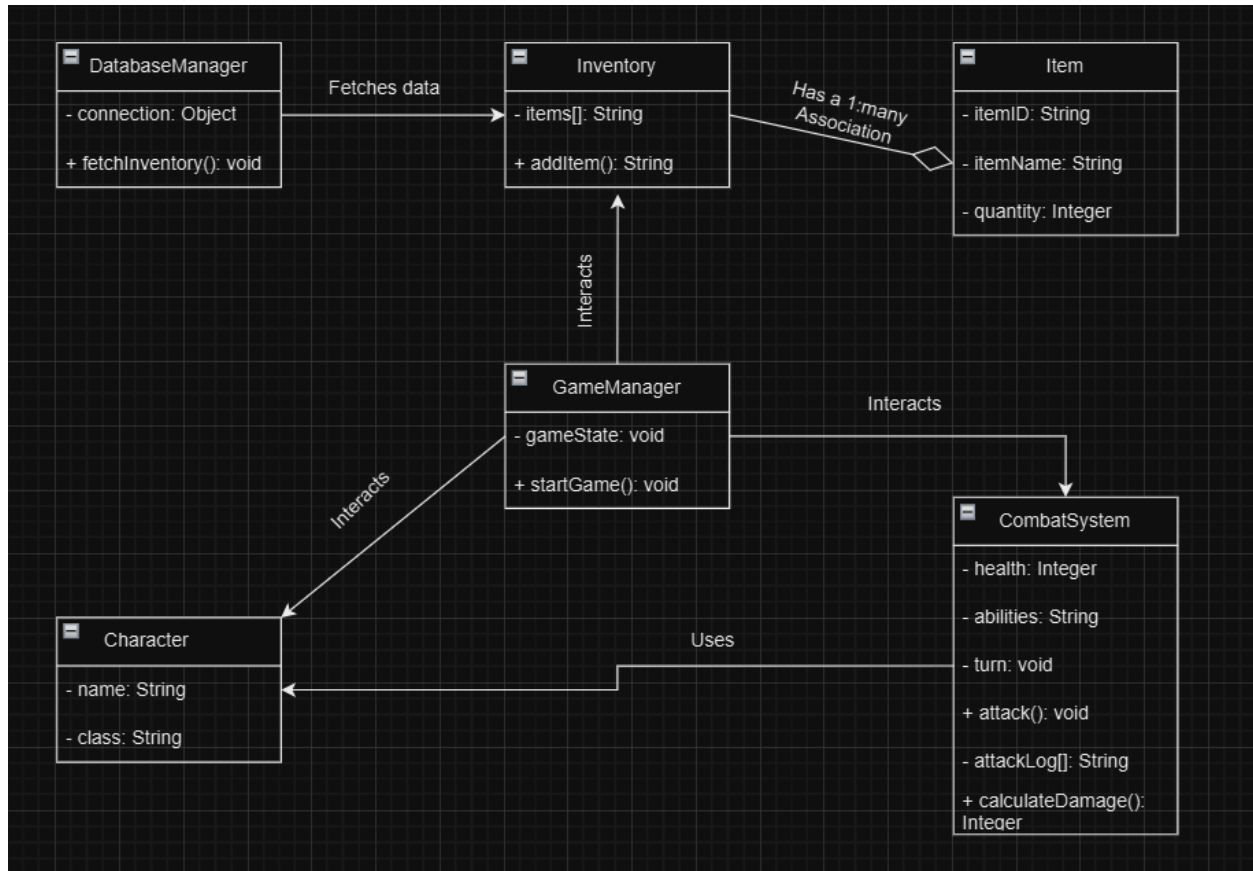
Usability:

- Intuitive user interface with a minimal learning curve.
- Responsive design for compatibility with desktop browsers.

Reliability:

- Ensure smooth integration of all the components for the web application.

Class Diagram



Association:

GameManager interacts with Character, Inventory, and CombatSystem to coordinate gameplay. CombatSystem uses the Character class to determine actions during a turn.

Aggregation:

Inventory aggregates Item, meaning that Inventory holds a collection of items, but the existence of Item objects is independent of Inventory.

Dependency:

DatabaseManager uses libraries like Mongoose to interface with MongoDB. This dependency facilitates saving and retrieving data. There is dotenv dependency for the access to the mongoDB due to security reasons.

Explanation of Predefined Libraries

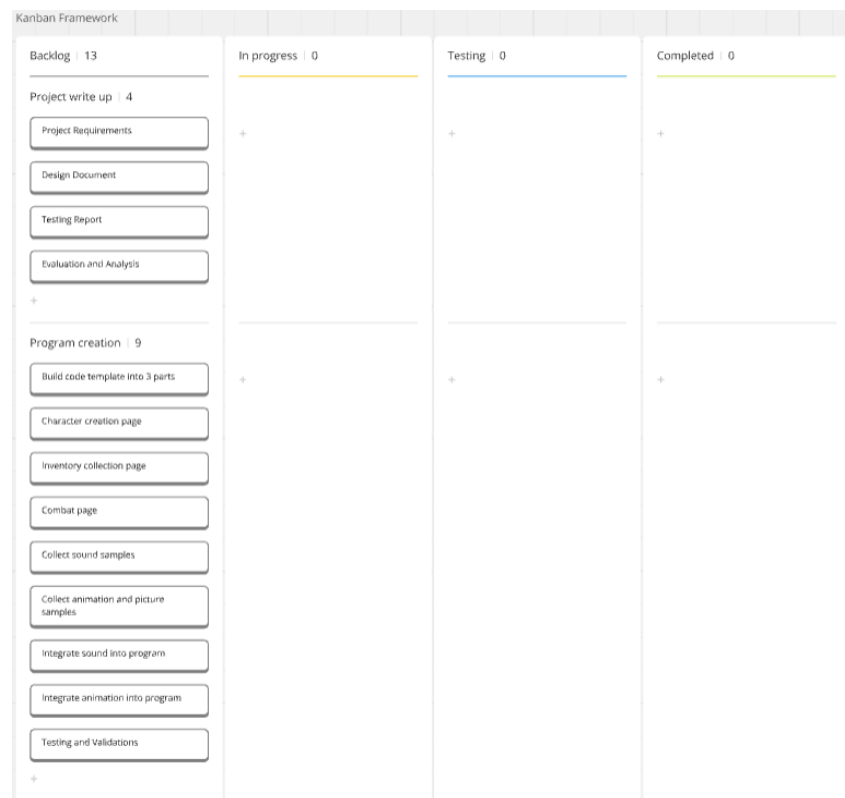
mongoose: Used by DatabaseManager to store and retrieve data such as Character attributes and Inventory items.

express: Powers the web server to manage client requests.

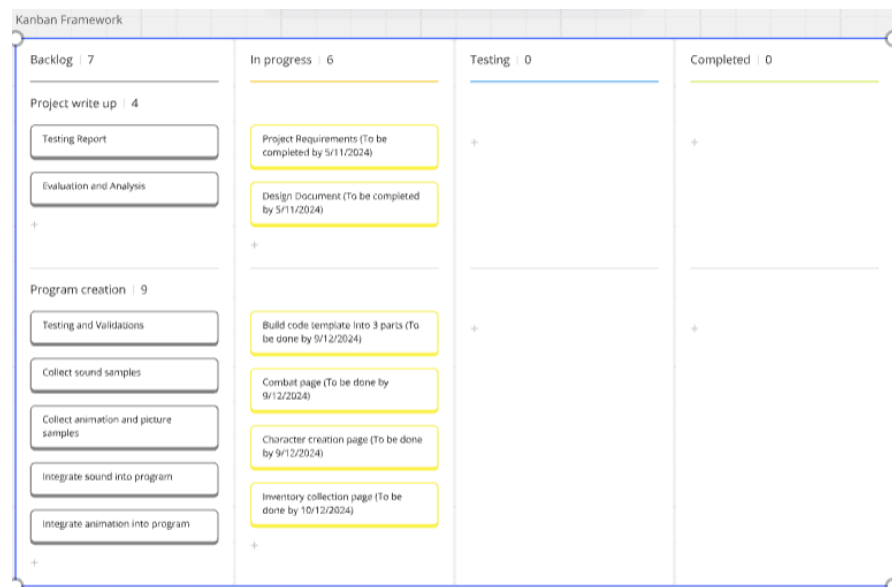
body-parser: Parses incoming requests for character creation and inventory actions.

Kanban Board

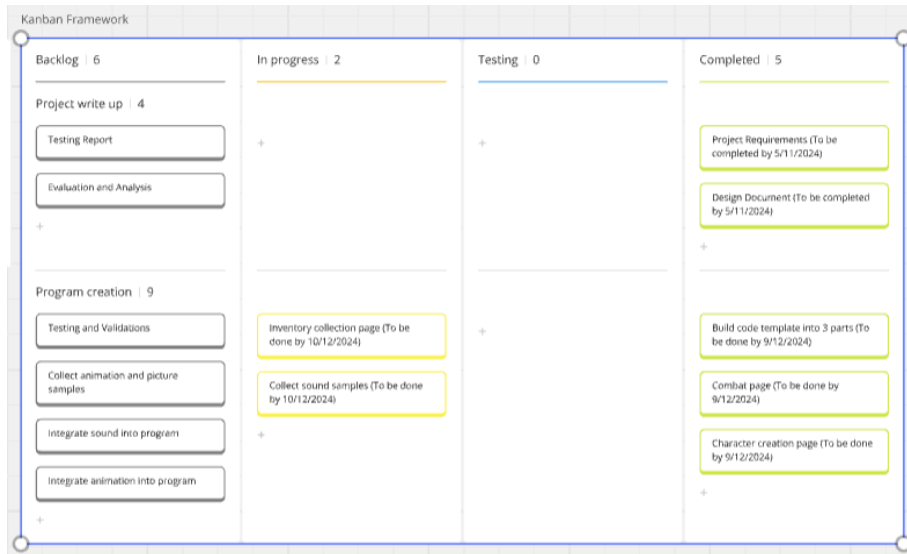
Version 1



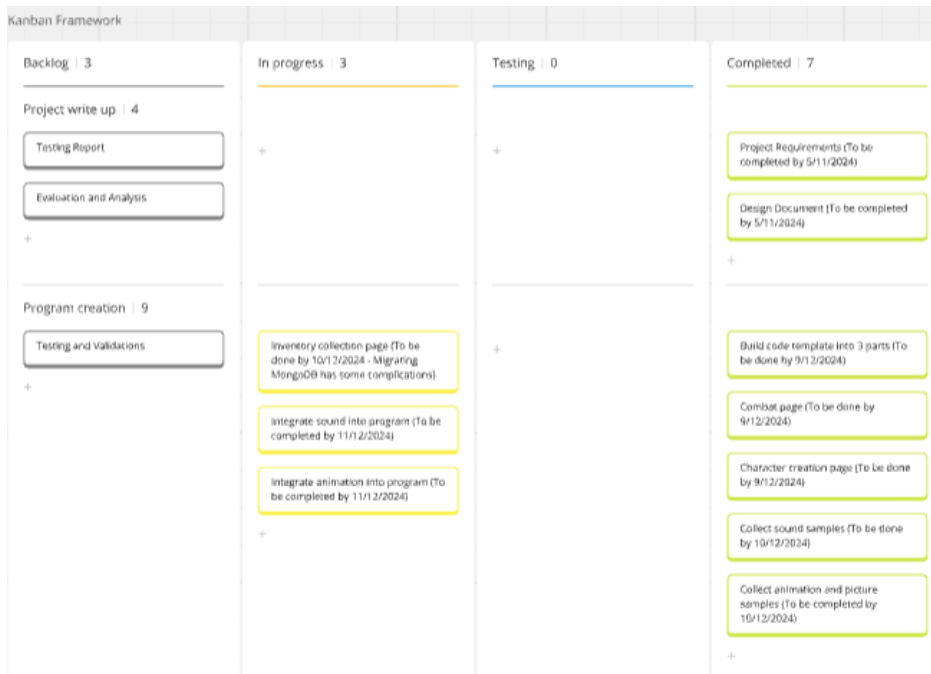
Version 2



Version 3



Version 4



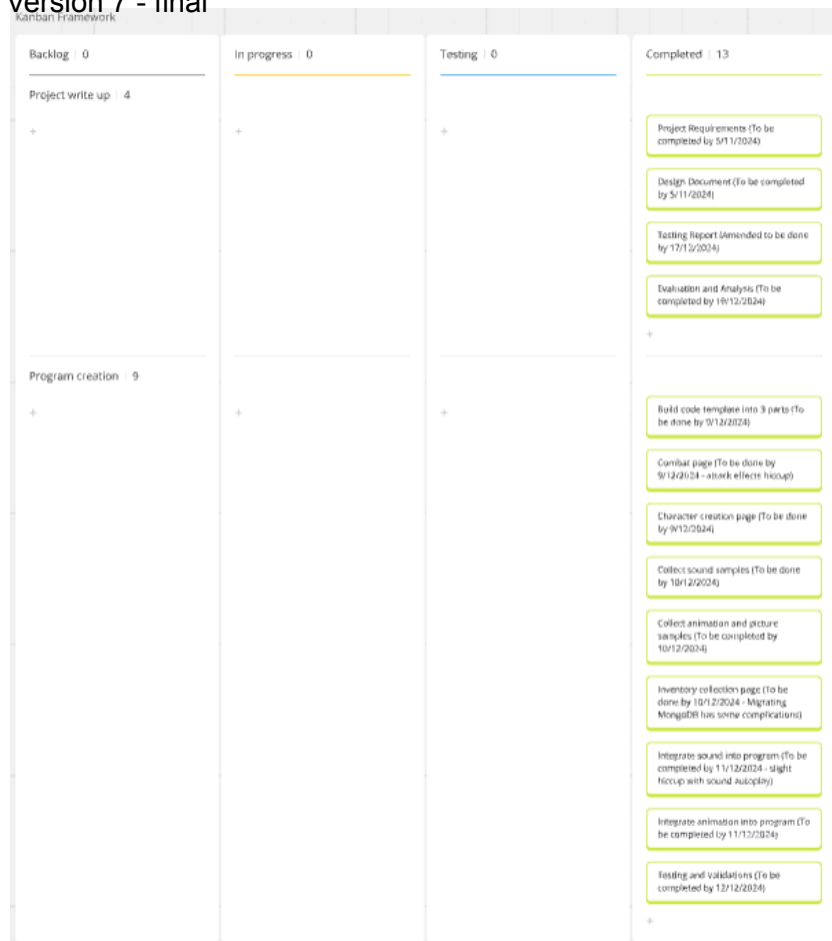
Version 5

Kanban Framework			
Backlog 0	In progress 2	Testing 1	Completed 10
Project write up 4	Testing Report (To be completed by 15/12/2024)		Project Requirements (To be completed by 5/11/2024)
	Evaluation and Analysis (To be completed by 19/12/2024)		Design Document (To be completed by 5/11/2024)
Program creation 9		Testing and Validations (To be completed by 12/12/2024)	Build code template into 3 parts (To be done by 9/12/2024)
			Combat page (To be done by 9/12/2024 - attack effects hiccup)
			Character creation page (To be done by 9/12/2024)
			Collect sound samples (To be done by 10/12/2024)
			Collect animation and picture samples (To be completed by 10/12/2024)
			Inventory collection page (To be done by 10/12/2024 - Migrating MongoDB has some complications)
			Integrate sound into program (To be completed by 11/12/2024 - slight hiccup with sound autoplay)
			Integrate animation into program (To be completed by 11/12/2024)

Version 6

Kanban Framework			
Backlog 0	In progress 2	Testing 0	Completed 11
Project write up 4	Testing Report (Amended to be done by 17/12/2024)		Project Requirements (To be completed by 5/11/2024)
	Evaluation and Analysis (To be completed by 19/12/2024)		Design Document (To be completed by 5/11/2024)
Program creation 9			Build code template into 3 parts (To be done by 9/12/2024)
			Combat page (To be done by 9/12/2024 - attack effects hiccup)
			Character creation page (To be done by 9/12/2024)
			Collect sound samples (To be done by 10/12/2024)
			Collect animation and picture samples (To be completed by 10/12/2024)
			Inventory collection page (To be done by 10/12/2024 - Migrating MongoDB has some complications)
			Integrate sound into program (To be completed by 11/12/2024 - slight hiccup with sound autoplay)
			Integrate animation into program (To be completed by 11/12/2024)
			Testing and Validations (To be completed by 12/12/2024)

Version 7 - final



Project Management Lifecycle

Primary Stakeholders: The course instructors who set the project requirements and grading criteria.

Secondary Stakeholders: End-users and coursemates, represented by peers who would interact with the application to roleplay a fantasy adventure.

Development Team: Myself, acting as the project manager and developer.

Initial Consultation

I began by consulting with the primary stakeholders to confirm the project's objectives: developing a Dungeons and Dragons (DnD) role-playing game with an interactive inventory system. The primary stakeholders outlined the following core requirements:

- A simple and intuitive user interface.
- Functionalities for adding, updating, and retrieving inventory items.
- Integration with MongoDB for inventory storage.
- Immersive features such as audio and animations.

Gathering Feedback

To incorporate the perspective of secondary stakeholders, I shared a prototype and conducted an informal survey. Feedback included:

- A need for a "how-to" guide for game navigation.
- Suggestions for additional features, such as damage modifiers for each class ability and dice-rolling actions between attacks to mimic the original DnD experience.

Incorporating Feedback

Based on this input, I refined the project requirements to include:

- A help menu for better navigation.
- Pop-up descriptors for class abilities that display when the user hovers over them.

Task Management

I utilised a Kanban board to manage tasks across stages: "Backlog," "In Progress," "Testing," and "Completed." Tasks were further subdivided into technical write-ups and hands-on coding, with specific deadlines attached for time management.

Project Stages

- Requirement Gathering and Design
 - Conducted interviews with the primary stakeholders.
 - Created the initial application skeleton code.
- Implementation and Integration
 - This stage was the most challenging and consumed almost half the project duration.
 - Integration of MongoDB with the hiddenObject DOM and combatSystem DOM encountered several errors.
 - An Agile approach with iterative testing and code reviews was used to ensure functionality met the project deadline.
- Audio and Animation Integration
 - Collected sound and animation samples and integrated them into the browser DOM.
 - Faced minor issues with Chrome's autoplay policy, which was resolved by requiring user interaction to trigger audio playback.
- Combat Mechanics and Accessibility
 - Fine-tuned the combat mechanics and CSS properties for accessibility.
- Testing and Error Handling
 - Conducted comprehensive testing, including regression testing, to ensure smooth error handling and improve the turn-based combat mechanic.
 - Added visual effects, such as health bar flashes, to enhance user experience.
 - Addressed a MongoDB connection security concern raised by the primary stakeholder by implementing dotenv for better security management.
- Final Evaluation and Reporting
 - Presented a demo to stakeholders.
 - Collected follow-up feedback for future improvements.

Evaluation

The project was initially designed as a simple interactive DnD game. The final product met all original project specifications while exceeding expectations by incorporating additional features identified during development. Despite instances of scope creep, the stakeholders were impressed by the team's ability to adapt and deliver the requested functionality.

Throughout the Software Development Lifecycle (SDLC), any errors or deviations were promptly communicated to the primary stakeholders, ensuring that all changes aligned with the overall project goals. Time management was key, as I allocated extra time for debugging and refactoring, allowing me to address major issues without impacting the project timeline significantly.

Future Improvements

Stakeholder and alpha tester feedback highlighted the following:

Positives

- The game features a strong foundation with an immersive user experience.
- Numerous functionalities and features enhance the gameplay.

Suggestions for Improvement

- Add a more comprehensive instructional guide for navigating the game, especially the hiddenObject section.
- Introduce additional features such as class ability damage modifiers and dice-rolling mechanics between attacks.

These suggestions will be prioritised for future updates to further enhance the gaming experience.

Testing Report

Introduction

The purpose of the DnD game project is to offer the user a glimpse into the interactive and fantasy world of Dungeons and Dragons.

- They can create an adventurer, and define their personality by choosing a powerful or cool name.
- Tailor their playstyle by choosing from the occupational classes.
- Search the damp and dark caves for hidden treasures.
- Battle and triumph over the dungeon boss.
- Or lick your wounds and try again.

The user will only need to input their character name within a text box on the initial landing page, all other interactions will be conducted by buttons and mouse clicks. They will have the option to toggle the music to increase the intensity. Page navigation will be found at the bottom centre of the screen. The user will mainly interact with the centre of the screen.

For this project, regex has been used to validate that the user's character name can only be alphanumeric input. In addition, normal, abnormal and extreme data have been used to test the upper and lower limits of what should be accepted as a character name. If the user is able to interact with every button and progress from the initial character creation to the final browser

where they can defeat the dungeon boss, the project is deemed ready to proceed to the next stage.

Test Plan

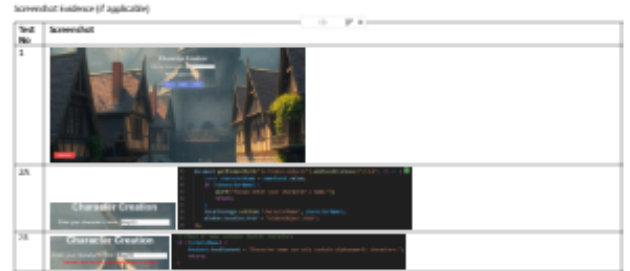
Testing the DnD game project - click the heading for the PDF file.

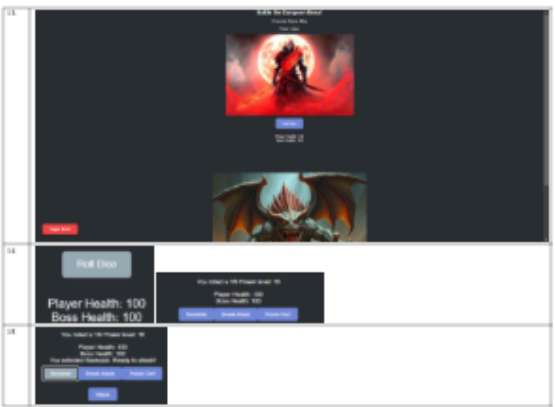
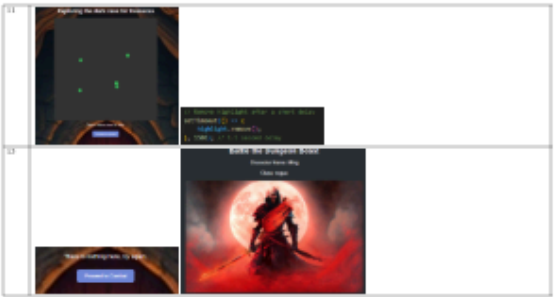
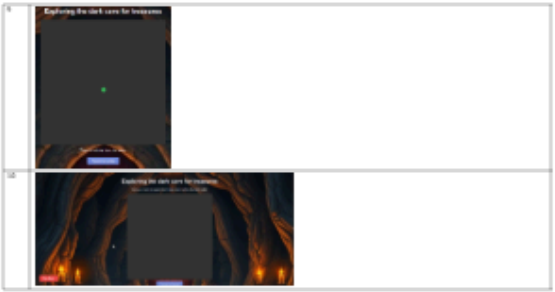
Testing - DnD Game Project

Test No.	Test Name	Description	Test type	Data	Expected outcome	Actual outcome	Action taken / Fixes made
1	Character creation	Landing page	Normal	None	The character creation landing page displays the background image, user input for name, class selection button, toggle music button, and button to go to the next page.	Pass	None needed
2A	Character creation	Does not accept special characters	Abnormal	Ming223	User is advised that special characters are not accepted	Fail	Program code to be amended to validate that special characters will not be accepted and error message advising the user.
2B	Character creation	Does not accept special characters	Abnormal	Ming223	User is advised that only alphanumeric characters are accepted	Pass	None needed
3	Character creation	Accepts alphanumeric characters only	Normal	Ming223	User successfully submits their character name and proceeds to next stage	Pass	None needed
4	Character creation	The name input cannot be blank	Abnormal	NA	User is advised that the character name is required to proceed ahead	Pass	None needed
5	Character creation	The name input must be between 5 and 20 characters	Extreme	Ming2234 567890001 12312	User is advised that the character name must be within range of 5 - 20 characters long	Pass	None needed
6	Music toggle button	User is able to toggle the music on and off	Normal	Click event	Toggle music is shown when the page is loaded, when the user clicks the red button, it changes between Play/Pause music and the music starts and stops accordingly.	Pass	None needed
7	HiddenObject	Landing page	Normal	NA	The hidden object landing page displays the background image, alongside a dark game area to search for hidden items, toggle music button and button to go to the next page.	Pass	None needed
8	HiddenObject	Searching for items by clicking on the game area	Normal	Click event	When user clicks on the game area, a green highlighter appears around that area. If an item is within that space, user is advised they found such item.	Pass	None needed
9	HiddenObject	Clicking on the same area which previously contain an item	Normal	Click event	That item should no longer be available and the user is advised that nothing is there	Pass	None needed

10	HiddenObject	Highlight does not work outside of the game area	Normal	Click event	The user shouldn't see any highlight effect outside of the dark game area.	Pass	None needed
11	HiddenObject	Highlighter will disappear after 1.5 seconds	Normal	Click event	User can search around the game area and see where they have previously clicked on but this will only last for 1.5 seconds before it disappearing in turn.	Pass	None needed
12	HiddenObject	User proceeds to next page	Normal	Button event	User clicks on proceed to combat button and is successfully taken to the next page.	Pass	None needed
13	Combat (Dungeon boss)	Landing page loads successfully	Normal	NA	The combat landing page displays the user's character name, their class, along with their class animation, roll dice button, current health display and the dungeon monster's image in the background.	Pass	None needed
14	Combat (Dungeon boss)	Roll dice button	Normal	Button event	After rolling the dice, your power level will be randomly generated from a 20-sided die and your class ability will appear.	Pass	None needed
15	Combat (Dungeon boss)	Selecting an ability to attack the dungeon boss	Normal	Button event	Selecting Backstab will update the combat log to remind the user which ability they have chosen then the attack button will appear.	Pass	None needed
16A	Combat (Dungeon boss)	Attacking the boss	Normal	Button event	Upon attacking the boss, the combat log will update with the amount of damage dealt between both parties, the bonus from items and potions used if applicable.	Fail	Combat log is displaying in the correct format but inventory item is not being integrated. Once items in inventory (i.e. being used in the damage calculation)
16B	Combat (Dungeon boss)	Attacking the boss	Normal	Button event	Combat log needs to show +3 bonus for each non-potion item found, in addition, user will heal 5 health for each potion found in the dark caves.	Fail	MongoDB database is running fine and items are being updated to the database but it's not being used when the damage function is applied.
16C	Combat (Dungeon boss)	Combat mechanics for inventory items	Normal	Button event	Combat log needs to show +3 bonus for each non-potion item found, in addition, user will heal 5 health for each potion found in the dark caves.	Pass	None needed
17	Combat (Dungeon boss)	Player health bar	Normal	Button event	Player health bar will flash red when receiving damage.	Pass	None needed
18	Combat (Dungeon boss)	Dungeon boss health bar	Normal	Button event	Dungeon boss health bar will flash red when receiving damage.	Pass	None needed
19	Combat (Dungeon boss)	Player health bar	Normal	Button event	Player health reaches zero and is notified that they have been defeated, a restart button appears.	Pass	None needed

20	Combat (Dungeon boss)	Dungeon boss health bar	Normal	Button event	Dungeon boss health reaches zero and is notified that they have been defeated, a restart button appears.	Pass	None needed
21	Combat (Dungeon boss)	Restart button	Normal	Button event	Clicking on the restart button will take you back to the roll dice button and start the combat phase from scratch.	Pass	None needed





Results

- The initial program was accepting special characters, as a result, the code was reviewed and amended to strictly alphanumeric characters only. Further error handling was done here to ensure that the character name must be between 1-20 characters long and it cannot be empty. (See evolution of the code in test 2A evidence compared to the current code block)
- Within the hiddenObjec'st browser, the items found were to be stored on MongoDB, but there was an integration issue. As it was the first time, I was using MongoDB, there were many errors, both technical and human. For example, when attempting to sync my MongoDB Atlas to VS Code, I had a syntax error of keeping the < > around the DB password. For the server.js file, the app.post and app.get properties had to be carefully scrutinised before I was finally able to run the server connection to the DB. This issue almost caused a delay in the project as a whole as I have spent over 4 days just resolving this integration issue.
- Within the combat mechanic, there was an issue with the integration of the MongoDB inventory items found and retrieved to be used as combat bonuses and benefits. Initially, I forgot to run the server to save the items, but the error still persisted. Upon closer inspection of the code, it turns out to be a variable name error (See test 16B evidence). I used Thunder Client extension on VS Code to understand that there was a get response from my API calls. After multiple trial and error, it seems the program has used the defaulted variable name. This has evolved from the initial proof of concept of a simple turn base attack system to include animation around the health bars, to utilising inventory items to boost your chance at successfully defeating the boss even if you have been unlucky with your initial dice roll.

Summary

Overall, the tests have been successful with a success rate of 85.7% as only 3 tests failed out of a total of 21 tests conducted. In addition, all failed tests have been re-conducted and have now passed.

Analysis of results

During the production of the program code, minor unit tests were conducted in agile settings to ensure that every new feature added was properly integrated. For example, the audio and video animation were tested consistently throughout the process to ensure that they were always functioning as intended. By following good testing standards and monitoring the project on the Kanban board, I was able to complete the project within the assigned deadline, albeit very close to the project's end date.

Rectified

Within my initial program code, the character name input had no error handling. Upon failing the test, I have amended this with a series of selection statements alongside their corresponding

error message so that the user can clearly understand what is accepted for this text box. In addition, I utilise the trim method() which removes whitespaces for the user input. Alongside, I used regular expression (regex) to validate that the character name inputted contains only alphanumeric characters. The regex used here looks like this /^[condition]+\$; where the / determines the beginning and the end of the regex, the ^ indicates the beginning of the string, + is a quantifier and \$ indicates the end of the string.

```
const isValidName = /^[a-zA-Z0-9]+$/.test(characterName);
```

For the combat mechanic, the program didn't recognise the character name that the user has chosen in the creation browser, this was crucial as the character name was used to also store the inventory items on the database. Although the program was defaulting to Player1, I can see that the integration of the various components are working in tandem to produce a full adventure experience for the user. Should I have additional time, I would perform some maintenance on this code block to understand why the program isn't using the dynamic variable input.

Conclusion

This project has passed all test requirements, meets all functional requirements and considers the low percentage of failed tests; therefore I can conclude that this project has been successful and is now ready to be pushed to the production stage and for further beta testing.