

## **Source + Roadmap update**

We have sent the live website to the teachers and obtained feedback. Based on the feedback, we have made certain adjustments to our project and added new features.

For the teachers feedback we have a few things planned:

- Adjust the quick add function to add 25 instead of 1 at a time.
- Added exp to quick add
- A sorting function to quickly sort by XP, gold or name.
- Adjust the inventory system so the admin could see all the cards a user has and delete items from their inventory
- Guild system to group students
- Complete overhaul of website design

Based on the feedback from Bazaar day:

- Make the dashboard more game looking/fun
- Leaderboard adjustments
- Display amount of gold you have on the shop page

Future plans:

- Keep working in the feedback loop, adjusting things based on requests
- Implementation of boss battles (math quizzes)

## Quality Assurance Plan / Results

### QA Plan

Requirements	Activities	Outcomes	Tasks
Register	Test registration form works correctly	Users are able to successfully register	Validate input, make sure no usernames are the same
Login	Test login flow as admin and normal user	Users are able to successfully login	Test valid and invalid login credentials
Edit students	Test ability to add/remove gold, exp, hitpoints, name change, delete student	Admin can successfully edit student details	Test form validation, students new data is saved correctly
Add reward	Test ability to add a new reward/item to the shop that students can purchase	Admin can successfully add a reward item to the shop with description and image	Validate input, correct message appears when image has been added, correctly redirects to the shop page
Remove reward	Test ability to remove an existing reward/item to the shop	Admin can successfully remove reward item from the shop	Validate the reward removal process, check data was saved and updated
Create guild	Test that students can be grouped into guilds	Admin can successfully add new guilds	Validate input, ensure new guild was created and is displaying proper information
Manage guild	Test ability to add/remove gold, exp, hitpoints as guild, check delete guild function	Admin can successfully edit guild details	Test form validation, check students new data is saved correctly when entered into the database as a guild
Legends board	Test the display and sorting of the legends board	Legends board displays and sorts correctly	Validate sorting works correctly and top 10 students are visible
Buy reward	Test ability to buy and item as a student	Users are able to successfully purchase items	Validate item was added to student dashboard and their gold was updated

### **Other functional and non-functional requirements:**

- Correctness
  - The code works correctly and has been verified with testing
- Usability
  - The application is intuitive for users to interact with and has been verified through various user testing activities
- Maintainability
  - Yes it will be easy to support this system over time, the database is not very big and there are a maximum of 65 users at once
- Reusability
  - Some of the styling components of our application can be reused as a template for future projects
- Portability
  - Our application runs on both Windows and macOS with no issues, we are also using AWS for our server and MongoDB for our database, we should be able to deploy it to other cloud platforms or migrate to other databases with minimal changes needed
- Security
  - Permissions are handled through Django's built in authentication system, privacy is handled with data encryption, and passwords of users are hidden

Project demo: <https://youtu.be/ytYIEdhAyFI>

Project commercial: <https://www.youtube.com/watch?v=Ls1eMi0NjD4>

Project day presentation:

[https://www.canva.com/design/DAGVTEFBgAU/CDAYN9L\\_Vf8edauJdjdL5A/edit?utm\\_content=DAGVTEFBgAU&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAGVTEFBgAU/CDAYN9L_Vf8edauJdjdL5A/edit?utm_content=DAGVTEFBgAU&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)