

|  |
| --- |
| **Abertay App Challenge**  Abertay App challenge is a website that allows students to attempt quizzes and coding challenges with the goal of increasing student confidence and engagement with learning.  **EH11 – Byte Club**  **Team Members – Snow White, Julie Whyte, Jack Tully, Joshua Wilkins, Harvey Williams and Michael Wilde**  CMP311: Professional Project Development & Delivery  **[BSc Ethical Hacking, Year 3]**  2024/25 |

**All work in this document is subject to an NDA. This work is confidential and MUST not be shared with others.**

*Note that Information contained in this document is for educational purposes only.*

|  |  |  |
| --- | --- | --- |
| **Team Member** | **Section** | **Technical Contribution** |
| Michael Wilde | Executive Summary | Coded – feedback.php, processFeedback.php, collaborated on Nav bar, style.css, cyberQuiz.php, codingQuiz.php  Database - feedback\_table |
| Snow White | Introduction | Coded – the index.php , gdpr.php , contactus.php page, codingQuiz.php and cyberQuiz.php ,processingScores.php, cyberQuizScipt.js and codingQuizScript.js. As well as helped on userProfile.php  Database – user\_scores |
| Joshua Wilkins | Method | Coded -  Login.php  Logout.php  ProcessLogin.php  Collaborated/Assisted -  Feedback.php  ProcessFeedback.php  Register.php  ProcessRegister.php  UserProfile.php  Links.php  Style.css |
| Jack Tully | Method | Coded -  @challangeManagement @feedbackManagement  @websiteManagement  adminDashboard  adminLinks  register  ProcessRegistration  Collaborated / Assisted -  ProcessLogin (admin functionality)  Database -  admin\_database  user\_database  feedback\_table |
| Harvey Williams | Results | Coded – style.css, links.php, challengeChoosing.php, codingChallenge.php, and cyberSecurityChallenge.php |
| Julie Whyte | Discussion | Coded- userProfile.php |

Executive summary

Traditional methods of learning such as lectures and reading, are often perceived as dry and disengaging. This can negatively impact student engagement and retention, ultimately affecting academic performance. To address this, Xander Purvis, a foundation lecturer at Abertay University, proposed the development of a website to host educational games designed to increase student engagement and confidence in learning materials. This website would also include data collection features to track student progress and gather meaningful insights.

Gamified learning has been proven to be an effective educational approach. A study by TalentLMS on gamification in the workplace reported that 83% of the respondents who received gamified training felt motivated, while 63% of the employees who did not receive gamified training felt bored and unproductive (Apostolopoulos, 2019).

The project aimed to develop an engaging application or website that helps students build confidence and enhance learning through quizzes and coding challenges. Byte Club aimed to develop a website from scratch with the following key features:

* Host educational games
* Allow users to register and log in
* Enable users to view their profile
* Track scores and display a top 10 leaderboard
* Provide feedback functionality
* Display contact details
* Allow admins to monitor student scores and respond to feedback
* Serve as a foundation for future development

Benefits for the client

Our project benefits the client by:

* Providing an engaging platform where students can learn in a fun and interactive way, which can improve retention and student confidence.
* Enabling data tracking, allowing admins to monitor student progress over time and identify frequently played games, aiding in tailored feedback and identifying skill gaps.
* Including a top 10 leaderboard to promote healthy competition and motivation among users.
* Delivering a strong foundation that can be expanded upon. While some desired features were not fully implemented, the project is scalable and can evolve into a university-wide educational resource with further development.

How we delivered

Byte Club hit the ground running, immediately diving into development following detailed planning in the previous semester. The earl preparation included defining project goals, sketching wireframes, planning colours, accessibility and how the website would be linked. The team would take an agile approach to development of the website.

To manage the workload effectively and ensure steady progress was being maintained all around the website, the team strategically divided tasks. Each team members was assigned to specific areas such as User Profile, Feedback, Contact Us, Register, Login, Home, and Admin pages. Once these foundational elements were completed, the team shifted focus to the core of the website – building the interactive cyber security and coding challenges. The team collaborated closely with one another aiding in any necessary troubleshooting and ensuring that the challenges were built in a way to gather user data effectively.

To support the platforms previously mentioned data collection side, the team utilised a service called Lochnagar. This is the university's own database service allowing the team to hold data without paying for a third-party service. Using the university’s Lochnagar service, the team created a dedicated database to securely store user login credentials, admin access details, feedback submissions and responses, and game-related data such as quiz name, date attempted, user email, and score.

The Results

The website features a consistent design and incorporates official Abertay branding, as shown in Figure 1, aligning with Abertay’s branding guidelines. The site is user-friendly, with intuitive navigation and a clean, simple interface that accommodates users of all technical backgrounds.

A screenshot of a website

AI-generated content may be incorrect.

Figure 1: Project website with clean, consistent theme

The challenges successfully meet the brief as placeholder games, establishing core functionality such as:

* Enabling users to play games
* Automatically tracking scores
* Displaying correct and incorrect answers post-quiz
* Storing final scores in a database accessible to admins

The rest of the website includes functionality such as viewing user profiles and previous scores, submitting feedback and receiving admin responses, registering and logging in, accessing contact information, and allowing admin monitoring of student progress.

The team then tested each area of the website through a white box approach, this thoroughly tested for any oversights or errors to ensure the website was in a suitable state to be handed over to the client. The test found that the website functioned as expected and allowed for seamless user experience.

Conclusion

The aims of this project were mostly met, the team built a website from scratch which met the aims of:

* Host educational games
* Allow users to register and log in
* Enable users to view their profile
* Track scores and display a top 10 leaderboard
* Provide feedback functionality
* Display contact details
* Allow admins to monitor student scores and respond to feedback
* Serve as a foundation for future development

The games are designed to track user scores and identify the most frequently played games among students. A top ten leaderboard is featured to encourage a healthy sense of competition between students. Admins have access to students' game scores, enabling them to monitor student progress over time. Additionally, users can submit feedback directly through the website, which admins can review and respond to.

However, some elements of the original project brief were not fully implemented due to technical and time constraints. Firewall restrictions prevented the dynamic uploading of new games and email dependant services like newsletters and password reset functionality could not be implemented without access to a mail server. Features such as difficulty tagging, a badge rewards system and fully functional search bar were not completed due to time constraints, these would be high priority when working on the project further. Furthermore, key security features such as including protection against file and directory traversal still need to be added to ensure a secure user experience.

Despite these limitations, the project was a success. The result is a scalable, well-structured foundation for an educational gaming platform that can increase student confidence, engagement and retention with learning materials. With further development, this site has the potential to grow to a university wide tool for all subject areas to learn and grow.

Table of Contents

[1 Introduction 8](#_Toc1329612793)

[1.1 Background 9](#_Toc291009243)

[1.2 Aim 10](#_Toc130690280)

[2 Method 10](#_Toc514895404)

[2.1 Registering 11](#_Toc802862034)

[2.1.1 Register 11](#_Toc2013512978)

[2.1.2 ProcessRegistration 11](#_Toc692247905)

[2.2 Login & Logout 12](#_Toc1624923966)

[2.2.1 Login 12](#_Toc1437841640)

[2.2.2 processLogin 12](#_Toc1500070727)

[2.2.3 Logout 14](#_Toc1049778579)

[2.3 User Profile 14](#_Toc970864179)

[2.3.1 User Profile 14](#_Toc1990185477)

[2.4 Challenges 15](#_Toc1370029752)

[2.4.1 Challenge choosing 15](#_Toc919783371)

[2.4.2 Challenge page 15](#_Toc1073939585)

[2.4.3 Quiz Page 16](#_Toc682544290)

[2.4.4 Quiz script 16](#_Toc1061840649)

[2.4.5 Process Score 18](#_Toc488109687)

[3 Method Continued 18](#_Toc1665675962)

[3.1.1 CSS –Jack 19](#_Toc1451517334)

[3.1.2 Index –Jack 21](#_Toc841994048)

[3.1.3 Links/navbar -Jack 23](#_Toc186354864)

[3.1.4 Contact us -Jack 23](#_Toc712339435)

[3.1.5 GDPR –Jack 24](#_Toc741316703)

[3.1.6 Feedback/process feedback –Jack 25](#_Toc807469713)

[3.1.7 Admin dashboard –Jack 27](#_Toc720447879)

[3.1.8 Admin links/navbar -Jack 27](#_Toc1881524764)

[3.1.9 Challenge management 28](#_Toc1564024281)

[3.1.10 Feedback management 29](#_Toc505351632)

[4 Results 30](#_Toc616301080)

[4.1 Results - written by Harvey Williams 31](#_Toc1315655188)

[4.1.1 Functionality 31](#_Toc320681214)

[4.1.2 Useability testing 37](#_Toc1171872161)

[5 Discussion 39](#_Toc749166235)

[5.1 General Discussion 40](#_Toc1745390832)

[5.2 Conclusions 43](#_Toc66686024)

[5.3 Future Work or Countermeasures 43](#_Toc1956313087)

[5.4 Call to action 43](#_Toc1000546656)

[6 References 44](#_Toc1754366466)

[Appendices 45](#_Toc1038842033)

[6.1 Appendix A - Deliverables & requirements (Required) 46](#_Toc1913838457)

[6.2 Appendix B - connectionString.php 48](#_Toc1095220738)

# Introduction

## Background

Confidence-building and student motivation are very important factors in modern computing education that directly affects academic achievement. Most students in computing and cybersecurity are self-doubting while working on coding principles and cybersecurity due to low hands-on practice instant feedback and gamification learning environments. Traditional learning methods such as lecturers and written exercises often fail to engage student’s interests , leading to slow skill development and disengagement. To address this issue , websites like Leetcode and gamified challenges like those found in the New York Times Crosswords have become popular as they boost motivation and improve memory retention. Leetcode offers a various coding problems to help users improve their programming skills through practice , much like to the New York Times Crossword which provides daily puzzles that engage users in problem solving activities. Research has confirmed the effectiveness of gamification in educational settings. A meta-analysis (Li, 2023) highlighted that the inclusion of gamification into education might improve student motivation, engagement, interest and learning outcomes. Additionally, students have proven that gamification can enhance students’ engagement and motivation , making the learning more dynamic and engaging (Cuervo-Cely, 2022). These platforms tend to lack tailored content specific to certain curriculum and might not cover the needs of all student groups.

To bridge these gaps, the project aims to develop an app or website called Abertay Challenges which is specifically for Abertay students, where students can engage with quizzes and coding challenges in a structured and enjoyable manner. The goal is to offer a learning support system where students can challenge their abilities, track progress and gain confidence. By incorporating interactive problem-solving elements that are inspired by New York Tims Crosswords and Leetcode , the platform will encourage students to practice coding and cybersecurity principles in a fun way. The platform will also provide valuable data to assess how these interactive tools impact the student’s improvement over time.

The client for this project is Xander Purvis , a foundation computing lecturer at Abertay with expertise in programming, user experience , applied AI and game design. With a strong focus on student engagement , Xander is keen to explore the potential of an interactive platform that will enable students to attempt quizzes and coding challenges with the goal of increasing their confidence and enthusiasm for learning. The application should include at least 2 main types of challenges :

* Coding Challenges – Students are presented with a programming problem and can input their solution. The application evaluates their answer and provides immediate feedback.
* Cyber Security Scenarios/Questions - Students engage with a cybersecurity-related question or scenario and submit their responses. The application will check their input and provide feedback. Points and badges for correct answers.

To enhance engagement, gamification aspects will be brought on the platform such as:

* A dynamic challenge system where 2 new challenges are revealed daily.
* A stretch goal – A leaderboard displaying the top ten users each week.

The app/website needs to be designed with longevity in mind, i.e., challenges must be stored and managed well. Administrators will have the ability to create and upload new challenges at any time, ensuring a continuous stream of fresh content for students. Also, the site will include a data collection component to track student interactions and their progress over time. Students will register with their student number , email and password. Admins can tag each challenge based on their difficulty , allowing for a structured analysis of student performance and their skill development. There may be other relevant metrics captured to measure effectiveness of the platform in improving student learning outcomes.

High dropout rates in Computer Science programs are a significant concern, particularly with UK universities. It has been noted that computer science courses have the highest dropout rates among all subjects with 9.8% of students writing off their studies before completion. Some of the reasons that have contributed to these dropout rates include financial challenges, inadequate academic support, and unrealistic expectations regarding the course content (Tech, n.d.). By providing an interactive and supportive learning environment, the Abertay Challenges TBC app aims to address these issues, perhaps reducing levels of dropout and increasing student retention.

The Abertay Challenges application aims to bridge the gap between traditional teaching methods and the need for interactive, engaging learning experiences in computing education. By giving students an interactive and gamified learning environment, students will have more enjoyable and engaging experience as they work through coding and cybersecurity exercises. By incorporating gamification elements and providing immediate feedback, this application aims to enhance student engagement, boost confidence, and ultimately improve learning outcomes in coding and cybersecurity subjects. Furthermore, tracking students’ progress and challenge difficulty will provide insightful feedback into how students improve over time , allowing educators to modify their teaching strategies.

Ultimately, this application has the potential to become a valuable tool for students looking to build confidence in their technical abilities while making learning more accessible and enjoyable.

## Aim

The aims of the Abertay Challenges project are to create an application/website that allows students to attempt quizzes or coding challenges with the goal of increasing student confidence and engagement with learning. This aim has several sub-aims which are:

* Educational Engagement – To create an interactive application/website with fun and educational cybersecurity and coding challenges to help users improve their skills.
* To create two challenge types - Include at least two types of challenges:

A cybersecurity scenario or question requiring the user to input the answer

A coding challenge where users input their solution

* User Feedback and Rewards – Informing the users if their answers are correct and showing how many students got it right. Rewarding correct answers with points and possibly badges.
* Leaderboard and Competition – Optionally implement a weekly leaderboard displaying the top ten users.
* Daily Challenges and Content Management – Ensure longevity by storing past challenges and revealing to new challenges daily, allow admins to add new challenges anytime.
* User Registration and Data collection – Enable users to be able to register with a username, student number and password.
* Skill Tracking and Analytics – Collect and track user performance data, including difficulty levels of challenges, to assess skill improvement over time.

# Method

## Registering

### Register

For the website to function, users had to be capable of creating accounts to store their details, information and test scores. To accommodate for that, the team created a “register.php” page which provided an HTML form for new users to create an account by entering their student number, email, and password. The POST method is then used to submit the details from the form directing them to processRegistration.php to be processed.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 21: registering form for users inputting their account details

A computer screen with colorful text

AI-generated content may be incorrect.Before the users details are processed however, a function is first run using JavaScript to verify the users attempted password. This ensures that the password complies with the sites password policies and that they have entered the same password twice, confirming that there are no user typos.

Figure 22: Script for verifying a secure chosen password

### ProcessRegistration

In processRegistration.php, the user input that is captured from using the POST method in register.php is first sanitised before use. The password entered is then hashed using PHP’s built-in password\_hash() function with the PASSWORD\_DEFAULT algorithm to ensure security.

Afterwards, the password, student number and email is posted to the “user\_database” database stored on Lochnagar, with the connection being made through the connectionString.php page (See Appendix B).

The users account details are then put through prepared statements to avoid SQL injection before the INSERT statement is used to upload the information.

A screen shot of a computer code

AI-generated content may be incorrect.

Figure 23: Function for inserting new users details into the user database.

## Login & Logout

### Login

Similarly to the registering page, login.php uses an HTML form with the user inputted details being submitted through the POST method to processLogin.php, where each label for user input is set to “required” as to make sure the user submits all details needed to login.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 24: HTML form for submitting login details

### processLogin

For processLogin.php, the server receives the data submitted through the POST method, storing them as local variables. Once the users input has been saved, the details stored in the user database is searched through a prepared SELECT SQL statement, looking for a match to the users email which acts as a primary key.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 25: Submitted login details are put through a prepared statement to search through the user database

If there is a match between the email entered and an email linked to an account in the database, the users password is compared to the hashed password saved in the database using password\_verify(). Upon successful verification, the session variable “loggedIn” is set to true, whilst the session variable for “email” is created to store the users entered email. The user is then headed to userProfile.php.

A computer screen shot of a computer

AI-generated content may be incorrect.

Figure 26: The submitted password is compared against the hashed password in the user database. Logging the user in if verified

If there isn’t a match between the entered details and the user database, the php page then searches for a match in the admin database. If there is a match to an admins login, the site follows a similar process to a user login, except the session variable “loggedIn” is replaced with “isAdmin”, allowing for access to the sites management.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 27: Users login details are searched for in the admin database

If there is a match between the inputted details and the admin database, but the stored database password isn’t hashed, the database is corrected using the UPDATE SQL statement.

A computer screen with text on it

AI-generated content may be incorrect.

Figure 28: If the admins password is stored in plaintext, it’s updated with a hashed password

Finally, if the entered user details don’t match any accounts in any database, the user is prompted with an alert that their details may be incorrect.

A screen shot of a computer screen

AI-generated content may be incorrect.

Figure 29: An error message prompts the user that their login details were incorrect

### Logout

When the user chooses to logout, the logout.php page, all the saved information from the sites session is collected and then deleted, before the user is headed back to “index.php”, logged out.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 30: Session variables are deleted to log user out

## User Profile

### User Profile

Using the SQL SELECT statements, the user profile page collects data from the feedback, user and quiz score databases. This displays to the user in HTML tables their student number and email, their totalled scored points across all tests, their feedback on the site and any response they got, their history of what quizzes they took, when they took them and what score they got, and finally the leaderboard of the top 10 students and their high scores on each quiz.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 31: Total user score across all quizzes are calculated

## Challenges

### Challenge choosing

challengeChoosing.php acts as a navigational page where users can select between the currently available game types. Here, two buttons are used as links to redirect users to either of the challenge explanation pages.

A computer screen shot of text

AI-generated content may be incorrect.

Figure 32: Button used to link to the coding challenge page

### Challenge page

From here on out, the code and files for the coding and cyber quizzes are almost identical. With the only difference in the final project being the links and content of the questions.  
  
Before being directed to each quiz page, first the user is headed to an explanation page, allowing the user to understand how the quiz works by listing it’s components. Once they are ready a button directs them to the respective quiz page.

A computer screen shot of a program code

AI-generated content may be incorrect.

Figure 33: Content of the coding challenge page

### Quiz Page

For each quiz php page, a set of dividers is used to host the interactive quiz which is hosted using JavaScript. The site loads the JavaScript logic from the JavaScript folder and then the DOM containers and their ids are used for the quizzes questions, results and controls to be injected.

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 34: Content of the coding quiz page

### Quiz script

AS previously mentioned, for both quizzes the content was stored completely within JavaScript files that were imported into the DOM. This was done by defining an array of question objects, each containing the question, a list of answer options, and the correct answer. When the quiz starts, the displayQuestion() function renders each question on the php page and uses the shuffleArray() function to randomize the option order each time.

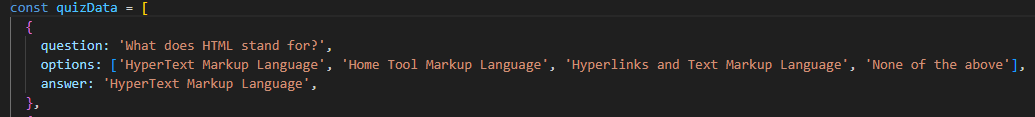


Figure 35: Example question from the coding challenge array of questions

A group of colorful text on a black background

AI-generated content may be incorrect.

Figure 36: Function to shuffle the order of questions

A computer screen shot of a program code

AI-generated content may be incorrect.

Figure 37: Loop that creates the options for each question

The checkAnswer() function is then triggered once the user submits their answers. This compares each selected answer to the actual correct answer and updates the users score accordingly. Incorrect answers are then stored for feedback using an array of objects.

A computer screen shot of code

AI-generated content may be incorrect.

Figure 38: checkAnswer() is used to either increase score or add the chosen answer to the list of incorrect questions

When the quiz ends, the displayResult() function displays the users final score and submits the result to processingScoresCoding.php via an AJAX request. The user’s email is then retrieved from the sites session to be used as the primary key in the scores database.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 39: displayResult() displays user’s score and posts the final score through an AJAX request

Finally, the showAnswer() function displays all of the users incorrectly answered questions, reinforcing learning by showing the user's incorrect answer followed by the correct one. Event listeners are used for the submit, retry, and review buttons allowing control over quiz flow, and the automatically loading the quiz on page access by initially calling the displayQuestion() function.

A computer screen shot of code

AI-generated content may be incorrect.

Figure 40: showAnswer() displays the list of questions the user got incorrect, alongside what they chose and the correct answer

A screen shot of a computer code

AI-generated content may be incorrect.

Figure 41: Event Listeners are used to check when the user clicks on one of the buttons. displayQuestion() is used to initially display the questions to the user

### Process Score

After receiving quiz scores through AJAX, processingScoresCoding.php inserts them into the scores database with a timestamp. Once again using email as the primary key in the table, and prepared statements to avoid SQL injection.

A screenshot of a computer program

AI-generated content may be incorrect.

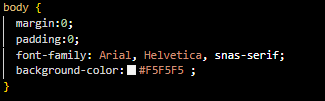
Figure 42: processingScoresCoding.php inserts the users score into the user\_scores database through a prepared statment

# Method Continued

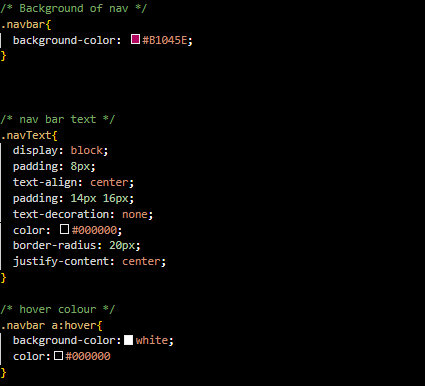
### CSS –Jack

A comprehensive and responsive CSS styling framework was developed to establish a consistent user interface across the web application that aligned with the brand image established by Abertay.

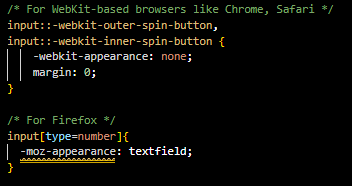
The body element was globally styled to reset default margin and padding, apply a sans-serif font family, and set a neutral background.



The navigation bar (.navbar) was styled with the Abertay colours and interactive hover effects on links (.navText), ensuring usability.

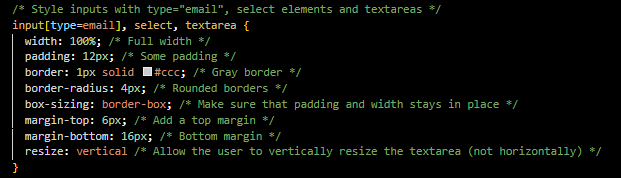


Number input fields were normalized across WebKit and Firefox by disabling spinner controls.



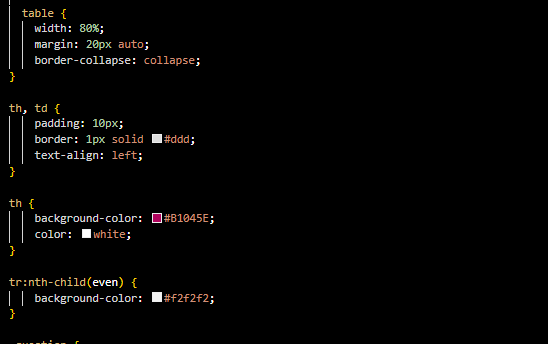
Forms were enhanced with uniform styles for input, textarea, and select elements, including padding, border-radius, and full-width responsiveness.

(This was replicated for all input fields such as text/passwords)

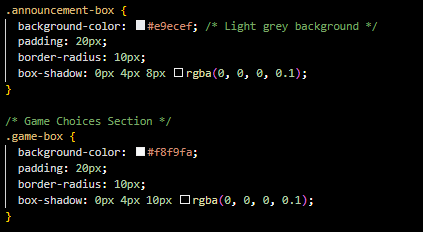


The submit button was given a hover effect to indicate activity.

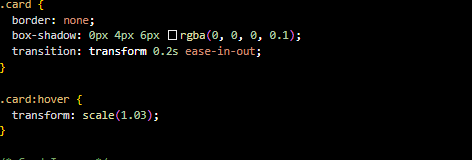


Tables were styled for readability using alternate row shading and defined column widths. 

Component-specific containers such as .announcement-box and .game-box applied shadow and padding for emphasis.



Cards were given a hover effect to indicate activity



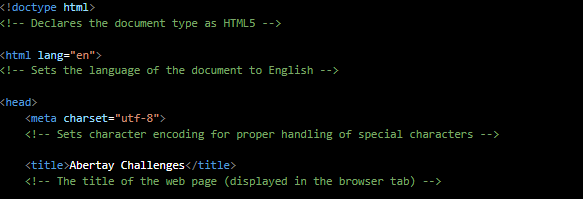
Flexbox was used for central alignment in .center-container to horizontally center UI components



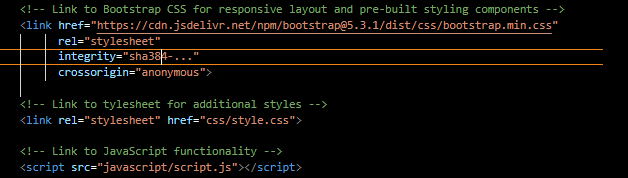
### Index –Jack

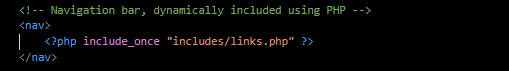
The index page establishes the site's structure using HTML5. It links Bootstrap for responsive design, a custom stylesheet, and JavaScript functionality. Content includes a PHP-included navbar, descriptive text about Abertay, and a centred image, ending with a branded footer.

HTLM5 structure and metadata:

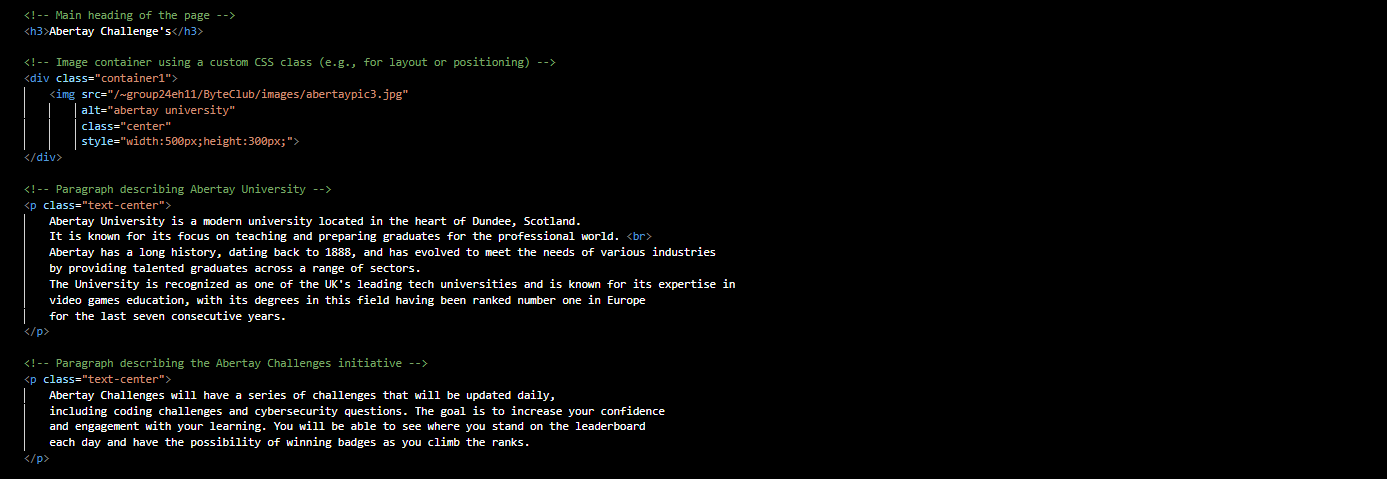


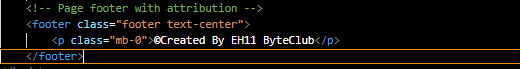
External recourse linking:

PHP nav bar:



Main content:

Footer include:

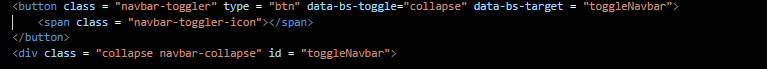


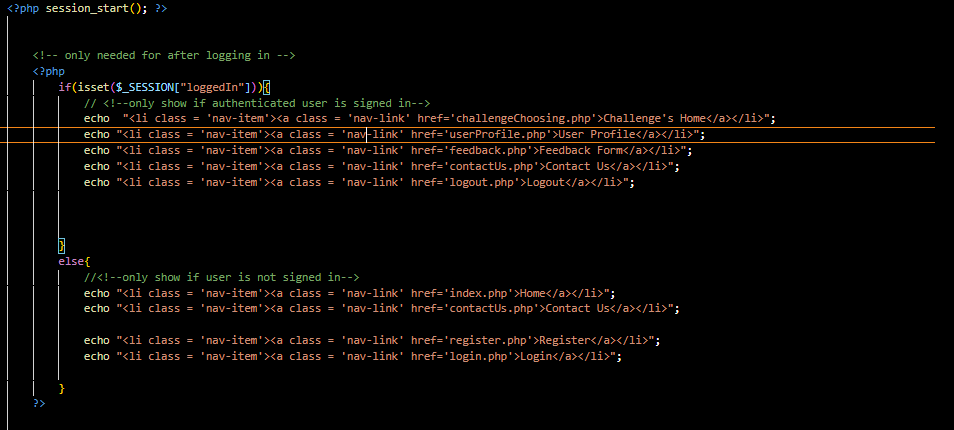
### Links/navbar -Jack

A responsive Bootstrap navigation bar was implemented with conditional PHP logic to determine user authentication. Navigation options dynamically change based on this. . A collapsible layout supports mobile responsiveness, and a search bar is right-aligned for quick access.

The <nav> is fluid in nature allowing for responsive readjustment on smaller devices:

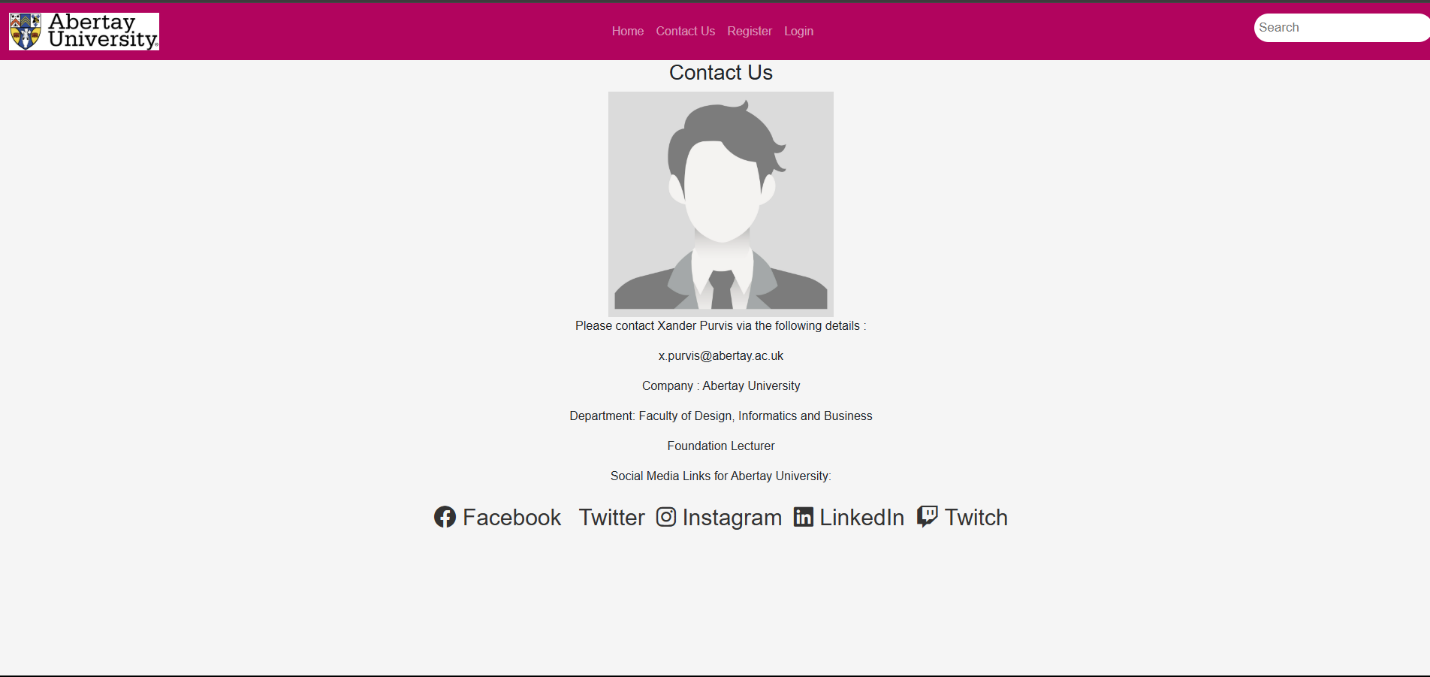
Collapsible buttons for mobile display:

Session controlled link availability:



### Contact us -Jack

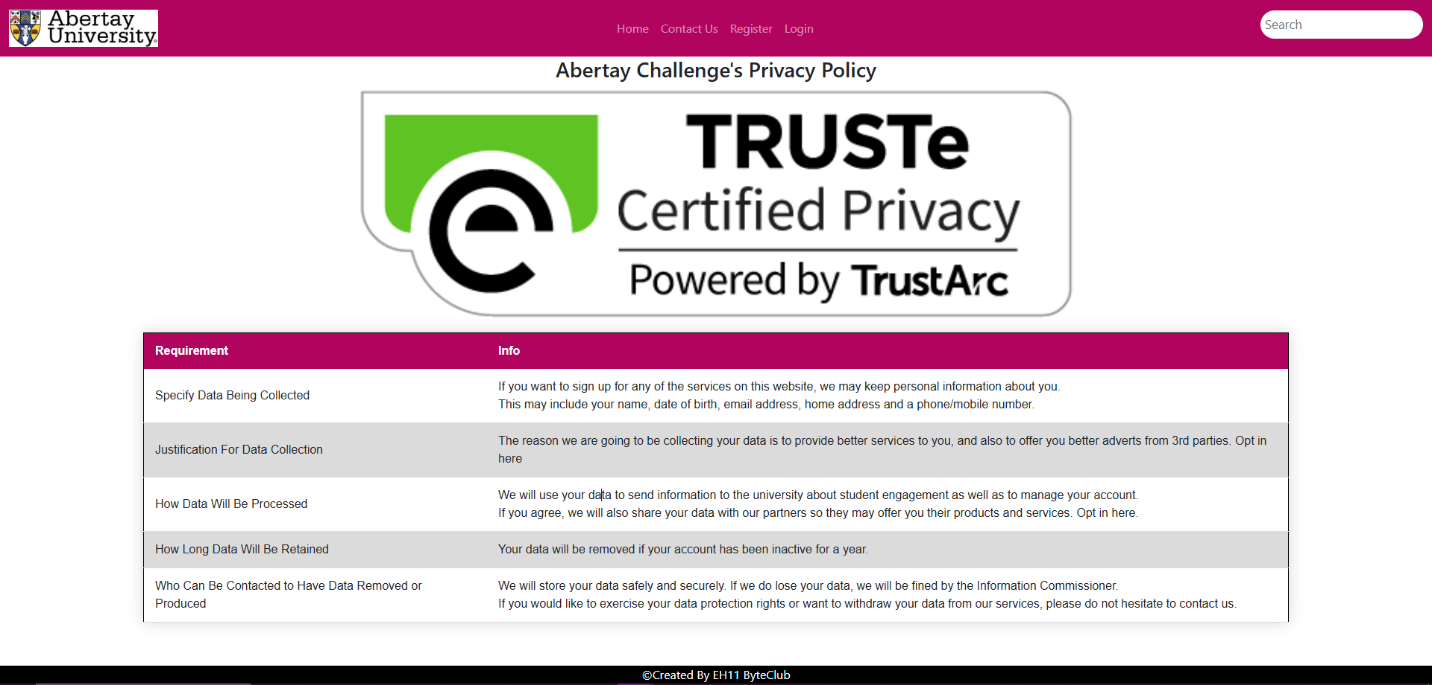
The contact page contains the same formatting as the previous pages and includes a placeholder image of the client as well as placeholder information (the general Abertay university contact) in place of the clients.

Inclusion of social media hyperlinks:



### GDPR –Jack

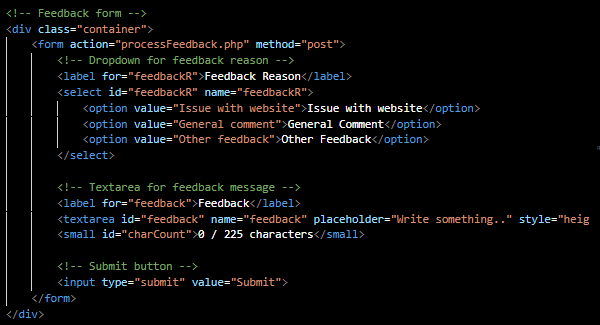
The GDPR page follows the same format as the previous pages and details what data is collected/handled by the website and how its handled in accordance with the law.



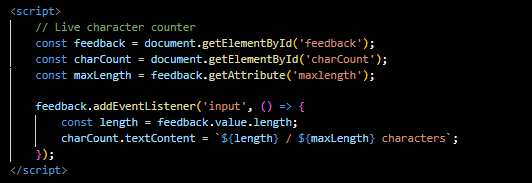
### Feedback/process feedback –Jack

A client-server feedback module enables users to submit comments categorized by type. The front-end in feedback.php uses a Bootstrap-styled form with real-time character count enforcement and conditional alert banners for feedback validation.

Form interface:

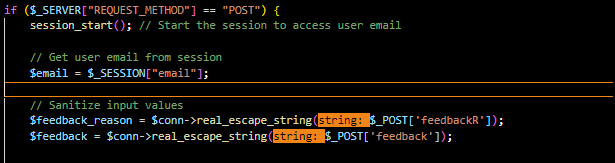


Character counter:

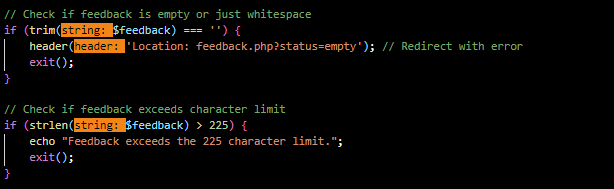


Dynamic alert for feedback response validation:

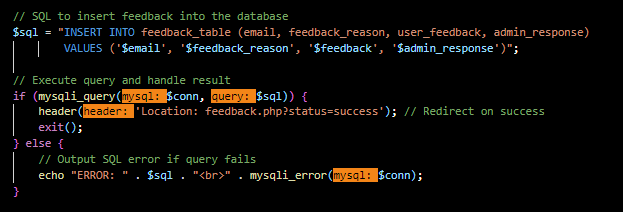
 Input sanitisation within processfeedback.php:



Logic to make sure the feedback submitted is valid:

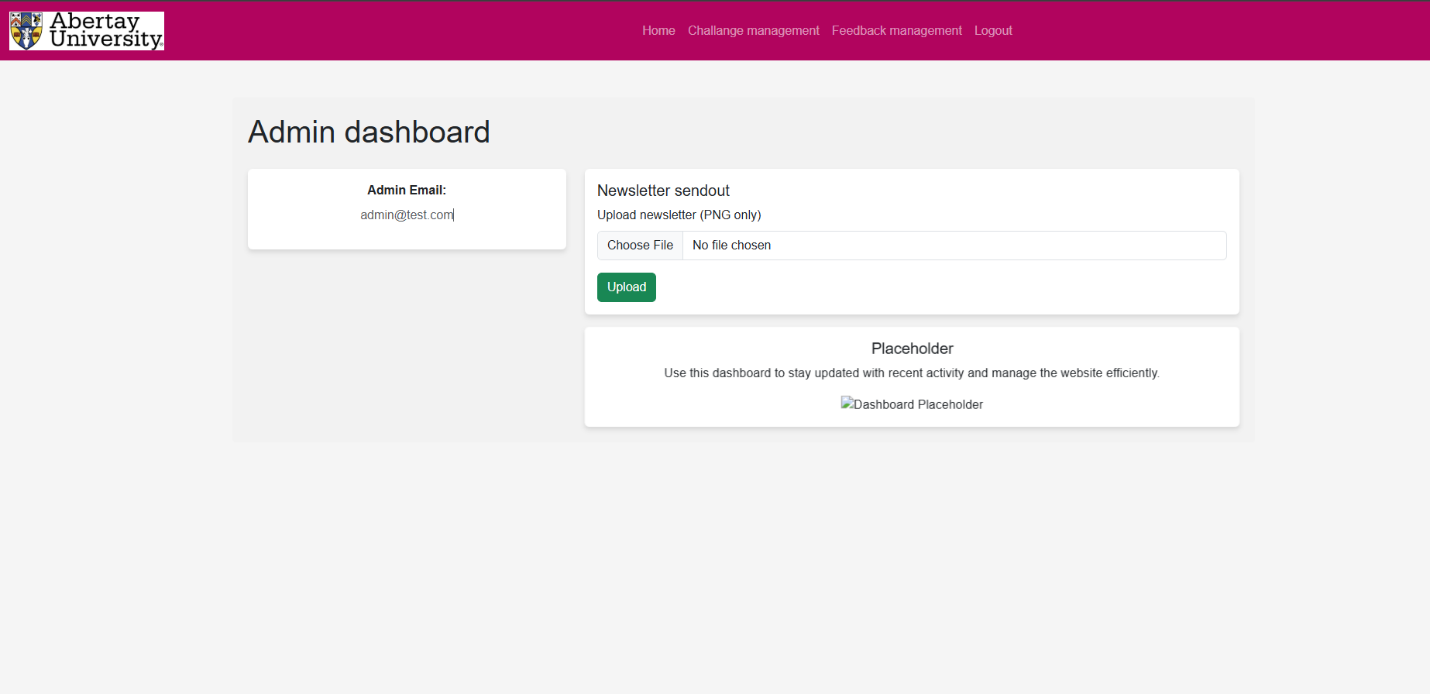


SQL query to safely update the database:

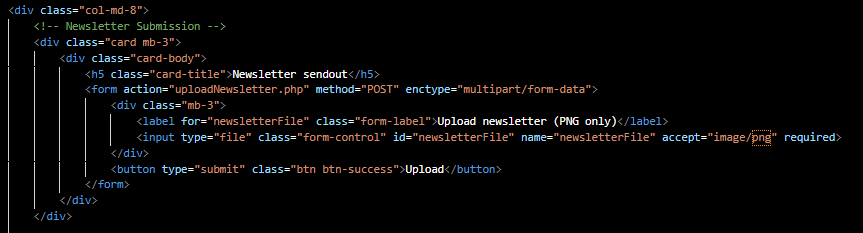


### Admin dashboard –Jack

The admin dashboard is a clean and easy to navigate webpage that allows an administrator to send a newsletter out to the registered users of the website aswell as navigate to the various links present from the custom admin navbar that will be detailed next.



The code related to uploading files restricts the filetype to exclusivly .png meaning even if an admin account was compromised its features could not be used to distribute malicious code in the case of an account breach.



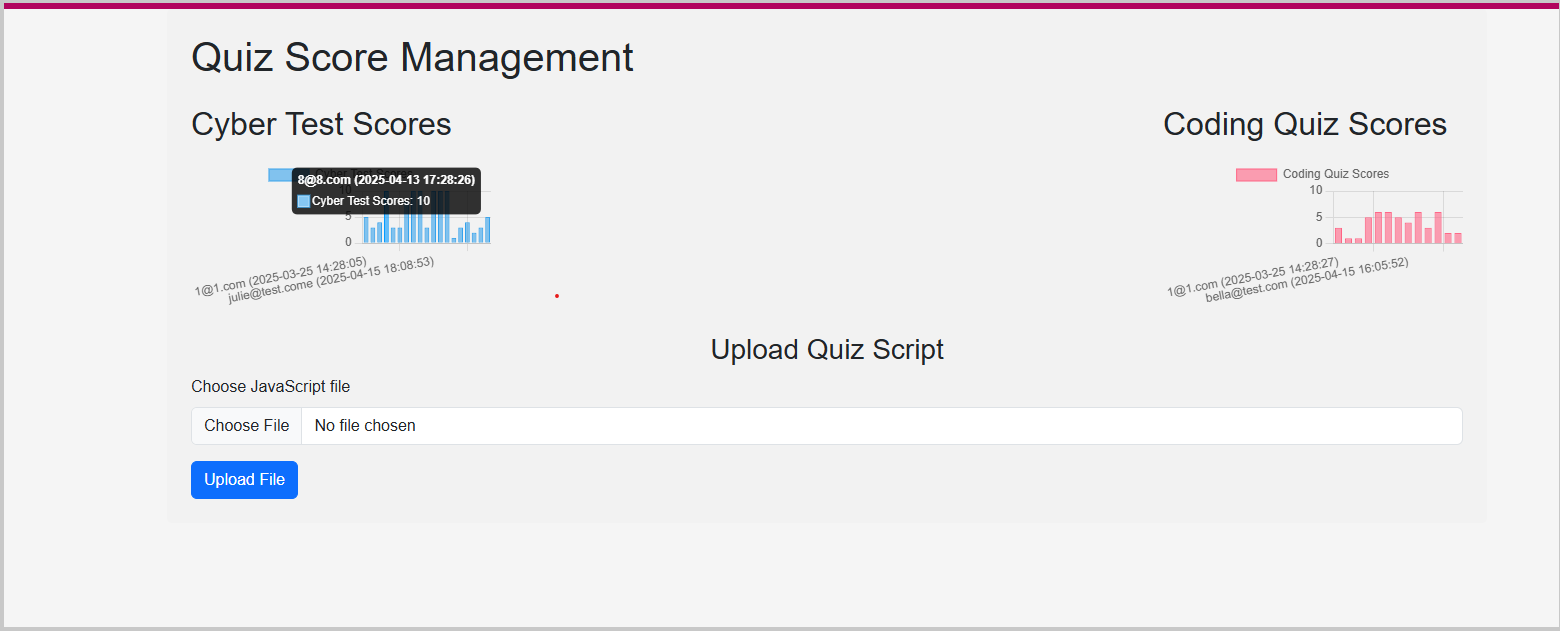
### Admin links/navbar -Jack

Upon admin logon the user is presented with a unique navbar that allows the user to navigate to the higher security pages being the challange and feedback management. The navbar still follows the same structure as the standard with all the same features.The admin navbar is only called from within admin pages which require a unique sessiion id to accsess meaning these pages can‘t be reached by a standard user.

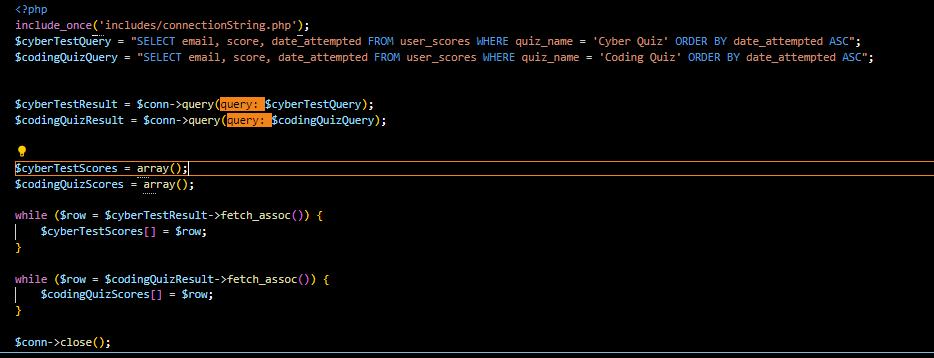


### Challenge management

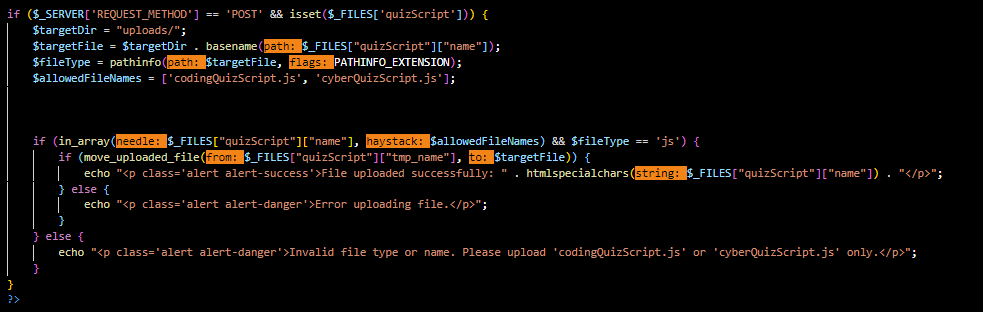
The challange management page allows an admin to see a graph for each type of quiz containing the username, time and score for each attempt taken allowing a students progress to be tracked. This page also contains a sanitised file upload for changing the quizzes present on the website in a way that could not be manipulated by an attacker. This is done in the same fashion as the newsletter upload.



The below code uses the graph containers set up in the css to display on the website translating the sql table containing the scores into an easy to read format for the admin accounts.

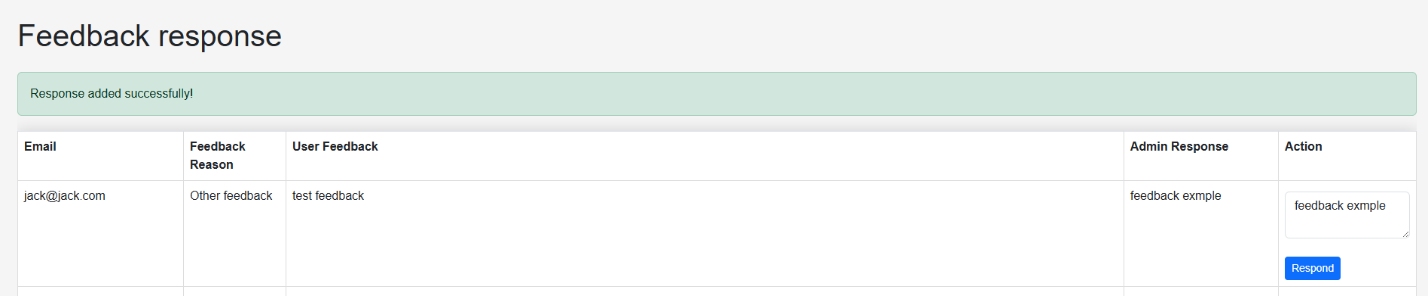


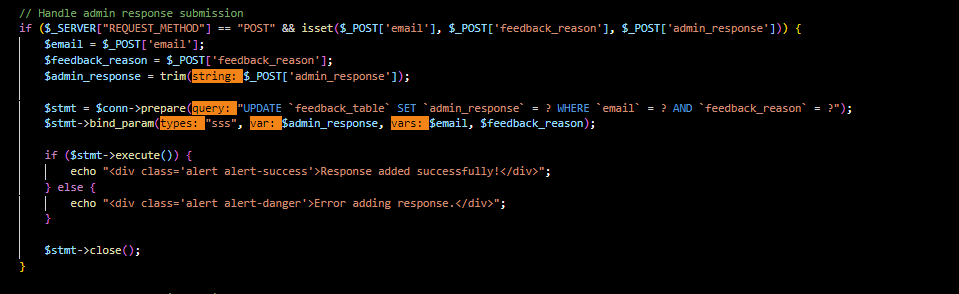
The file upload sanitisation only permitting the specific filetype/ naming convention used through the website to ensure that an error such a misnamed file cant cause the website to malfuction or for any other scripts to be executed.



### Feedback management

The feedback management page activley displays all feedback submitted in order of most recent, alongside this is displays the admins response to eliminate the potential of two admins responding to the same feedback, all admin accounts share the same access to the feedback page.

This response updated the sql table for feedback allowing it to dynamicly update the users profile page where they can view reponses to their feedback submissions.

When the feedback is displayed on the admin page it is encoded in a way so that even if script is managed to be uploaded via feedback response from the user accounts (which shoudnt be possible) it is porevent from running on admin side where the most potential harm could be caused.



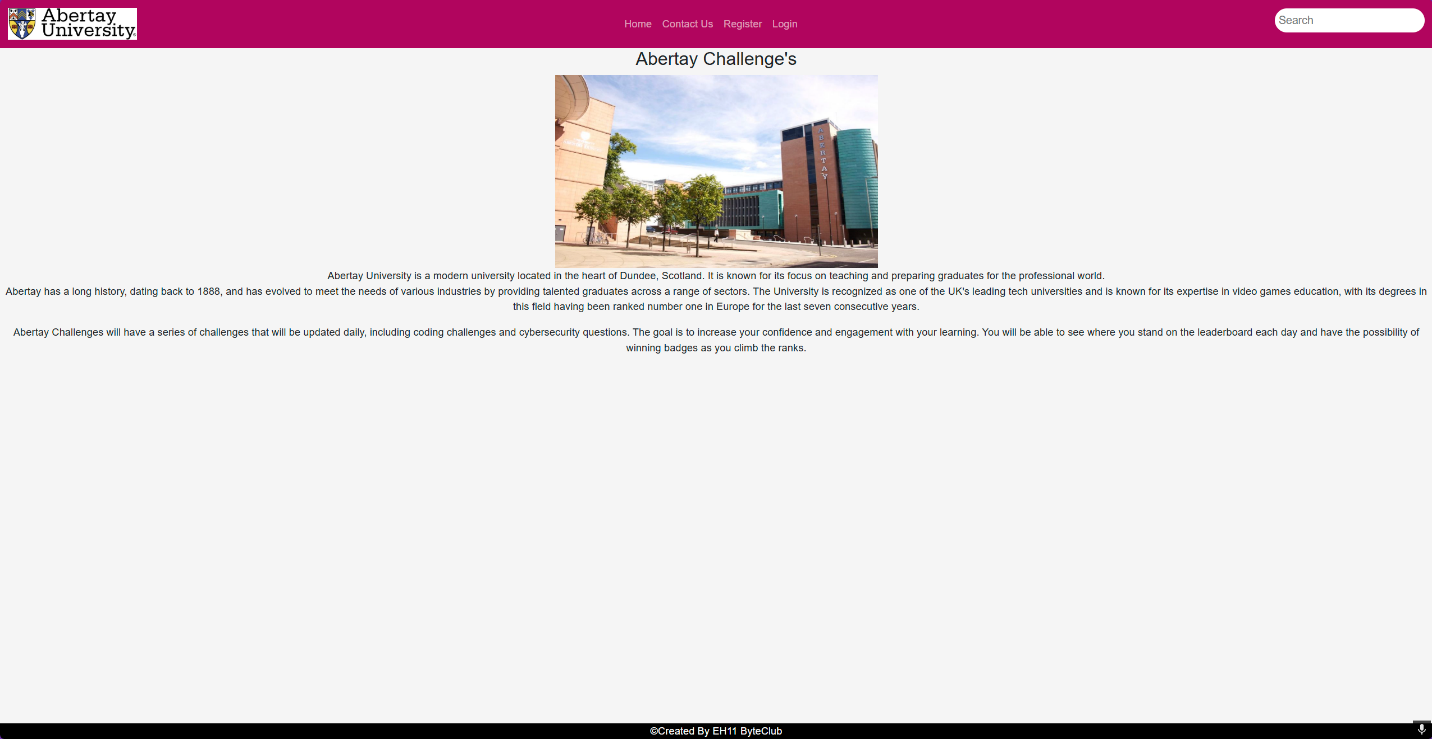
# Results

## Results - written by Harvey Williams

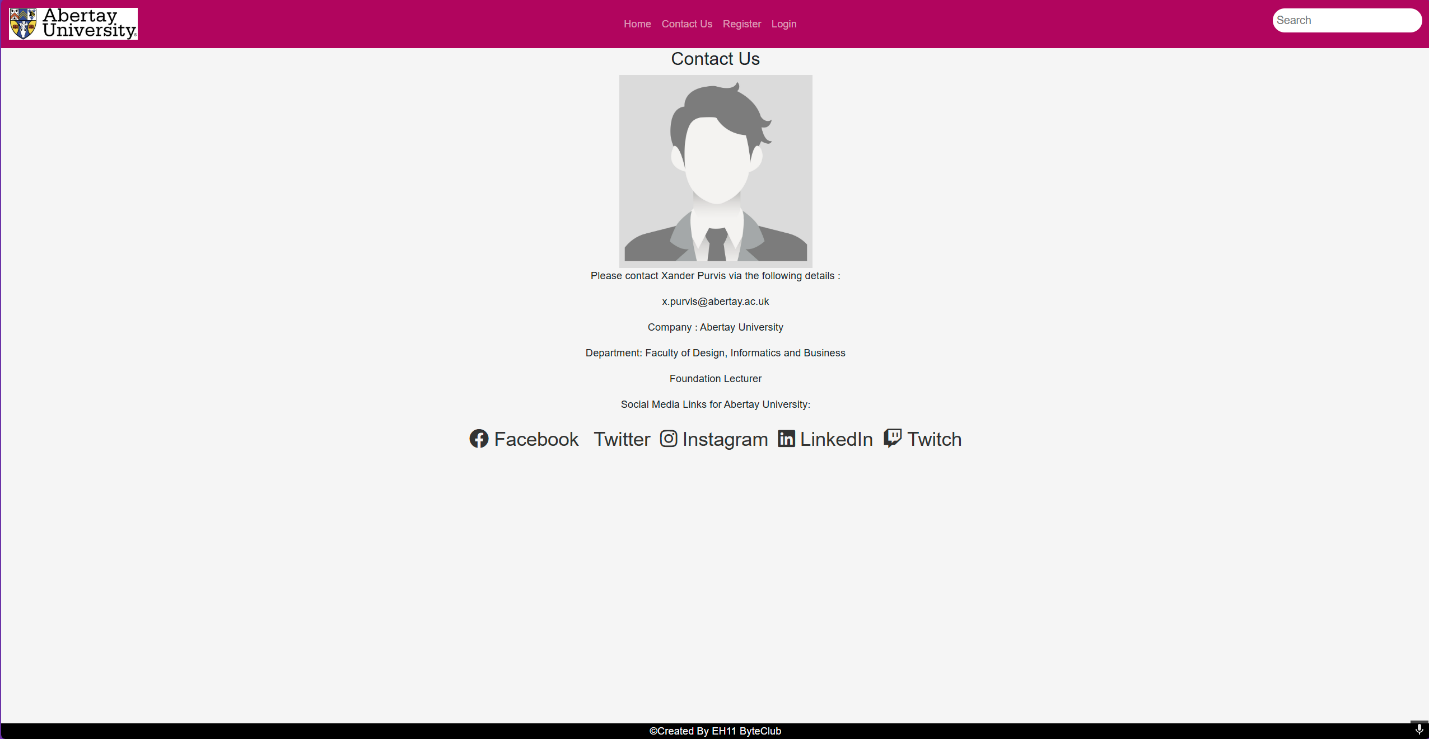
To evaluate the functionality, performance, and usability of our website prototype, a range of tests and data collection methods were conducted. This included functionality testing across all key pages, performance checks (such as responsiveness of the pages), a basic security analysis, and early usability feedback. Our goal was to identify potential issues, assess how well the website performs under typical usage scenarios, and ensure a positive user experience. Below, are the findings, supported by screenshots, gathered during the testing phase.

### Functionality

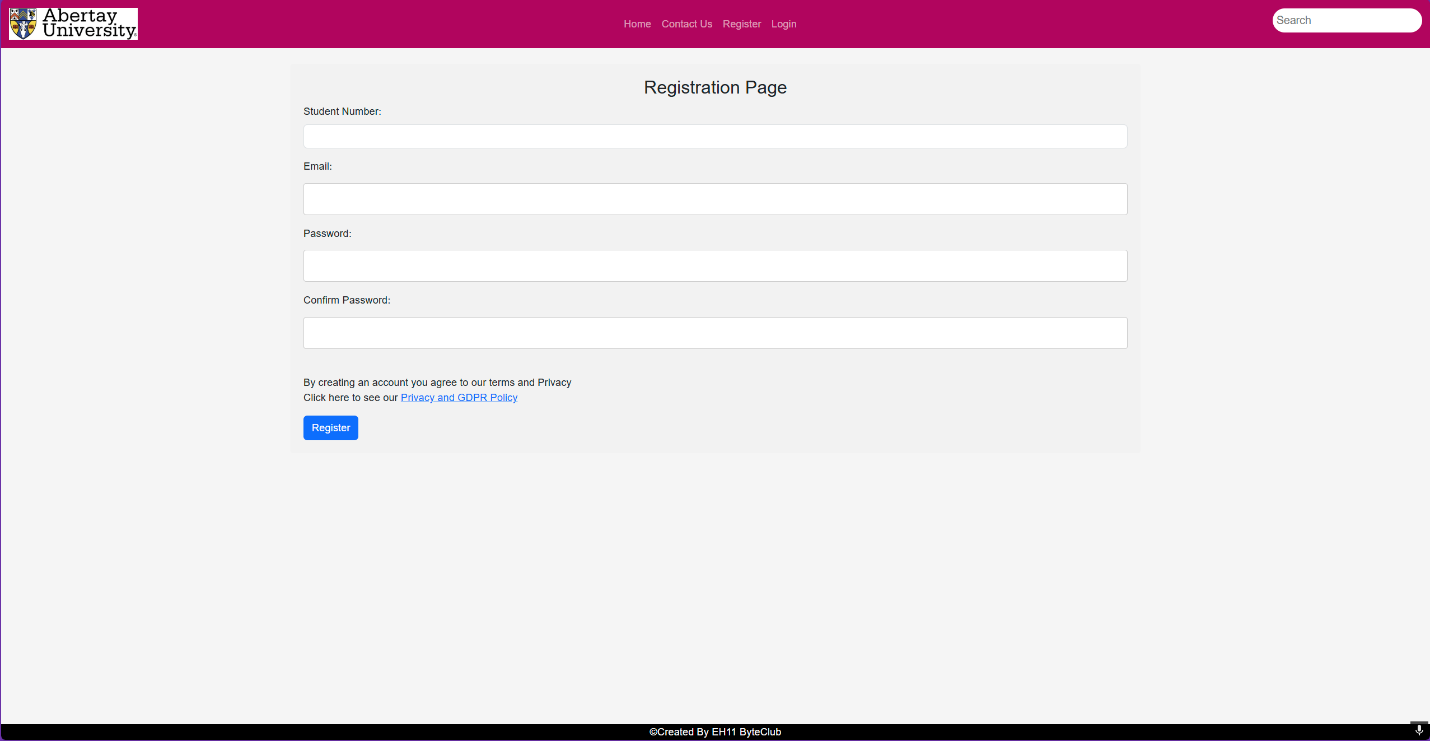
As the website the group designed was a prototype, the website contains only the main pages needed to provide a solid base for further development to take place, these pages include features like including user authentication, quizzes, and profile management. Each page was tested to ensure that it performed as intended and met the goals set by the project.

*Figure 1 – Main page*

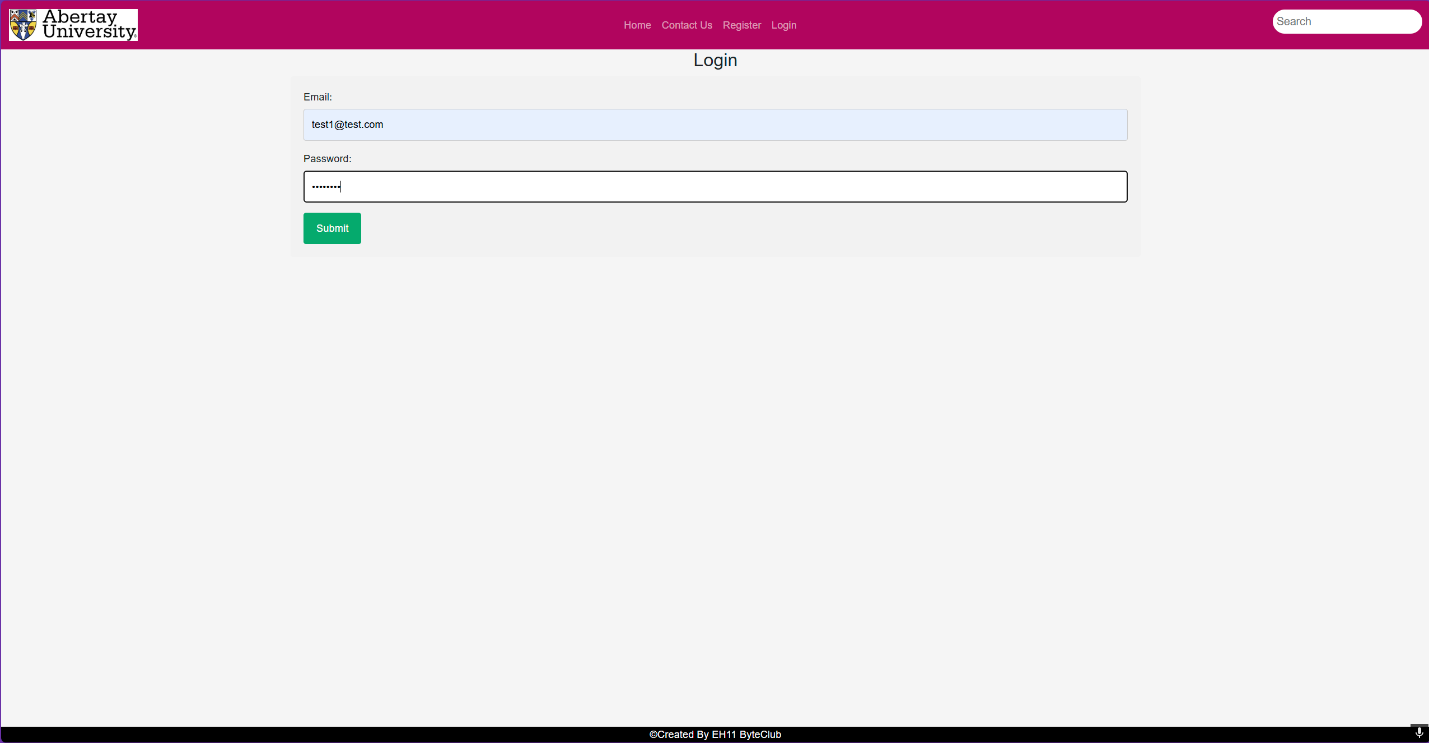
The first page that the user is met with after visiting the website is the home page (seen in figure 1). This page allows the user to navigate to other aspects of the website using the navigation bar (Nav bar). As the user won't be logged in at this point there will only be four pages accessible to them, which are the home page, contact page, registration page, and the login page.

*Figure 2 – Contact page*

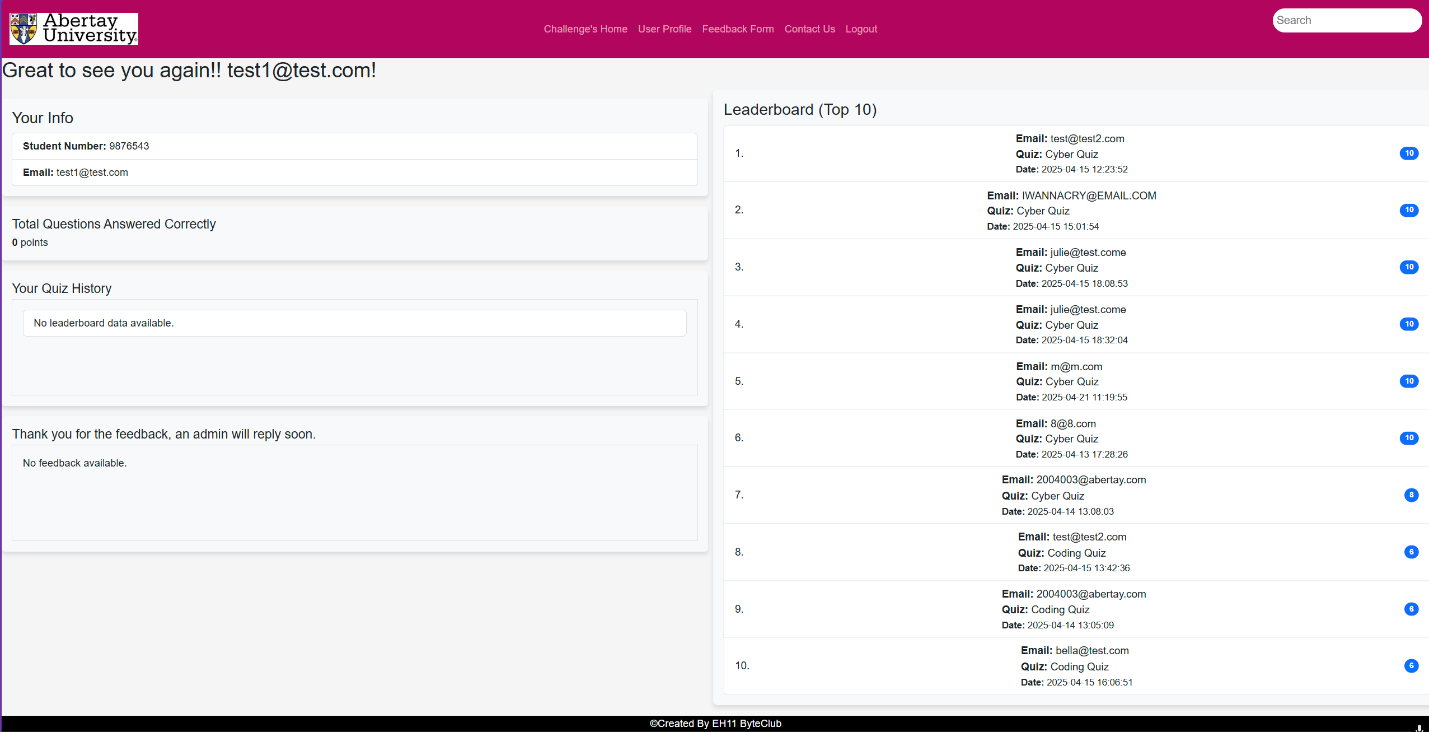
If the user navigates to the ‘contact us’ page (figure 2) they will be greeted with our client's contact information as well as all the university’s social media links which are fully functional and takes the user to the corresponding media platform.

*Figure 3 – Registration page*

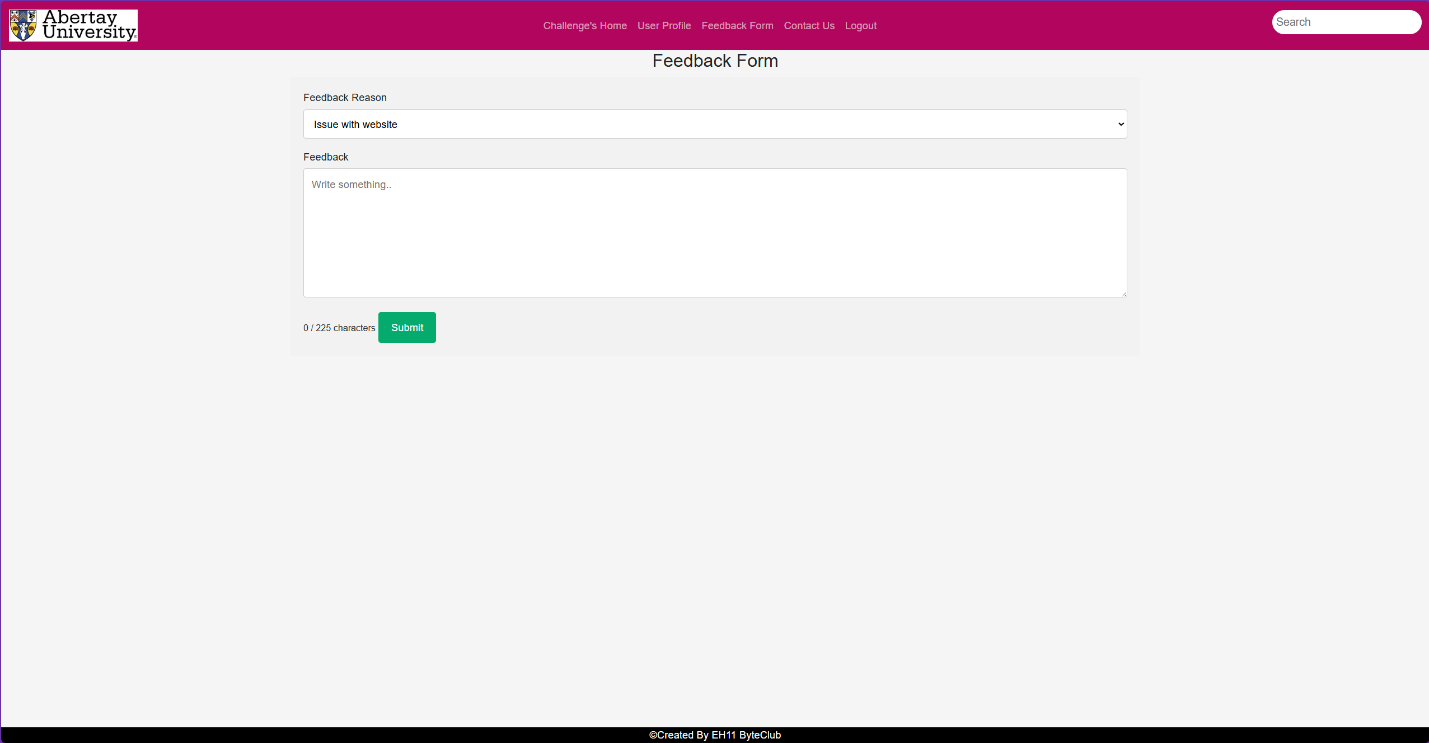
The registration page is where new users will have to input their student number, email, and a password of their choice into the text fields to register on the site, which will allow them access to the hidden pages. The website will then store this information in the database and the passwords are encrypted, meaning they are secure. On this page there is also the GDPR page link (seen in figure 3) where the user can read the legal legislation, so they know how their data is being used.

*Figure 4 – Login page*

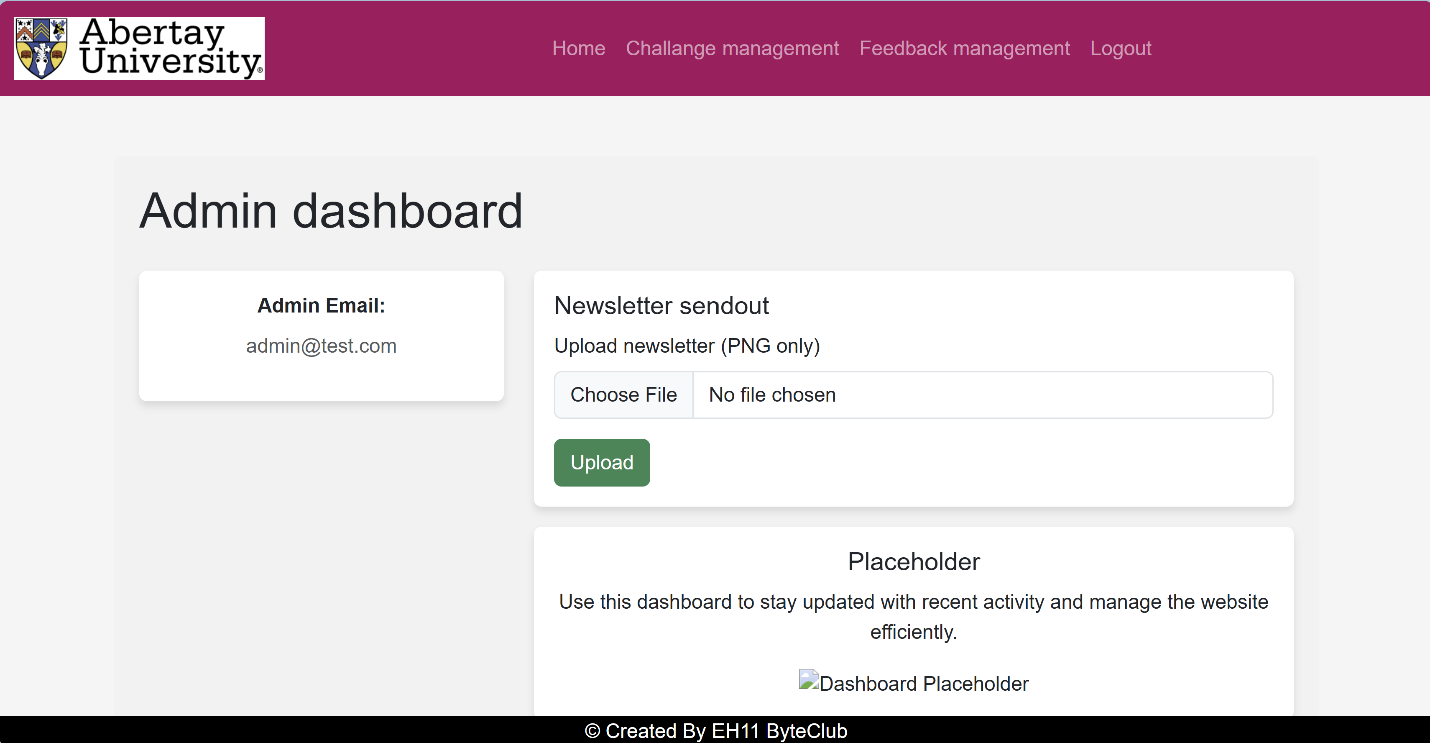
The final page the user has access to if not logged in is the login page, which the user will also be redirected to after they have successfully registered on the previous page. In figure 4 there is a test account’s information being shown.

*Figure 5 – User profile page*

After successfully logging in, the user is taken to their user profile page (figure 5) where they can see stats about themselves and the top 10 users who have gotten the highest scores on the quizzes. There’s also the feedback box that will allow for the administrator to respond to any feedback the user has provided using the feedback page.

*Figure 6 – Feedback page*

Shown in figure 6 the user feedback page is a page where the user can provide feedback to the administrator while selecting the option they wish to provide feedback on. This is decided by the user using the dropdown menu above the text box where they can then submit a message that the administrator can then see on the admin page.

*Figure 7 – The admin dashboard*

When first accessing the admin page (seen in figure 7) the user will be greeted with a few options with them being newsletter sendout, challenge management, and feedback management. These options will allow an admin to change aspects of the websites as well as responding to feedback (as mentioned in figure 6).

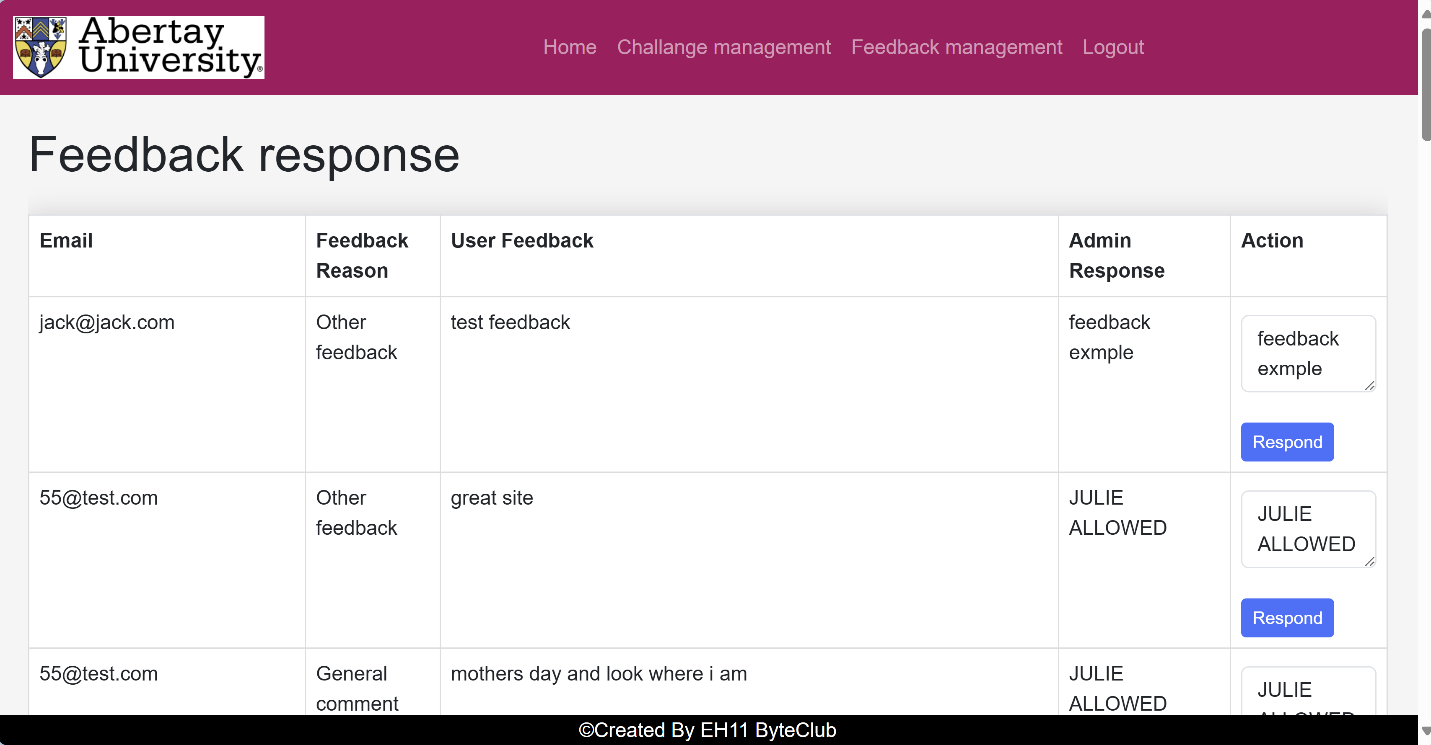
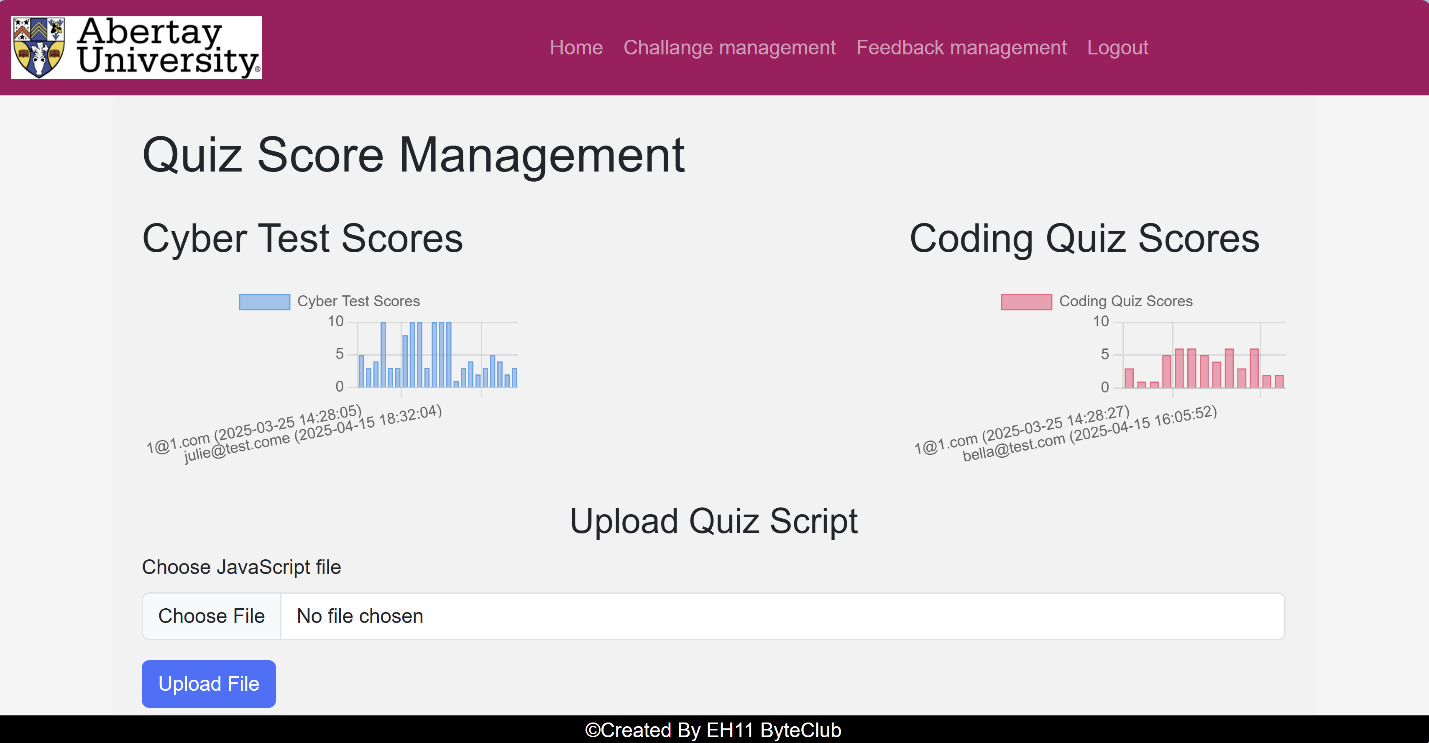
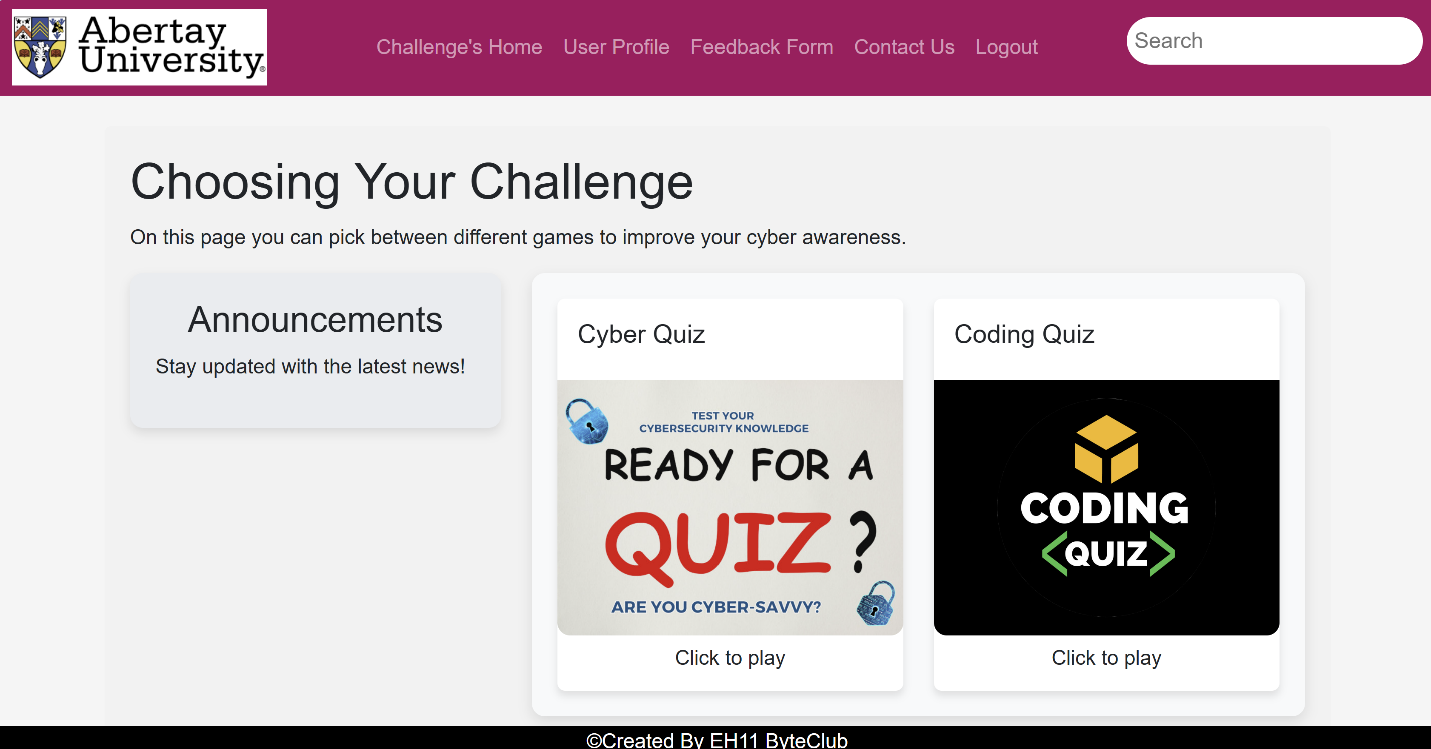
*Figure 8 – Feedback response page*

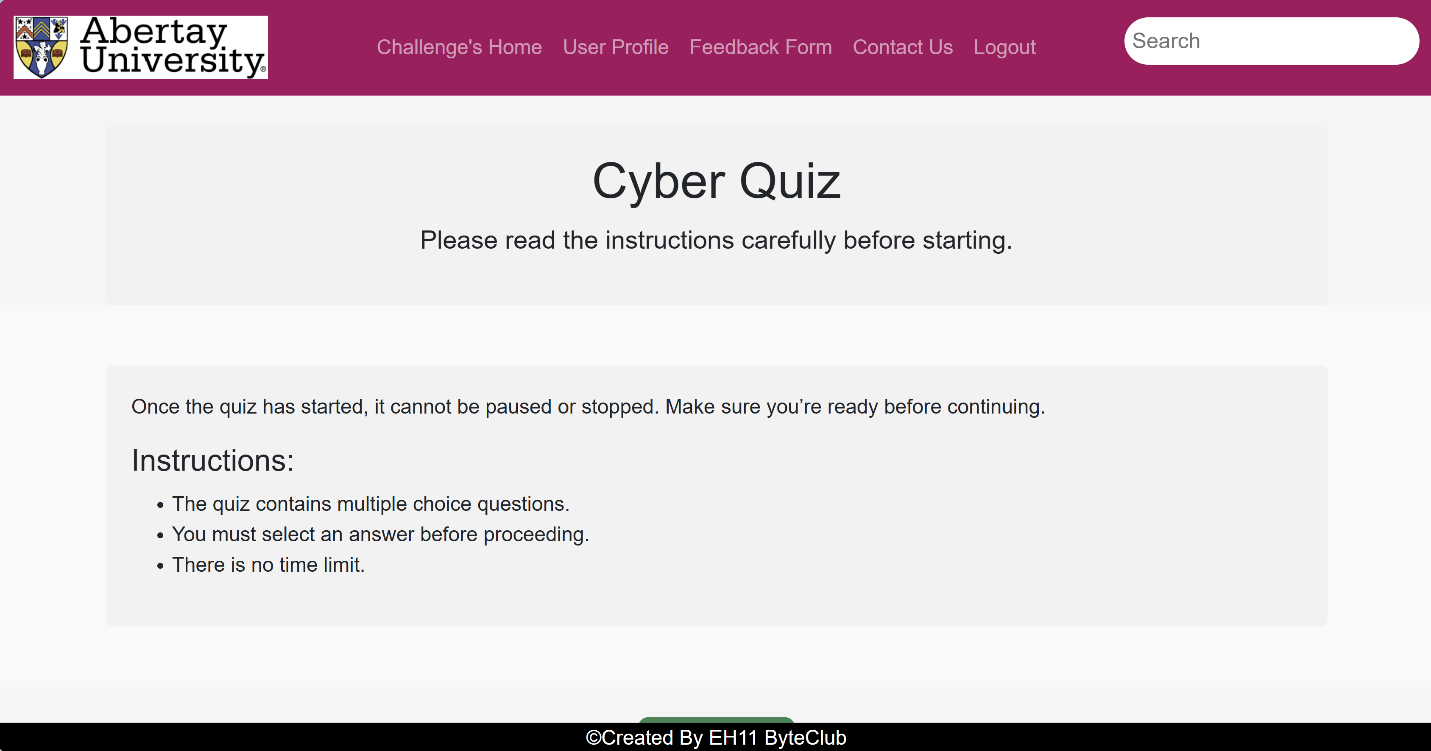
Figure 8 is a page that allows for direct communication between the admin and the user on the site is the feedback response page. This page like mentioned above is a place for users to provide feedback good or bad which the admin can access in one place with an easy-to-read format as well as the page categorises it and allows for a response to be sent if need be.

*Figure 9 – Analytics page*

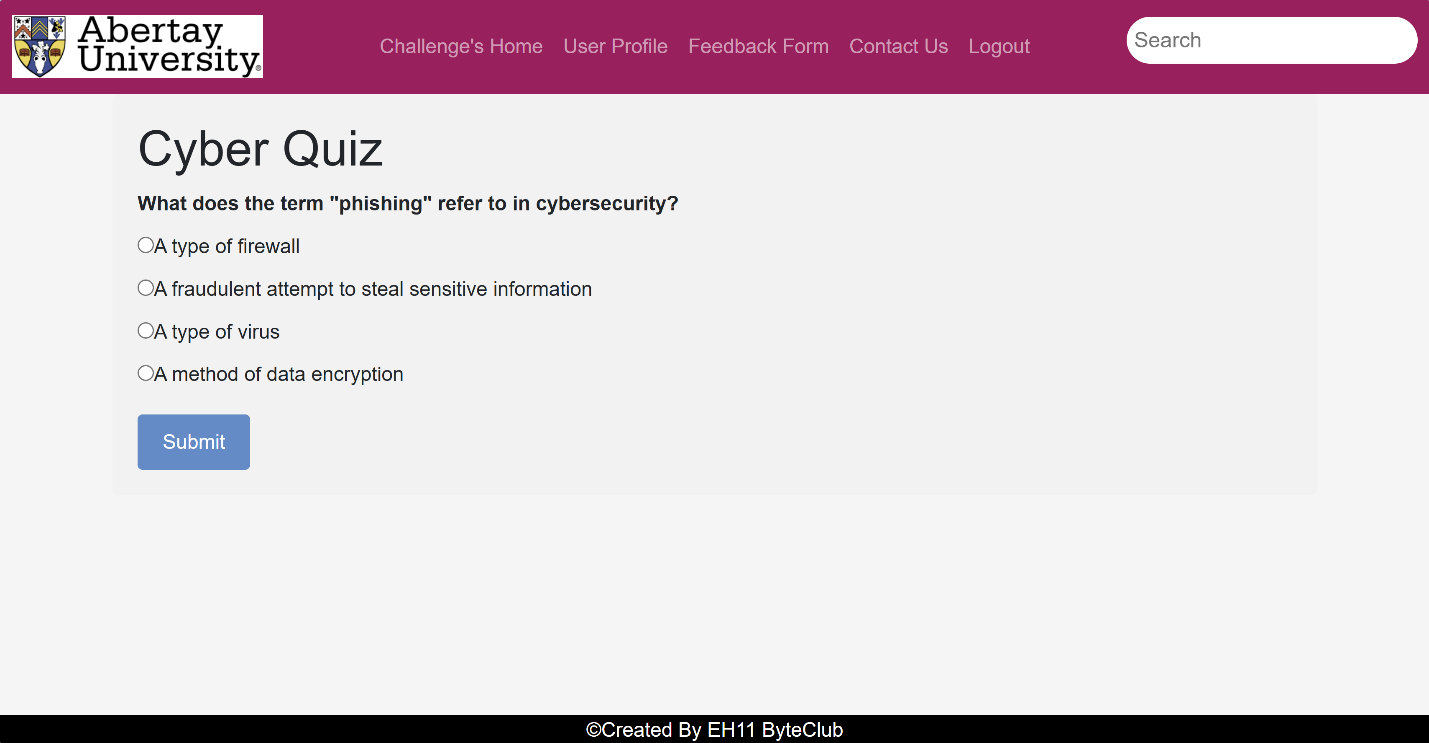
The final page that the administrator has access to (figure 9) is the score management page where they can see all the data collected from the quizzes, this data being the number of questions the users got right. This allows for the administrator to see what questions people are struggling with and which one's people are getting right. There’s also an option for the administrator to add a new quiz simply by uploading a file with the new quiz inside.

*Figure 10 – Challenge choosing page*

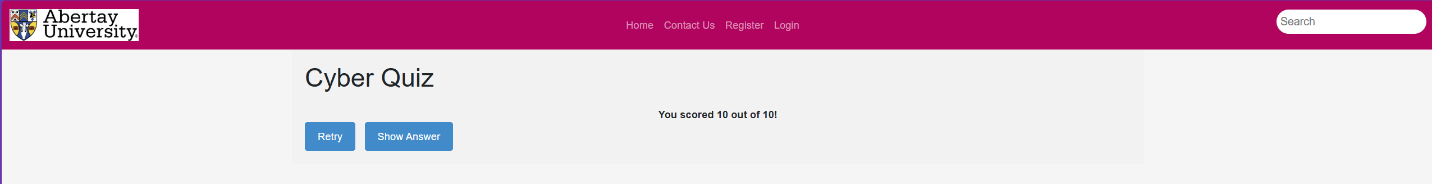
The final page that a user will have access to from the nav bar is the challenge’s home page (seen in figure 10) where they are able to pick which challenge they would like to undertake. There are currently two challenges to pick from as on the current state of the website, the cyber quiz and the coding quiz, both designed to test the user's knowledge using multiple-choice questions.

*Figure 11 – Informational page*

When the user clicks on a quiz (in this example the cyber quiz) they are greeted with an informational page where the quiz is explained to them to make sure that they are fully aware of what to expect from the quiz, these instructions can be seen in figure 11.

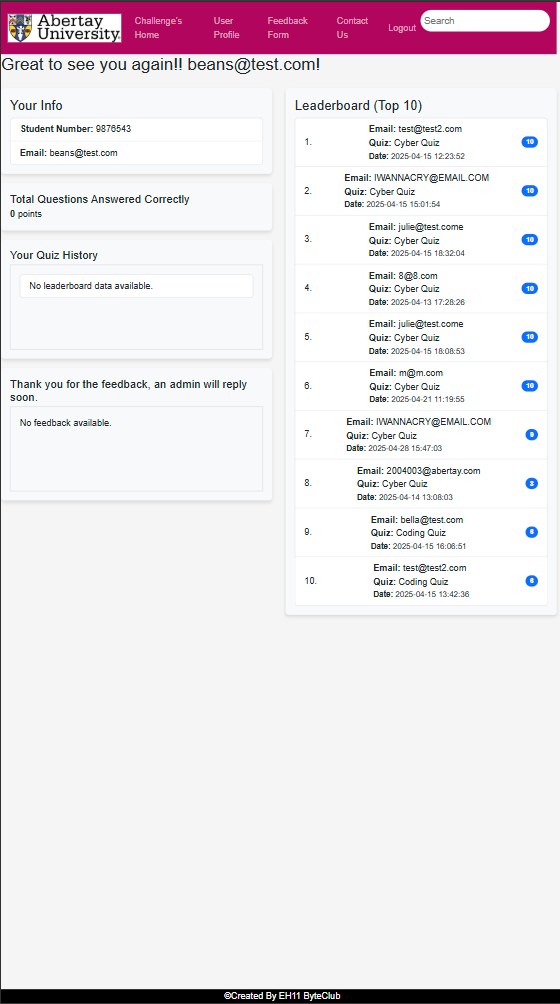
*Figure 12 – The Quiz*

After reading through the informational page and the user has clicked submit, they are taken to the final part of the website, the quiz (seen in figure 12). This page consists of 10 multiple choice questions that only appear after the previous one has been completed, this once complete will provide the user with their score as well as providing the answer for any questions they got wrong. As seen in figure 13.

*Figure 13 – Final score at the end of the quiz*

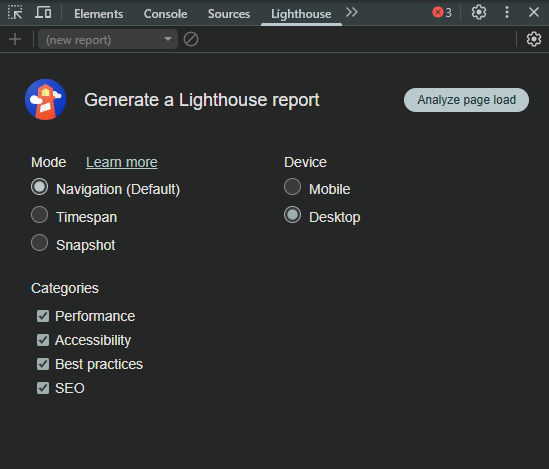
### Useability testing

As this website is going to be run on many different devices it is important that the website can run on smaller devices. To accommodate for this the website uses bootstrap to make the pages responsive and to show this the user profile screen is used as an example (seen in figure 14) where the website is running on an iPhone se and the site changes to accommodate the smaller screen.



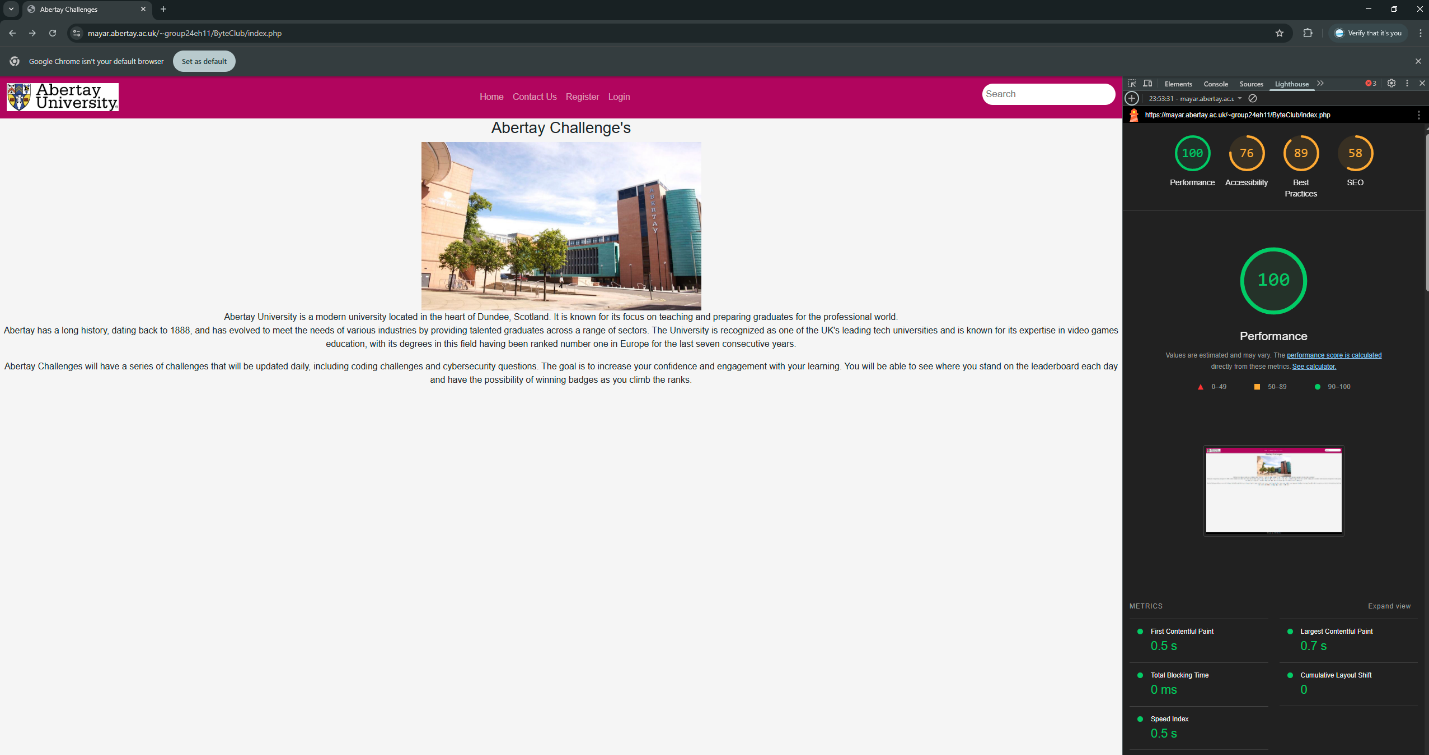
*Figure 14 – Website as seen on an Iphone*

There was also a test done using Lighthouse. This is an open-source tool that allows for testing of performance and accessibility and can be accessed on google chrome through the developer tools. In figure 15 the setup for this process can be seen.



*Figure 15 – Lighthouse setup*

After finishing the scan, it provides easy to read indicators for how well the site is running and from the testing it shows that the website performed very well with it getting a 100 on performance and the lowest score being in SEO (seen in figure 16) which checks if the website is following basic search engine optimisation, which as a prototype is bound to happen. Further information can be found in the discussion section.

*figure 16 – Lighthouse scores*

# Discussion

## General Discussion

The web app has been very successful with minimal issues. The web app adheres to the Mobile Web

App Best Practices (2010) (W3C, 2010) On the initial download, the app will present the permission

statement, explaining the data it requires. This gives the user control. A website privacy policy and

GDPR compliance statement are included. (European Union, 2016)

User details and game/scores information are securely on the SQL database. A major risk is SQL injection, we have prevented this by sanitizing and validating user inputs. (OWASP, 2021). Passwords are securley salted and hashed. The database can only be accessed by the admin accounts only; there are no delete rights on the user side to the database.

The integration of user analytics will give the client valuable insights into the student behaviour and engagement with the app., when it comes to assignments or tests then they can properly see if there has been a positive impact. Based on the research the outlook for success is very promising. The web app was tested using Lighthouse. *Chrome Lighthouse (2025)* This is a tool built into Google that tests web sites and web apps, if there are any problems it lets you know. The test gives a score back on 5 key areas. After testing the score was 73/100 which is good. The SEO score of 58 indicates room for improvement such as meta tags, descriptive link text and improved page structure. This would improve discoverability.

Performance – How fast the website loads and responds. 100/100

Accessibility - Is it inclusive? There should be no barriers to accessing the web to people who may have physical or situational disabilities. 78/100

Best Practices – If the website uses safe and modern coding standards. 89/100

SEO- Search Engine Optimization, can search engines find and understand the web app. 58/100

Progressive Web App (PWA) - If it works like an app on phones. No score available.

A study by Cigdem et al. (2024) explored the impact of leaderboard gamification on student engagement and academic achievement in engineering education. and published last year 2024.

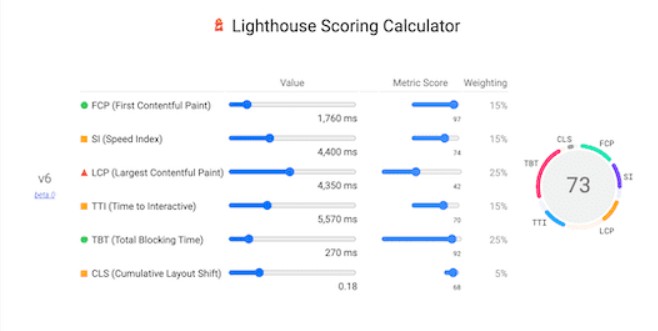
At a military university in Turkey with 159 2nd Year Engineering students. Groups were split in 2 78/81 the data was adjusted using ANCOVA, so that both groups had baseline knowledge. The experimental group had a gamified web app with a leaderboard. The other group Control only had text quizzes. This study was carried out over 8wks. The end results show the students with the gamified and leaderboard app completed more quizzes, and engaged more with the app. The test at the end of 8wks included practical work and an exam. These results are from the test. The control group with a text quiz average score was 60.51, the group that had the gamification and the leaderboard their average score was 68.20, showing almost 8 points between them which is a significant difference. Results show the gamified group scored higher in both and their answers were consistent. This confirms that gamified web apps that have a leaderboard will help the students stay engaged in learning and have better results in coursework.

By providing the gamified system, complex technical topics become more approachable and enjoyable. There is a trend in education to use these learning apps, from young children all through adulthood.

Performance Testing Results.

Lighthouse is the tool that was used in the testing and auditing process, it helps you to understand how well the web app is performing across key areas. Lighthouse is a free tool within Google Chrome under DevTools. Multiple audits were performed with the results in the appendix. Early performance tests provided a score of 73. Audits taken later showed the majority 100% scores. The pages are tested manually as Lighthouse is not designed to test the whole web app at once. *Google Developers. (2024*

Fig 16 The overall performance test shows the web application loads quickly, memory usage while the page loaded was stable, with JavaScript completed in under 1 second. This is a great result for web responsiveness and efficiency, and exactly what the client asked for in a fast friendly web app.



*Figure 17*: Lighthouse Technical Test (Google Lighthouse, 2025)

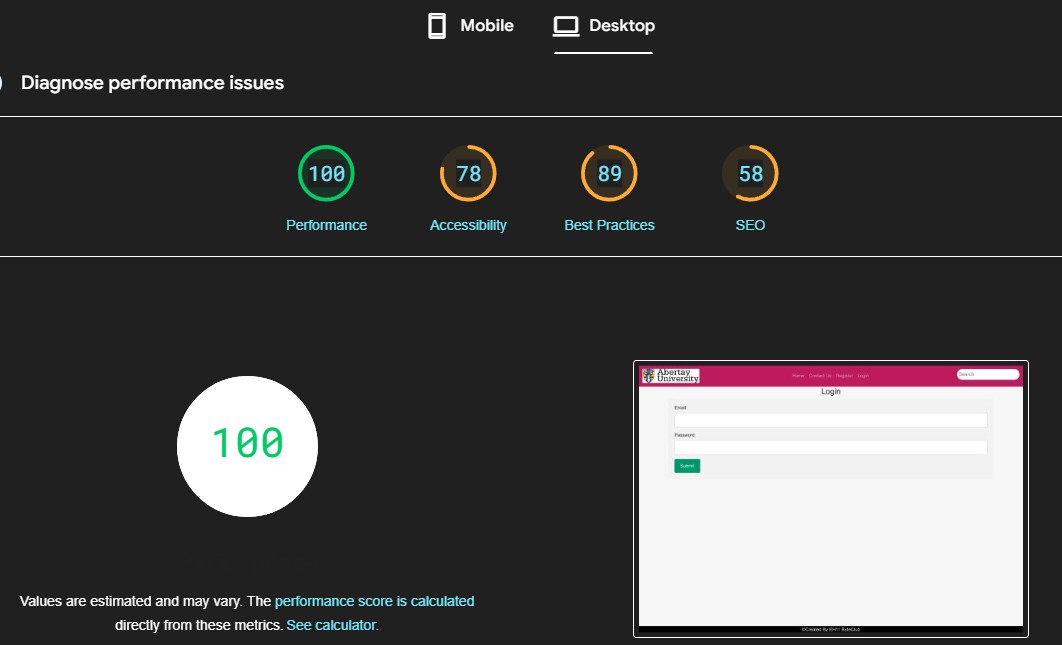


Figure 18: Performance score (Google Lighthouse, 2025)



Figure 19: Lighthouse performance audit. (Google Lighthouse, 2025)

The standards used for the implementation of this web application is OWASP top 10. This web page has information on all aspects of security and design strategies to produce a secure web app. The web app went through a comprehensive test, failing on two, both of which need access to the server. Otherwise, it would have been a pass. Advice is to sign up to the email alerts for security information.

Injection – This includes SQL database we have tested the site against this.

Vulnerable and Outdated Components – Updating software is paramount, inclusive of the server, web app and the database. The versions of the components we have used are included.

Key Performance Indicators.

A key performance is that web app aligns with the current student portal which centralizes access to the students' academic resources, and the campus digital services. From a business perspective the web app is expected to reduce administrative duty by automating some of the educational materials which will help staff save time and money. The design reduces the need for extensive training.

|  |  |  |
| --- | --- | --- |
| Client Requirement | Performance Indicator | How It Was Measured |
| Easy to use web app | (SUS) Score > 68 | System Usability Scale survey with users |
| Improve student engagement | Increased number of quizzes completed | Data analysis from SUS |
| Fast web app | Score >80 | Google Lighthouse audit |
| Boost student learning and outcomes | Higher scores in tests | Research before and after test |
| Mobile compatibility | Mobile responsive design | Tested on different devices |
| Protect personal data. | Official Certificate GDPR | GDPR Certificate |

*Figure 20*: Key Performance Indicators

## Conclusions

* It provides user satisfaction with an intuitive and responsive design, safeguarding the clients position in the education market.
* Data analytics provide valuable insights to the users who may be struggling/excelling.
* It has the flexibility to grow and support other degree programmes.
* Feedback from the user test data confirmed the app is well received.
* Safeguards user data and is up to date with cyber security standards.

## Future Work or Countermeasures

* Upgrade the security when we gain access to the client's server.
* Advanced data analysis, for more insights into user behaviour.
* Storing the games in the database.
* Advice the client to familiarise themselves with OWASP Top 10, also to download and use the OWASP Dependency Check Tool, this scans for security issues in 3rd party libraries that the web app uses., it can be used automatically

## Call to action

* We would like to offer 3 months technical support.
* An information session with students and staff.
* http://www.ByteClub.co.uk

# References

Apostolopoulos, A., 2019. *The 2019 Gamification at Work Survey.* [Online]   
Available at: https://www.talentlms.com/blog/gamification-survey-results/  
[Accessed 15 April 2025].

Cuervo-Cely, K. R.-C. F. &. R.-E. J., 2022. Effect of Gamification on the Motivation of Computer Programming Students. *Journal of Information Technology Education: Research,* Volume 21, pp. 1-23.

Flinders, K., 2019. *Computer science undergraduates most likely to drop out.* [Online]   
Available at: https://www.computerweekly.com/news/252467745/Computer-science-undergraduates-most-likely-to-drop-out  
[Accessed 22 March 2025].

Li, M. M. S. a. S. Y., 2023. Examining the effectiveness of gamification as a tool promoting teaching and learning in educational settings: a meta-analysis. *Frontiers in Psychology,* Volume 14, p. 1253549.

Tech, W., n.d. *Why do students drop out of tech studies?.* [Online]   
Available at: https://wawiwa-tech.com/blog/why-do-students-drop-out-of-tech-studies/  
[Accessed 22 March 2025].

Chrome Lighthouse [Online]

Available at: https://developer.chrome.com/docs/lighthouse/overview

[Accessed] 27 April 2025]

W3C. (2010). Mobile Web Application Best Practices. [Online]

Available at: <https://www.w3.org/TR/mwabp/>

[Accessed] 27 April 2025]

OWASP Top 10

Available at: OWASP Foundation. (2021). OWASP Top 10 – 2021: The Ten Most Critical Web Application Security Risks. <https://owasp.org/Top10/>

[Accessed 27 April]

Cigdem, H., Ozturk, M., Karabacak, Y. et al. Unlocking student engagement and achievement: The impact of leaderboard gamification in online formative assessment for engineering education. Educ Inf Technol 29

Available at: <https://link.springer.com/article/10.1007/s10639-024-12845-2>

[Accessed 27 April]

European Union. (2016). General Data Protection Regulation (GDPR). Regulation (EU) 2016/679.)

Available at: https://commission.europa.eu/law/law-topic/data-protection\_en#funding-to-support-the-implementation-of-the-gdpr

[Accessed 27 April]

# Appendices

## Appendix A - Deliverables & requirements (Required)

**Agreement Form: Project Deliverables**

|  |  |
| --- | --- |
| **Group Name, Names of Team Members, and Program** | **EH11 – Byte Club**  **Team Members – Snow White, Julie Whyte, Jack Tully, Joshua Wilkins, and Harvey Williams** |
| **Subject specialist’s Name (Client)** | **Xander Purvis** |
| **The deliverables listed below will be submitted by the team by the due date.** | |
| **Part A deliverables** | **To be agreed by program specialist and team, for example:**   * Appropriately commented website/source code * Database * Database Creation Code * Investigation report * GitHub Repository * GDPR Policy * Presentation |
|  | * Requirements Specification, signed off by the program specialist (see overleaf) * Testing Log * Evaluation of project |
| **Subject specialist’s (Client) signature** |  |
| **Team members’ signatures** | **Snow White**  **Julie Whyte**  **Joshua Wilkins**  **Harvey Williams**  **Michael Wilde** |

**Agreement Form: Requirements**

Group Name: Byte Club

Team members (print): Snow White, Julie Whyte, Jack Tully, Joshua Wilkins, Harvey Williams, and Michael Wilde

Project Title: Abertay App Challenge

Please refer to the attached documentation for full details of the project. The requirements are listed in Table 1. The signatures below indicate that the requirements for this project have been agreed by the project stakeholders.

Any changes to the project documentation should be made using the correct change authorization procedure agreed with the program specialist.

Table 1

|  |  |
| --- | --- |
| **ID** | **List of Agreed Requirements (fill in)** |
| 1.  2.  3.  4  5.  6.  7.  8.  9. | The app or website should contain at least two challenges/games.  The first should present the user with a coding problem and allow the user to input their solution. The second should present the user with a cybersecurity scenario or question and allow the user to input their answer.  Once the user has submitted their answers, they should be informed whether their answer was current or not, and how many students got the answer. Correct inputs should result in the user earning points and possibly badges.  To ensure the life span of the app/website the daily challenges should be stored in the app/website/database with two new challenges being revealed to the users at the start of the day.  Admins should be able to add as many new daily challenges as they wish at any time.  To determine if interactions with the app positively impact students, there may be a method for data to be collected that measures student interaction.  Users should be able to register to the app with their username, student number, and password, and all challenges should be tagged by the admins who submit them so that the difficulty of the challenges can be tracked.  The app/website can utilise Abertay’s branding  The app/website should capture sufficient data to be able to ascertain whether students interacting with it are indeed building confidence and skills over time. |

|  |  |  |
| --- | --- | --- |
| **Stakeholders** | **Signatures** | **Date** |
| Team members | Snow White  Julie Whyte  Joshua Wilkins  Harvey Williams  Michael Wilde  Jack Tully | 03/02/2025  03/02/2025  03/02/2025  03/02/2025  03/02/2025  03/02/2025 |
| Subject Specialist |  | 17/02/2025 |
| Client (if applicable) |  | 17/02/2025 |

## Appendix B - connectionString.php

A computer screen shot of a program code

AI-generated content may be incorrect.