University of Aberdeen

School of Natural and Computing Sciences
Department of Computing Science
2024 – 2025

Programming assignment – Individually Assessed (no teamwork)

Title: JC2503 – Web Application Development	Note: This assignment accounts for 30% of your total mark of the course.

Learning Outcomes

On successful completion of this component a student will have demonstrated to be able to:

- Write and run basic Web applications
- Judge how websites and web applications should be structured
- Apply techniques to support producing correct JavaScript code in Node.js platform
- Identify appropriate platforms and modules to be used in their applications
- Apply appropriate programming concepts for creating a new web application

Information for Plagiarism and Collusion: The source code and your report may be submitted for plagiarism check in *Codio*. Please refer to the slides available at *MyAberdeen* for more information about avoiding plagiarism before you start working on the assessment. Excessive use of generative AI (GenAI) tools, such as ChatGPT or DeepSeek, for writing the code or the report can also be considered as plagiarism. You can use GenAI for proofreading or assisting to write small pieces of code, but the general structure of your code and report should be developed by yourself. As this is individual work, submitting similar work with another student can be considered as collusion. Also read the following information provided by the university:

https://www.abdn.ac.uk/sls/online-resources/avoiding-plagiarism/

Extension Policy: According to the extension policy of University of Aberdeen, teachers are not allowed to grant deadline extensions for coursework assignments. Extensions may be requested from the school administration by e-mail: uoa-ji-enquiries@abdn.ac.uk.

Extensions require strong justifications (such as serious illness or grievance). For unauthorised late submissions, 8% of the mark will be deduced for the first 24 hours, and then 4% for each consecutive day, up to 7 days. Submissions more than 7 days late will not be marked.

Introduction

In this assignment, your task is to build a simple website containing information about yourself, as well as an interactive quiz application for two players to play against each other. The website should contain three pages. The first one (root) is the introduction page where you tell briefly who you are, what you study, etc. The second page will contain more detailed information about you from the chosen point of view. You can, for example, explain your professional interests, what courses have you taken, and what topics in your field you are especially interested in. Alternatively, you could also explain something about your leisure activities and achievements for example in some hobby that is important for you. The third page should contain a quiz application, with more detailed requirements explained later in this document.

The webpages should include a navigation system allowing user to move easily from each page to another. The first two pages can be static pages implemented in HTML and CSS, but for the third page, you need to implement also back-end functionality using JavaScript on Node.js platform. The final project, including HTML, CSS, and JavaScript code and the project report, should be submitted in Codio. The high-level page structure is shown in the picture below.

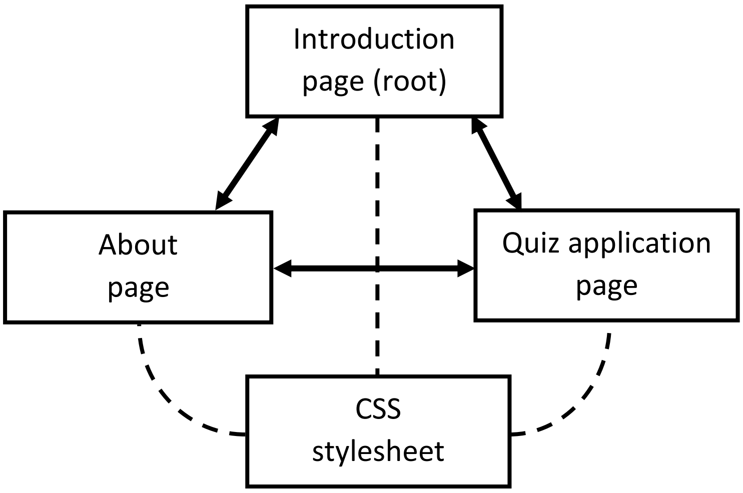

Figure 1. Illustration of the website structure.

General Guidance and Requirements

Your assignment code and report must conform to the requirements given below and include the required content as outlined in each section. You **must** supply a written report, along with the corresponding code, containing all distinct sections/subtasks that provide a full critical and reflective account of the processes undertaken.

While we are not marking your use of Git in this assignment, you are strongly encouraged to do this work inside a Git repository. By doing that you gain the benefit of rolling back your edits, or doing work in branches, and then merging it to the main branch as you accomplish tasks along the way.

This assignment is done in a Codio box, as before. If you decide to work on your own device, then be sure to move your files to Codio regularly, and to also build and run the application, so that the files to run the application are there when we mark it.

Note that it is your responsibility to ensure that your code runs on Codio. If you do the assignment on another device and then move it to Codio, you need to ensure it runs on Codio, and that it runs as expected. By default, your server should run using **npm start**. If your implementation uses some other command to start the server, it must be mentioned in the report. The Box URL in Codio should be configured to connect to your server properly, so that your implementation can be tested directly without additional configuration steps. Note that you should avoid using the full Codio box URL (like *something-something-8080.codio-box.uk*) directly in your code, because the teachers will test the code in their box with different URL. *Use relative URLs only!*

Requirements and Marking Criteria for the Website Structure [20 marks]

The Introduction page (root page) should include a personal introduction and an overview of the page in general. The About page should include more detailed information about an aspect of yourself you choose, for example your professional or private interests. You can decide the content for the Introduction and About pages yourself. The third page will contain quiz application with more specific requirements given in a separate section, but the HTML code for the quiz application should also follow the criteria below. There should be enough content to demonstrate that you can create HTML pages with different elements, such as header, footer, paragraphs, images, sidebar, etc.

The marking criteria for the web structure are the following:

- The HTML code is written in valid HTML 5 and passes HTML validation **[10 marks]**. For any validation errors, 2 marks are deducted, and for any warnings, 1 mark is deducted. Marks can also be deducted if the code is very short or trivial. Minimum mark is 0.
- There is a functional navigation system between the pages **[5 marks]**.
- The content is appropriate and organised logically **[5 marks]**.

Requirements and Marking Criteria for the Styling and Design [20 marks]

Your webpages must be styled with CSS to improve the presentation of the content and aid the user in using the quiz application. If you use Bootstrap, you must include a custom style sheet with valid CSS modifying the default Bootstrap styling.

You should include enough CSS code to demonstrate that you can use CSS to define different colours, layouts, etc. You can choose the colours and style of the website yourself, but you should follow the chosen style consistently on all the three pages. While creativity is appreciated, usability is more important than beautiful design: you should avoid using large videos or flashy animations that can be distractive and slow to download.

The marking criteria for the style and design are the following:

- CSS code passes CSS validation **[10 marks]**. For any validation errors, 2 marks are deducted, and for any warnings, 1 mark is deducted. Marks can also be deducted if the code is very short or trivial. Minimum mark is 0.
- Visual appearance, responsiveness, accessibility, consistency of the design **[10 marks]**.

Requirements and Marking Criteria for the Quiz Application [30 marks]

The quiz application should allow users to connect to your website and challenge another user for a quiz game. When you enter the page, you should first write your name. Then, the names of the other players online are shown, and you can challenge another player by clicking the name. Other players can also challenge you. The player challenged can accept or reject the challenge.

The game itself is simple. Quiz questions are shown to both player with answer options. The player who clicks the right answer faster will win the round and gets 2 points, while the opponent gets 0 points. Player who clicks a wrong answer will get 0 points, and the opponent will get 1 point instead. At least five questions will be presented, and in the end, the winner is the player with more points. Between two questions, there should be a short break (about 5-10 seconds) to show the result of the round is shown before showing the next question. Figure 2 below illustrates the flow of the game.

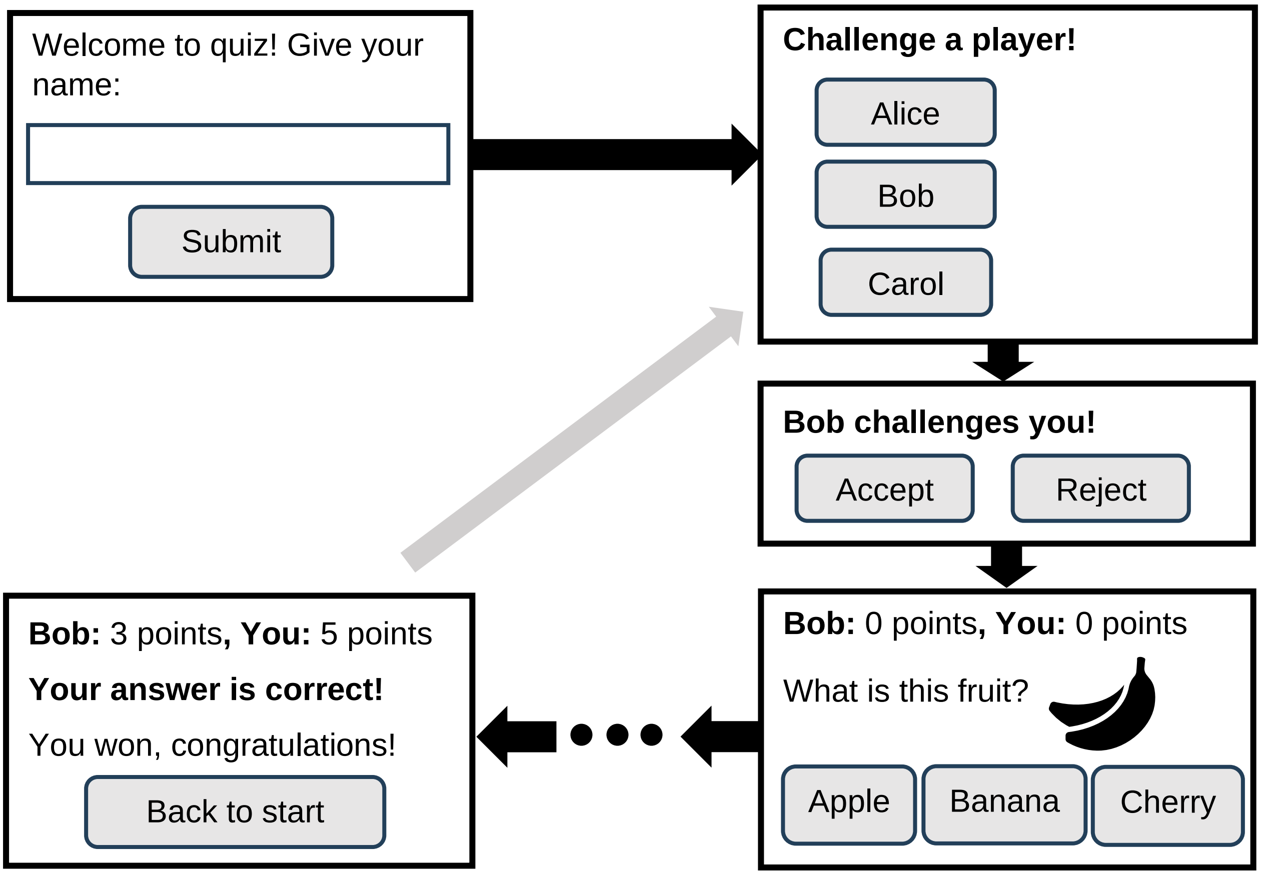

Figure 2. Flow of the quiz application illustrated.

In the example above, the question is to recognise some fruit from a picture, but in your quiz you can choose the theme yourself. The questions can be about general knowledge, recognising national flags, logos of football clubs, brands for specific products, or whatever theme you find interesting. You should have at least five questions. Different functionalities of the game are marked as follows:

- There is a mechanism for keeping track of players, listing the players and allowing them to challenge each other **[10 marks]**.
- The game logic is functional as explained above **[10 marks]**.
- The application supports multiple concurrent games between different players **[5 marks]**.
- General usability of the game application: at each stage, players are properly informed about the status of the game, the user interface is responsive, the delay between questions is reasonable, etc. **[5 marks]**.

Partial marks can be awarded if a feature is implemented, but it does not work correctly due to a programming error. Note that the style of the quiz application will be assessed as part of the styling and design for the whole website.

Requirements and Marking Criteria for the Project Report [30 marks]

You should write a report by editing the **readme.md** file that is autogenerated in Codio. Your report should describe the overall design of the website and the quiz application, including how the quiz application client communicates with the server, with respect to handling events and using Socket.IO. The report should also include a section for reflection, detailing challenges faced, and lessons learned during the work. Appropriate references should be included, and if you have used GenAI tools at any stage, this should be acknowledged. The recommended length of the report is 500-700 words. Illustrations are encouraged, but their absence will not be penalised.

The marking criteria for the report are the following:

- Structure and completeness (all the required aspects are covered) **[10 marks]**.
- Clarity and readability (language and illustrations are understandable) **[10 marks]**.
- The report is consistent with the implementation **[5 marks]**.
- References are adequate **[5 marks]**.

Submission

You should submit the code and the report in Codio, using the Codio box linked in MyAberdeen for the coursework assignment. **The deadline is 30 May 2025 at 23:59 (Beijing Time).**

Contact

For any questions or clarifications, you can contact the course teachers: Dr Xiaonan Liu (xiaonan.liu@abdn.ac.uk) for AI, Dr Yuan Wen (yuan.wen@abdn.ac.uk) for BMIS, and Dr Jari Korhonen (jari.korhonen@abdn.ac.uk) for CS.