UNIVERSITY OF ABERDEEN

University of Aberdeen

School of Natural and Computing Sciences

Department of Computing Science

2023 – 2024

Programming assignment – Individually Assessed (no teamwork)

Title: CS2053 – 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. The use of large language models, such as ChatGPT, for writing the code or the report can also be considered as plagiarism. In addition, 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/

Introduction

In this assignment, your task is to build a simple website containing information about yourself, as well as an interactive quiz application. 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.

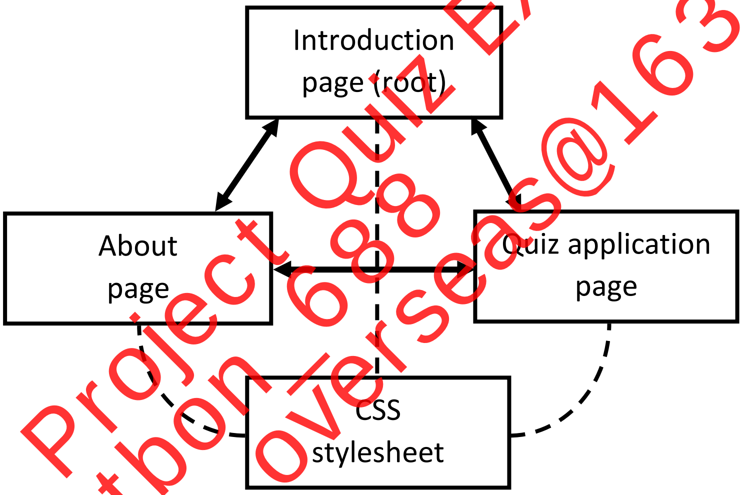


Figure 1. Illustration of the website structure.

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 documentation. 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 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 Web Structure [25 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, but the page structure should follow the technical requirements specified in the marking criteria. The third page will contain a 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 must be written in valid HTML 5 and pass HTML validation **[10 marks]**. For any validation errors, 2 marks are deducted, and for any warnings, 1 mark is deducted.
- There should be a functional navigation system between the pages **[5 marks]**.
- Semantic structure of the code should be logical **[5 marks]**.
- Useful extras, such as using front-end JavaScript for interactive elements **[5 marks]**.

Requirements and Marking Criteria for Styling and Design [25marks]

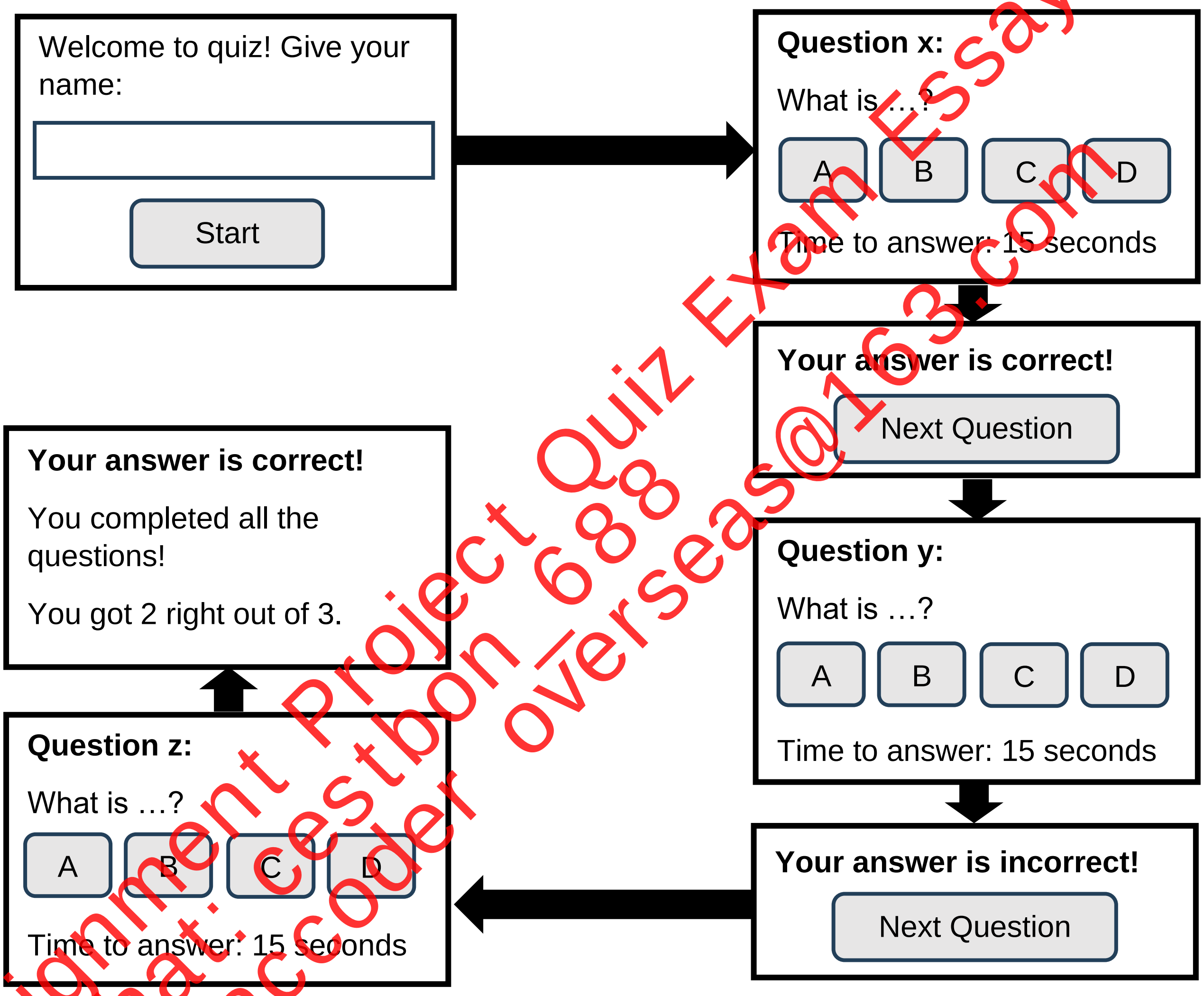
Your webpages must be styled with CSS to improve the presentation of the content and aid the user in using the quiz. If you use Bootstrap, you must include a custom style sheet with valid CSS modifying the default Bootstrap styling. You can choose the colours and style of the website yourself, but you should follow the chosen style consistently on all the three pages.

The marking criteria for the web structure are the following:

- The CSS code must pass CSS validation **[10 marks]**. For any validation errors, 2 marks are deducted, and for any warnings, 1 mark is deducted.
- Page design (sensible padding, responsive layout, consistency of the design) **[10 marks]**.
- Documentation (comments) in the code **[5 marks]**.

Requirements and Marking Criteria for the Quiz Application [25 marks]

The quiz application should allow users to connect to your website and answer the quiz questions. Each question should be shown for a specific maximum amount of time, for example 15 or 30 seconds, or until the user chooses an answer. Then, the result should be shown until the user presses the button to move to the next question. If the time runs out before the user submits the answer, it is counted as an incorrect answer. When the user submits the answer, they will get immediately a response from the server and they will be shown whether the answer was correct or not. After the user completes all the questions, the final result is shown. Figure 2 below illustrates the full quiz cycle.



Welcome to quiz! Give your name:
Start
Question x:
What is …?
A
B
C
D
Time to answer: 15 seconds
Your answer is correct!
Next Question
Question y:
What is …?
A
B
C
D
Time to answer: 15 seconds
Your answer is incorrect!
Next Question
Question z:
What is …?
A
B
C
D
Time to answer: 15 seconds
Your answer is correct!
You completed all the questions!
You got 2 right out of 3.

Figure 2. Quiz cycle from the user's perspective.

In the example above, there are only three questions, but in your quiz, there should be at least ten questions. You can choose the theme of the quiz yourself. It can be about general knowledge, recognising national flags or brands for specific products, or whatever theme you find interesting.

- The user is shown the full set of questions sequentially, as described above, and the user's answers are emitted to the server. In this stage, timing is not yet required. **[5 marks]**.
- The user is forced to move to the result when the time runs out. **[5 marks]**.

- The application shows a countdown or an inverse progress bar to inform the user continuously about the remaining time **[5 marks]**.
- The total number of correct results and the total time used for answering is recorded and reported to the user after the last question **[5 marks]**.
- The server maintains a leaderboard of the users with most correct answers (the fastest user is given advantage if there is an equal number of correct answers), and the leaderboard is shown to the user after the last question **[5 marks]**.

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

Requirements and Marking Criteria for the Project Report [25 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, as well as the challenges faced during the development of the website. You should also explain how the quiz application client side communicates with the server, with respect to handling events and using Socket.IO.

The marking criteria for the report is the following:

- Structure and completeness (all the aspects are covered) **[5 marks]**.
- Clarity and readability (the language is understandable) **[5 marks]**.
- Design explained **[5 marks]**.
- Challenges discussed **[5 marks]**.
- References to the sources **[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 will be announced separately.**

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.