

Cloud Based Web assignment

Rhys Hughes

by Rhys Harri Hughes

Submission date: 14-Jan-2024 06:06PM (UTC+0000)

Submission ID: 221917501

File name:

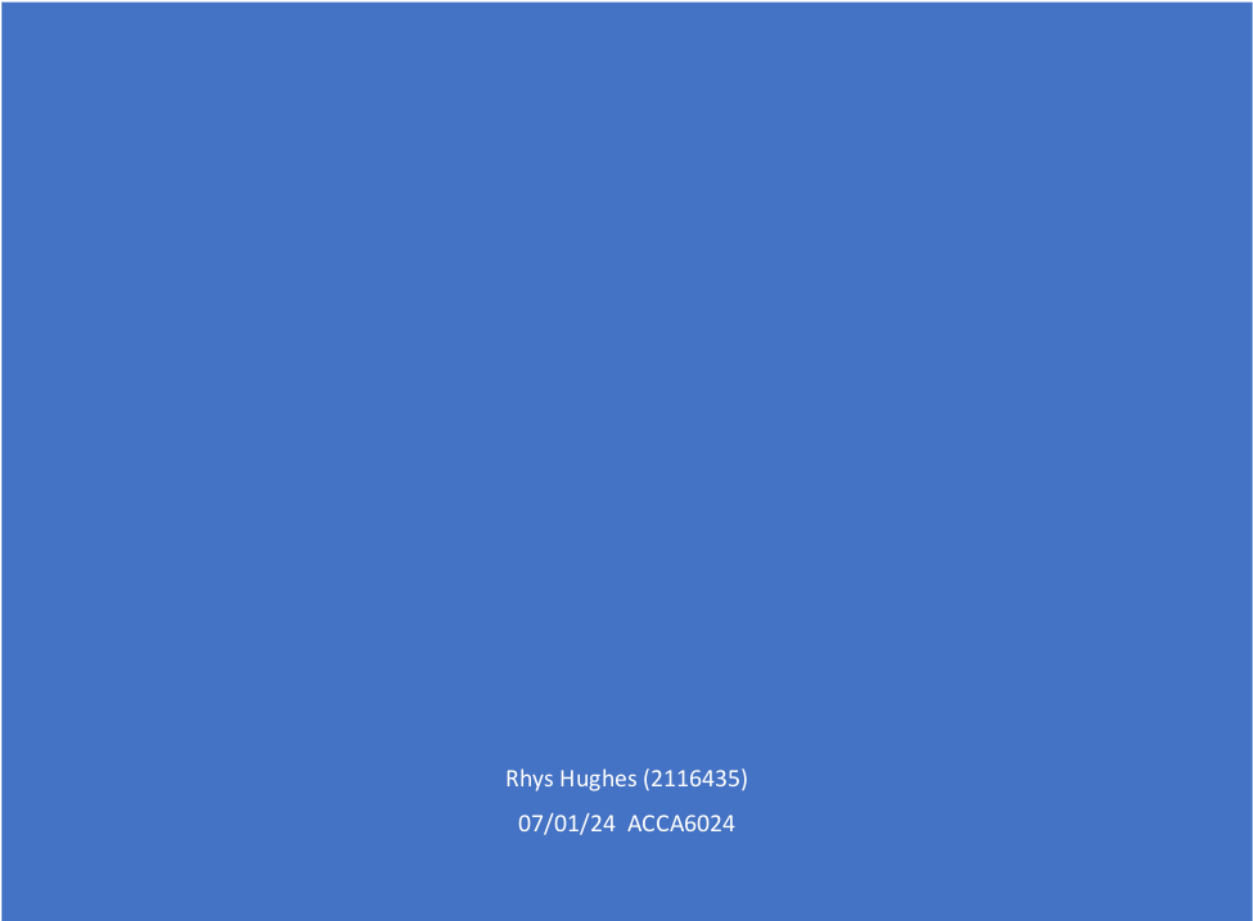
50363_Rhys_Harri_Hughes_Cloud-Based_Web_assignment_Rhys_Hughes_536492_2008517945.docx
(267.97K)

Word count: 2205

Character count: 11069



ASSIGNMENT 1 & 2: CLOUD BASED WEB SERVICES



Rhys Hughes (2116435)
07/01/24 ACCA6024

Table of Contents

Table of Contents	1
1.0 Introduction	2
2.0 User Requirements	3
2.1 Data requirements	3
2.2 website requirements	3
3.0 Design	4
3.1 Basic layout	4
3.2 Edit mode	4
3.3 Create new book	4
3.4 Filtering	4
4.0 Testing	5
4.1 Testing tables	5
5.0 Conclusion	12
6.0 References	13
7.0 Figures	14

1.0 Introduction

This report will detail the creation of a cloud based web service that uses Ajax and a REST API to create a dynamic webpage. The organisation selected is a bookstore called magic quill Books. To this end the user requirements such as the data and design requirements are defined. The JavaScript is also tested to ensure that it works and meets the organisation's needs.

2.0 User Requirements

2.1 Data requirements

For this case 8 pieces of data were chosen to identify and describe the individual books. These are the Unique Identifier called "Id" which is used as the identifier in the database, the "Name" of the book which is a text string, the "Description" which is a short description of the contents, also a text string, the release date "ReleaseDate" which is stored in a date format, ISBN which is numeric, Genre, and Audience which are both text. These were chosen to have a variation in data types. On the HTML side the name is a text input, the description is a textarea, the release date is a date input, the ISBN is a number input, and the genre and audience are dropdown selections.

2.2 website requirements

The website requirements are designed to get as much information to the user as possible and to have the ability to create (POST), edit (PATCH), delete, and filter the different books. It is also designed around one dynamic page. Also, for the website to be more responsive such as with resizing and for quicker design times, Bootstrap will be used.

3.0 Design

3.1 Basic layout

The basic webpage layout can be seen in Figure 1. The header is at the top with the buttons to access the Bootstrap Offcanvas filter and create pages. In the middle is the table containing the books with an edit button on the side. This table is dynamically created by getting the information from the database. It automatically creates new rows for each of the books.

3.2 Edit mode

It was decided to have per record editing. Clicking on the edit mode changes that specific books table row to allow for editing. This can be seen in figure 2. Input boxes replace the book information, and the information is automatically inserted into them. It also replaces the edit button with three different ones. The first is a SAVE button which patches the data into the database. The second is the discard button, this discards any changes by reloading the table, the final one is the delete button which deletes the record from the database.

3.3 Create new book

The create offcanvas modal can be seen in Figure 3. The modal is opened by clicking the create button in the header. In the modal, all the data required can be inputted. Once inputted it automatically refreshes the table to show the new input data.

3.4 Filtering

The filtering modal can be accessed by clicking the filter button in the header which can be seen in figure 4. Here it allows the user to select how they want to filter and a reset filter option. It was decided against going for multiple option filters and instead having per type filtering.

4.0 Testing

Various information is required for software testing. The primary information required is the project's name, a detailed description of what is being tested and the test itself [1]. The other information required are the specific features tested and the criteria for success. The testing document that will be used contains the test title, feature description, test description, expected results, and test results, success or failure (checkbox).

4.1 Testing tables

Test title:	Page load displays table		
Feature description	LoadBooks() gets the books from the database and creates a table containing them all		
Test description	Refresh the page and see whether the table loads		
Expected results	Table loads with book data		
Test results	Table loads with book data	Success?	✓

Test title:	Edit mode toggles on correctly		
Feature description	The edit button on the table enables the toggle mode that takes the data from the books and sets them as the default for the inputs.		
Test description	Edit button gets clicked		
Expected results	Edit button gets clicked and the information from the row is inserted into the input fields		
Test results	Edit button gets clicked and the information from the row is inserted into the input fields	Success?	✓

Test title:	Edit mode discard		
Feature description	If discard is clicked the table is reloaded and edit mode is exited.		
Test description	Once in edit mode the discard button is clicked, the table should reload.		
Expected results	Discard button is clicked and any data in the input fields should not be sent to the database and exit mode is exited as the whole table reloads.		
Test results	Discard button is clicked and any data in the input fields should not be sent to the database and exit mode is exited as the whole table reloads.	Success?	✓

Test title:	Edit mode save		
Feature description	In edit mode clicking the save button patches the data in the input fields to the database		
Test description	Edit mode is enabled, data is inputted into the input fields, the save button is pressed and the database gets updated with the new information.		
Expected results	The information in the input fields is Patched to the database and then the table reloads showing the new data		
Test results	The information in the input fields is Patched to the database and then the table reloads showing the new data	Success?	✓

Test title:	Edit mode delete		
Feature description	The delete button deletes that book from the database		
Test description	When in edit mode the delete button is pressed causing the database to delete the book.		
Expected results	The delete button is pressed and the book is removed from the database and then the table is refreshed showing that it's gone		
Test results	The delete button is pressed and the book is removed from the database and then the table is refreshed showing that it's gone	Success?	✓

Test title:	Create book name check success		
Feature description	Input test on the book name input (assuming all other input fields also pass their tests)		
Test description	Something is inputted into the name		
Expected results	This information is sent to the database and the table is refreshed to show this information		
Test results	This information is sent to the database and the table is refreshed to show this information	Success?	✓

Test title:	Create book description check		
Feature description	Input test on the book description input (assuming all other input fields also pass their tests)		
Test description	Something is inputted into the description		
Expected results	This information is sent to the database and the table is refreshed to show this information		
Test results	This information is sent to the database and the table is refreshed to show this information	Success?	✓

Test title:	Create book release date check		
Feature description	Input test on the book release date input (assuming all other input fields also pass their tests)		
Test description	Something is inputted into the release date		
Expected results	This information is sent to the database and the table is refreshed to show this information		
Test results	This information is sent to the database and the table is refreshed to show this information	Success?	✓

Test title:	Create book ISBN check		
Feature description	Input test on the book ISBN input (assuming all other input fields also pass their tests)		
Test description	Something is inputted into the ISBN		
Expected results	This information is sent to the database and the table is refreshed to show this information		
Test results	This information is sent to the database and the table is refreshed to show this information	Success?	✓

Test title:	Create book Genre check fail		
Feature description	Input test on the book Genre input		
Test description	Current selection is please Select should cause error		
Expected results	Alert requesting the user select a Genre, information is not sent to the database		
Test results	Alert requesting the user select a Genre, information is not sent to the database	Success?	✓

Test title:	Create book Genre check success		
Feature description	Input test on the book Genre input (assuming all other input fields also pass their tests)		
Test description	Something is inputted into the Genre		
Expected results	This information is sent to the database and the table is refreshed to show this information		
Test results	This information is sent to the database and the table is refreshed to show this information	Success?	✓

Test title:	Create book Audience check fail		
Feature description	Input test on the book Audience input		
Test description	Current selection is please Select should cause error		
Expected results	Alert requesting the user select a Audience, information is not sent to the database		
Test results	Alert requesting the user select a Audience, information is not sent to the database	Success?	✓

Test title:	Create book Audience check success		
Feature description	Input test on the book Audience input (assuming all other input fields also pass their tests)		
Test description	Something is inputted into the Audience		
Expected results	This information is sent to the database and the table is refreshed to show this information		
Test results	This information is sent to the database and the table is refreshed to show this information	Success?	✓

Test title:	Filter book Name check fail		
Feature description	Input test on the filter Name input		
Test description	Input is blank and the filter by Name button is pressed		
Expected results	Alert requesting the user input a Name, no filter is applied		
Test results	Alert requesting the user select a Name, no filter is applied	Success?	✓

Test title:	Filter book Name check pass		
Feature description	Input test on the filter Name input		
Test description	Input contains something and the filter by Name button is pressed		
Expected results	The table is filtered by the input		
Test results	The table is filtered by the input	Success?	✓

Test title:	Filter book description check fail		
Feature description	Input test on the filter description input		
Test description	Input is blank and the filter by description button is pressed		
Expected results	Alert requesting the user input a Name, no filter is applied		
Test results	Alert requesting the user input a description, no filter is applied	Success?	✓

Test title:	Filter book description check pass		
Feature description	Input test on the filter description input		
Test description	Input contains something and the filter by description button is pressed		
Expected results	The table is filtered by the input		
Test results	The table is filtered by the input	Success?	✓

Test title:	filter book Genre check fail		
Feature description	Input test on the filter Genre input		
Test description	Current selection is please Select should cause error		
Expected results	Alert requesting the user select a Genre, no filter is applied		
Test results	Alert requesting the user select a Genre, no filter is applied	Success?	✓

Test title:	Filter book Genre check success		
Feature description	Input test on the filter Genre input		
Test description	Something is inputted into the Genre		
Expected results	The table is filtered by the input		
Test results	The table is filtered by the input	Success?	✓

Test title:	filter book Audience check fail		
Feature description	Input test on the filter Audience input		
Test description	Current selection is please Select should cause error		
Expected results	Alert requesting the user select an Audience, no filter is applied		
Test results	Alert requesting the user select ab Audience, no filter is applied	Success?	✓

Test title:	Filter book Audience check success		
Feature description	Input test on the filter Audience input		
Test description	Something is inputted into the Audience		
Expected results	The table is filtered by the input		
Test results	The table is filtered by the input	Success?	✓

Test title:	Filter reset		
Feature description	Filter reset button refreshes the table		
Test description	A filter is applied then the filter reset is pressed		
Expected results	The table is reset without any filters after the button is pressed		
Test results	The table is reset without any filters after the button is pressed	Success?	✓

5.0 Conclusion

In conclusion the webpage achieves the organisations' goal of a dynamic webpage that uses AJAX and a REST API. Improvements to the solution would be with the filtering. The filtering is done client side after getting all the data, a more efficient and correct way of doing this would be using AJAX and AXIOS to make a custom request to the database that uses SQL to filter the data before sending it to the client.

6.0 References

- [1] Y. Zaremba, "What Makes a Good QA Documentation (and How to Apply It to Your Own)," IEEE Computer Society, 19 October 2022. [Online]. Available: <https://www.computer.org/publications/tech-news/build-your-career/applications-of-good-qa-documentation>. [Accessed 7 January 2024].

7.0 Figures

Figure 1: Basic webpage design

The image shows a basic webpage design for 'Magic Quill Books'. The page has a dark blue header bar with a 'Create' button on the left and a 'Filter' button on the right. The main content area has a blue background. In the center, there is a table with 7 columns: Name, Description, Release Date, ISBN, Genre, Audience, and Edit. The table has 6 rows of data, each with an 'Edit' button in the last column.

Name	Description	Release Date	ISBN	Genre	Audience	Edit
						Edit
						Edit
						Edit
						Edit
						Edit

Figure 2: After clicking edit

Create

Magic Quill Books

Filter

Name	Description	Release Date	ISBN	Genre	Audience	Edit
						Save Discard Delete
						Edit
						Edit
						Edit
						Edit

Figure 3: Create offcanvas

Create new Book

Name

Description

Release Date

ISBN

Genre

audience

Create

Magic Quill Books

Filter

Description	Release Date	ISBN	Genre	Audience	Edit
					Edit
					Edit
					Edit
					Edit
					Edit

Figure 4: Filter offcanvas

Create

Magic Quill Books

Name	Description	Release Date	ISBN	Genre

✕

Filter

filter

Name

filter

Name

filter

Name

filter

Name

Reset Filter

Cloud Based Web assignment Rhys Hughes

GRADEMARK REPORT

FINAL GRADE

76/100

GENERAL COMMENTS

Assignment 1:

=====

Rest & Ajax (45): 42

Requirements (15): 7.5

Design & Docs (30): 18.5

Demo (10): 7

Overall: **75%**

Assignment 2:

=====

Rest API (40): 30

Patch (10): 9.5

Test doc (30): 23

Testing (10): 7.5

Demo (10): 7

Overall: **77%**

Feedback:

=====

The requirements discussion was a little cursory, I was expecting a list of features/operations required.

A number of "wireframe" diagrams provided, but I was anticipating some discussion and analysis to accompany the design.

Some good evidence of testing. You needed to provide a little discussion of what was required in this documentation.

Your demo video was very blurry, but just about managed to follow along.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16
