Year 3 Server-Side Web Development

Repeat/ Supplemental Assessment 2020

Mode: Individual, Elapsed

Due Date: Friday 22nd May 2020, 22.00

Instructions:

- Follow these instructions and read ALL parts of the assignment carefully
- Include your name and student ID in filename of your upload (e.g. X00000000-FirstnameLastname-CA.zip)
- The node_modules folder must be deleted from your application before zipping
- Upload via Moodle before the deadline.
- You maybe required to present your work.
- University plagiarism rules apply.

Description

You are required to build a web application with CRUD functionality (create, read, update, and delete) for movie or tv series database.

All users should be able to browse the database via a website which should organised by category/ genre. Only admin users should have permission to add, edit, or delete new movies.

User access control:

- Users can view records
- Admin staff can make changes
- Access should not be granted without successful login

User interface/ front end website:

- Show the movie/ tv categories – a category can be selected to show movies in that category.
- Search for movies by name.
- Show the number of movies in each category
- Implement admin functions.

The application should include client and server components with a database storing data:

1. The server should be a RESTful web api built using Nodejs and Express.
2. The client should be built using HTML, CSS, and JavaScript. Use the Fetch api to make calls to the server api endpoints.
3. Use an Azure SQL database for storage

Deliverables

Part 1 (15%)

- Provide a database design
- Define the api endpoints required.
- Define access control for public and admin users.

Include as a document (.docx, .doc. or .odt) in your zip file.

Part 2 (20%)

Using the provided Azure SQL DB, create the required tables, and add sample data.

Azure DB details

Server:	**sswd-db.database.windows.net**
DB:	**sswd-students**

You will be provided with login details separately – check your student email

- Create a user with limited permissions for access via the server app.
- Provide an SQL script to create the database

Part 3 (25%)

Build the server-side Rest API which will allow the data to be accessed.

- Define routes to act as api endpoints (supporting required CRUD functions).
- Use Passport to control access to resources.
- All data should be sent and received as JSON.
- All input data should be validated before use.

Part 4 (25%)

Build a client web application using HTML, CSS, and JavaScript. The application should:

- Run in a single page
- Use Bootstrap 4 for layout and styling
- Display data in an appropriate format
- Allow a user to login, logout.
- Control access to functions by conditionally display UI options based on whether a user is logged in.

Part 5 (15%)

Marks will be awarded for documentation and code quality and descriptive code comments.