DS3103 Webutvikling

Exam

Autumn 2022

Practical information

- This home exam can be solved and delivered alone, in group of 2, or in group of 3. If you work in a group of 2 or 3, you will be expected to code more and have more complexity in code, than if you work alone.
- When you deliver the solution folder in WISEFlow it must first be zipped – you are to deliver 1 main folder containing the React folder, and the Web Api folder. **Note: remove the node_modules** folder from the React folder before uploading to WISEFlow!
- In this exam you work fullstack with ReactJS project, and .NET/C# Web Api project. The .NET/C# Web Api is used by the ReactJS project to do CRUD to the database.
- Assessment guide for internal and external assessors, and for students is at the end of this exam text
- Read through the entire text before starting
- Do not add functionality which requires login with password/username
- The exam is seen as part of the learning process in the course and you can still consult with the teacher and/or with student assistants. If you are uncertain about any point, take contact with the teacher.

Focus areas of this exam

- HTML5 and CSS3, responsive Design with media queries and CSS Grid, CSS Framework
- ReactJS
- .NET/C# Web Api
- Ajax to connect ReactJS and the Web Api
- Universal Design

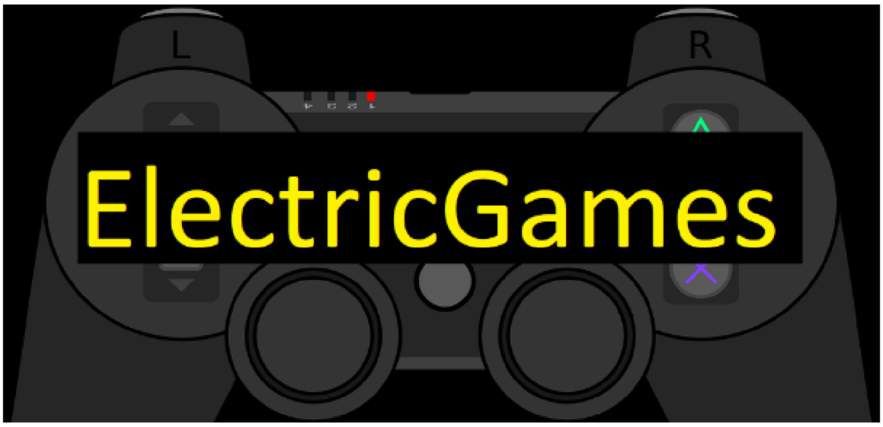
Tip before starting coding

The entire solution will be very dependent on how the information in the database looks. Use some time to plan which properties/attributes the information has, and which functionality you need and want to have in the solution before starting coding. Else you may find yourself having to make changes throughout the solution which may take time. For those working in groups the above mentioned is also important. When one knows how the information looks it is possible to work with frontend and backend at the same time.

Case: ElectricGames

(email from the boss):

A task to you, our fullstack developers!



We have gotten a big gaming company as customer. The company is called ElectricGames and wants to be a new platform for games from different platforms (Xbox, Playstation, Nintendo Switch).

They want a Web Api with a database which has information about Games and GameCharacters. They also want a frontend solution which makes use of the Web Api, and which is both informative and fun (possibly some quiz for example).

A **Game** should at least have the following information:

- Id, Title, Platform (for example Playstation 5), ReleaseYear, Image

A **GameCharacter** should at least have the following:

- Id, Name, Game (which game the character is from), Image

In addition, they also want a HTML page in wwwroot which describes how to use the Web Api: which endpoints are available, how to use the endpoints, which information is available etc.

I'm counting on you for this and know you'll solve this task in a great manner! 😊

Best regards

The Boss

WeAreFullStackDevelopers

07.11.2022

Functionality

The main functionality is based on that the Web Api has methods that do CRUD (Create, Read Update, and Delete) to the Database. The methods in the Web Api are used from the frontend through HTTP requests (GET, POST, PUT, DELETE).

The main functionality in the Web Api is this:

- Get all of something
- Get something by id
- Get something by other property than id, for example GetByName
- Delete something
- Create something (including image upload)
- Update something

The functionality in the frontend makes use of the above methods and in addition not the point about being informative and fun from the email "which is both informative and fun (possibly some quiz for example)."

The amount of implemented functionality and the quality of the solution affects the grade.

Regarding group size and size of solution

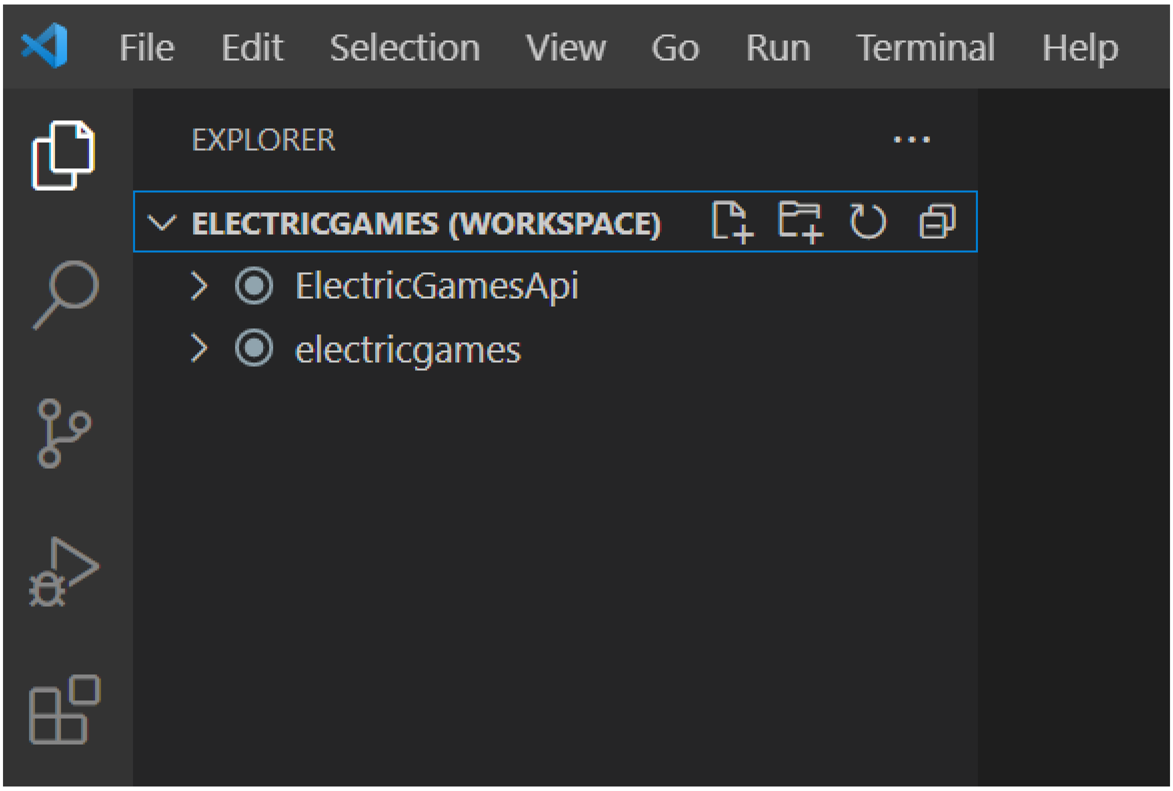
If you work alone, it is enough for you to have only Game or GameCharacter in the database and hence less functionality. If you work 2 you should have both. If you work 3 together you are expected to have both Game and GameCharacter and add a 3rd model class. This is of course not obligatory but will affect the grade.

Report Universal Design (UD, Norwegian: Universell Utforming)

Answer the questions below in your own words (no copy paste). Include print screen where you find it useful. Save the file in pdf format and include it inside the folder containing the rest of the solution.

1. What is Universal Design about in context of a web page?
2. What is contrast about in UD?
3. Test the contrast of a button with a contrast test tool such as this, and include a print screen of the result to include in the report: https://www.tpgi.com/color-contrast-checker/
4. How should a <img> be made according to UD?
5. Which role does semantic coding play in UD?

The print screen below shows how it may look in Visual Studio Code when you have both the Web Api project and the React project opened. These are two separate project folders. Both folders are put into a common folder, zipped, and uploaded to WISEFlow.



Assessment guide exam DS3103 Webutvikling

For students, internal assessors, and external assessors.

The table below gives an indication of how the exam will be assessed. Since the solution can be solved in many different manners and focus on different things the percentages can't/shouldn't be exact.

There is much more variation in what can be done in the frontend part than what can be done in the backend part.

Frontend (around 60%)	**Comment**
Component-based development in React JS, programming with JavaScript as taught in the course.	
React Routing	
HTTP requests to Web Api for CRUD	
Responsive design with CSS, CSS Grid, media queries, Bootstrap CSS framework.	For the grid one can choose CSS3 grid or the grid in the CSS framework.
Context	
Backend (around 40%)	
Use of interfaces and classes	
Controllers with CRUD methods to Database.	
Picture upload to the images folder in wwwroot	
wwwroot	This is the part referred to in the case which describes the Web Api. See page 2.
General things (included in the 100% above)	
Amount of code and complexity. Also, variation of code counts as something positive.	If in group more is expected.
Structured, tidy, good code (for example, not having unnecessary code repetition), modularization (for example breaking up code into functions/modules etc. especially in the JavaScript)	
Report Universal Design (weight of around 5%)	