

# idg2100-2024-oblig3

This document contains the description and starter code for `oblig3: IDG2100 Spring 2024`.

## Goal

Prove and demonstrate:

- your understanding about `React` library and how to use it to create `SPAs` using [React Router v6](#);
- you can build react components and understand the `component lifecycle`;
- you understand the difference between `stateless`, `stateful` and `controlled` components;
- your ability to connect the front-end to the back-end;
- your understanding of backend authentication/authorisation processes by protecting routes using JWT
- your understanding of security

## Context

In this "oblig" you will be presented with a brief description of the problem you have to solve using what you have learnt so far.

This is a **group assignment** that involves three students. Groups are self-assigned, and you are responsible for finding and joining one. Please be proactive and ensure you are part of a group as soon as possible. Your task is to implement the project entirely from scratch.

However, you may incorporate code fragments from your previous mandatory assignments (obligs) if necessary. If you do use code from previous assignments, be sure to cite the code clearly in the comments and identify the student who created it and the "oblig" in which it was created.

This assignment has two parts: **a coding component** and **an on campus oral presentation**. You must submit your code via both GitHub and Blackboard (please refer to the [Delivery](#)

section for specific instructions). The details regarding the dates and requirements for the physical oral presentation will be posted on Blackboard. Not attending the physical oral presentation will result in a non-passing grade.

Although, you may utilise snippets of code from tutorials or official documentation, you must clearly acknowledge the sources in the comments of your code. Plagiarism or cheating will be deemed to have taken place if the submitted code shows substantial similarities to other students' assignments or projects found online. In such cases, the matter will be reported to the NTNU appeals committee for further examination. If you have any doubts regarding the use of materials for your project, please reach out to the instructor for clarification.

If the assignment is graded as "not approved" you will have an additional opportunity based on the following conditions:

1. The first version of the project must have been delivered within the set deadline (never after);
2. The project must consist on a significant piece of work (i.e.: do not deliver an empty assignment);

For the second attempt you will receive a very short deadline to fix your project.

# Brief: The SUPER Assessor - Administrator Interface v1.0.0

## Scenario

Following the successful development of the SUPER Assessor's front-end components and the back-end API, the next step is to create a secure, intuitive interface for administrators. This interface will allow administrators to manage the card and user collections efficiently, ensuring that only authorized personnel can make changes to these critical components of the platform. The interface will serve as a crucial tool for maintaining the integrity and quality of the content available to educators and students.

## Key Features

- **Welcome page:** A page that shows an intro to the game and showcases some of

the cards.

- **Login Page:** A secure login mechanism for administrators to access the dashboard.
- **Dashboard Overview:** Once logged in, the dashboard provides an overview of the platform's statistics, including the total number of cards (and types) and total number of users.
- **Cards Management:** A section dedicated to managing the Mission and Assessment Cards. Administrators can create, read, update, and delete (CRUD) cards.
  - Admins (only) can perform CRUD operations on cards.
  - Admins can create cards in bulk by uploading a JSON file.
  - Admins can edit the icon of a specific card or change the default icon for a specific set of cards within the same category. If the icon hasn't been edited, the default icon for the card category will be used. Otherwise, the card will display a custom icon.
  - Admins will also have the ability to select cards in bulk and generate a PDF file with the visual representation of the selected cards (both sides - ) using the web component from assignment 1. The pdf file will have a two column layout where the first column will show the front of the card and the left column the back side. See [Custom HTML elements with React](#) and [Web components with React \(old documentation\)](#).
- **Users Management:** A section for managing user accounts. Administrators can view user details (except passwords), delete users, and modify user roles or permissions (⚠ *in this version users will not have access to the user interface but keep in mind that future versions may add that functionality. Therefore, develop the system having that in mind*).
- **Protected Routes:** Ensure that the CRUD operations and the administrator dashboard are only accessible to users authenticated as administrators (both front-end and back-end).
- **Logout Functionality:** Enable administrators to securely log out from the dashboard.

# Task

- Understand the requirements: Review and understand the project requirements to ensure you have a clear understanding of what needs to be built.
- Research how to implement protected routes using `React router v6` : Investigate how to implement protected routes using React Router v6 to ensure the dashboard is secure and accessible only to authenticated users.
- Research secure ways of implementing authentication/authorization in the backend: Focus on JWT, refresh tokens, and HTTP only cookies, to ensure that the dashboard is secure.
- Define the User Interface: Based on the requirements, define the user interface of the dashboard. Consider the design and layout, the types of components to be used, and the flow of the dashboard. Consider creating wireframes in this step.
- Decide what version of the API you will use for this assignment and identify if you will need new routes or endpoints to extend the functionalities of the existing API: Identify if you will need to create new routes or endpoints to extend the functionalities of the existing API.
- Build the functional Prototype: Build a functional prototype of the dashboard using React components. Start with the landing page and continue to build out the other pages. Ensure that the design is responsive, user-friendly, and meets the requirements of the project. Ensure that all components work as expected, and the dashboard is fully functional.
- Write a good readme file explaining how to install and deploy the application in a local machine: Include a script in the backend to seed the database with some dummy data for testing purposes.
- Prepare a `POSTMAN` collections to test the `APIs` .

Note: this assignment must include the code of the front-end and the back-end.

# Delivery

This assignment must be delivered in two different places: GitHub classroom and Blackboard.

- To deliver the assignment in GitHub Classroom, you only need to make sure all

your changes and commits are pushed to your Git repository.

- A Pull request is created automatically when the repository is cloned. Feedback will be included there if needed. Do not remove or close that Pull Request.
- Only the changes in the "main" branch will be considered for giving feedback or grading the assignment.
- It is imperative that you work exclusively with this Git repository to ensure that all modifications are trackable and your code is backed up on a regular basis. Hence, you should commit your progress directly to this repository each time you make advancements.
- Before delivering the assignment in Blackboard, make sure your project has all the files it needs. Delete any file, folder or info that is not needed (this is `.git/` folder, `node_modules`, etc.). Zip the project and upload the file to Blackboard.
- Don't forget to add/update all the `API` specs in `documentation` and your query collection in the `postman` folder, if needed.
- Remember you will have to present your project orally and on campus (date posted on BB).

# Useful resources

You can find additional resources in

## Front-end

- [React Router 6: Private Routes \(alias Protected Routes\)](#)
- [Creating Protected Routes With React Router V6](#)
- [\(video\) React Protected Routes | Role-Based Authorization | React Router v6](#)
- [\(Video\) The New Way To Create Protected Routes With React Router V6](#)

## Back-end

- [Node.js + MongoDB API - JWT Authentication with Refresh Tokens](#)
- [React 18 Authentication with Node.js JWT API](#)
- [React + Axios - Add Bearer Token Authorization Header to HTTP Request](#)