# Testing environment/rules Grade 100/100

* For this test, you may use your notes, textbook, reading materials on Perusall, your completed lab assignments, and the internet as a resource for this Hands-On test.
* This test is being administered as a **take-home test,** and is due back Monday 12/04/23 at 8:00AM
* The test must be submitted via a GitHub repository, and you must give collaborator/admin access to your instructor.
* This is an individual project, you are expected to complete all work by yourself without collaboration/input from other students.
* All code must be your own original work. Code found online, on websites like Stack Overflow, Medium, etc. is strictly prohibited. *(Example code found on* [*developer.mozilla.org*](https://developer.mozilla.org) *is permitted though.)*

# How to submit your code

As we have done with previousassignments, you need to create a new GitHub repository for this project.

Details are listed below:

* Name the repository **"awd1111-exam-5"**
* Make the repository **"private"**
* Add **evangudmestad** as a **Collaborator/Admin**

# Initialize the project

1. Clone the repository to your computer.
2. Open the repository folder in Visual Studio Code.
3. Create a React.js application

# Remove unused example code

**Delete** the following css files and **remove all import statements** for them:

* index.css
* app.css

Replace the **App.js** with the following code:

|  |
| --- |
| function App() {  return <div>Hello World!</div>; }  export default App; |

# Install dependencies

Install the following dependencies:

* **bootstrap**
* **lodash**
* **moment**
* **react-router-dom**
* **react-toastify**

# Project Overview

* Build a small 3-page application using React.
* Use React Router to navigate between pages.
* Use React Toastify to show toast messages.
* Each of the 3 pages will include a small form as described below.

# index.jsx (5pts)

Implement Bootstrap via the local dependency **(1pt)**

Implement React Router via the local dependency **(4pt)**

# Navbar.jsx (10pts)

Create a **Navbar** component from the following starter code:

|  |
| --- |
| import { NavLink } from 'react-router-dom';  function Navbar() {  return (  <header>  <nav>  <ul>  <li><NavLink to="/page1">Page 1</NavLink></li>  <li><NavLink to="/page2">Page 2</NavLink></li>  <li><NavLink to="/page3">Page 3</NavLink></li>  </ul>  </nav>  </header>  ); }  export default Navbar; |

*Continued on the next page.*

Use the following Bootstrap classes on the **<header>** element: **(2pt)**

* **navbar**
* **navbar-expand** so that the navbar is always horizontal
* **navbar-light** or **navbar-dark** depending on the desired color scheme
* **bg-light,** or **bg-dark,** or **bg-primary,** or **bg-secondary** depending on the desired color scheme (do not use bg-success/bg-danger/bg-warning/bg-info as these have semantic meanings)
* **sticky-top** to keep the navbar visible at all times

Use the following Bootstrap classes on the **<nav>** element: **(1pt)**

* **container** or **container-fluid** depending on desired width of the navbar

Use the following Bootstrap classes on the **<ul>** element: **(1pt)**

* **navbar-nav**

**Do the following for each link in the navbar: (5pt)**

* All **<li>** elements must have the **nav-item** class.
* All **<NavLink>** elements must have the **nav-link** class.
* Update the path and text for each link. *(See instructions on the following pages.)*
* To support **react-router,** use **NavLink** elements instead of **<a>** elements in the navbar.

|  |
| --- |
| <NavLink className="nav-link" to="/login">Login</NavLink> |

When all the previous steps are completed, add the Navbar component to the layout in **App.js (1pt)**

*Continued on the next page.*

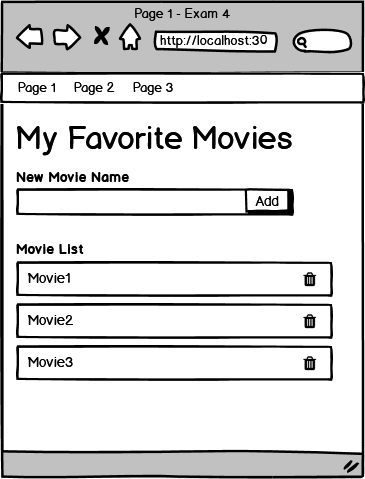
# Page1.jsx (25pts)

Create a page that stores and displays the user's list of favorite movies.

* This page must be accessible via the route **"/page1"**
* Store the list of favorite movies in a state variable named **"items"**
* Initialize the movie list with **3 of your personal favorite movies,** within the **useState()** hook as shown below:  **(5pts)**

|  |
| --- |
| const [ items, setItems ] = useState([ 'Movie1', 'Movie2', 'Movie3' ]); |

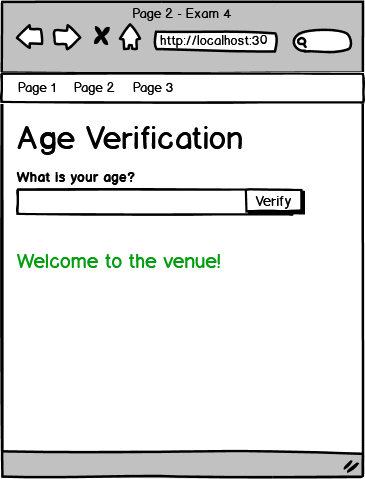
* This list of movies must be displayed on screen at all times. Use the [**\_.map()**](https://lodash.com/docs/4.17.15#map) function to transform the movie list into HTML elements. **(5pts)**
* Ask the user to enter their favorite movie into a text box. Have them click a button to add the movie to the existing array. **(5pts)**
  + If the user leaves the field blank, display an error message instead of adding the movie to the array.
  + If the movie is added, display a toast message informing the user that it has been added.
* Add a delete button to each movie, so that the user can remove movies from their list. **(5pts)**
* Use **Bootstrap v5** classes to style the whole page. **(5pts)**



# Page2.jsx (25pts)

Create a page that implements a simple age verification form.

* This page must be accessible via the route **"/page2"**
* Ask the user to enter their **age** in a textbox and click a button to verify their age.
* The <input> field must have **type="text"** to fully test all possibilities.
* If the user is **21 or older,** display a message saying **"Welcome to the venue!" (5pts)**
* If the user is **under 21,** display a message saying **"You're not old enough!" (5pts)**
* If the user does not enter an **integer,** display a message saying **"Please enter your age!"** *All of the following are not integers: "17.5", ".5", "-", and "". You must handle all of these cases.* **(5pts)**
* If the user enters **an age below 1 or above 200,** display a message saying **"Age out of range!" (5pts)**
* Use **Bootstrap v5** classes to style the whole page. **(5pts)**



# Page3.jsx (35pts)

Create a simple review page for a single product/service.

* This page must be accessible via the route **"/page3"**
* Use **any real world or fictional product/service** of your choice. **(2pts)**
  + Include a **title** and **description** of the product.
  + Create 3 fake reviews for the product that will be displayed initially.
* Provide a way to view previous reviews and ratings. **(5pts)**
* Use [**\_.meanBy()**](https://lodash.com/docs/4.17.15#meanBy) to calculate the average rating of all reviews. And display it as shown. **(2pts)**
* Allow users to add their own reviews. Each review must include the following information: **(15pts)**
  + Author's Name
    - *Entered into a text input*
  + Review Text
    - *Entered into a text area*
  + 5-star rating (in the range [0, 5] inclusive)
    - *Use radio buttons to select the rating*
  + Date & Time of posting
    - *Use the current date and time*
    - *Do not include date/time fields on the form*
* Display validation errors, if a field is left blank. **(2pts)**
* When the user submits a review, clear out the review form. **(2pts)**
* When the user submits a review, display the new review and update the average rating. **(2pts)**
* Users can rate the product an unlimited number of times.
* Use **Bootstrap v5** classes to style the whole page. **(5pts)**