**Student Name: Weight: 15%**

**Student ID:** **Marks:** **/100**

# Assignment: Exploring Front-End Development Components

## Scenario

New Generation High School is a new school that wants to create a student portal. The administrator of the web application wants to be able to see a list of students and modify the student list to add new students.

## Equipment and Materials

For this assignment, you will need:

* Node.js
* Next.js

## Instructions

1. Working in groups, as indicated by your instructor, review the scenario and the *Marking Criteria* section below.
2. Install Node.js.
3. Create a new Next.js project.
4. Create a list of students in JSON format that contains data for five students.
5. Create a single page web application that includes the following:

* A custom static navbar
* A list of students
* A footer that includes the school information
* Student information that contains:
* Student’s first name and last name
* Student’s date of birth
* Student’s current grade
  + A form to add new students

**Notes:**

* The components must contain data validation.
* You must use custom components for each of the elements mentioned above. Feel free to create any other components as necessary. Choose whether a component will be a client and server component.

1. Deploy the website.
2. Submit the following to Brightspace by the posted due date:

* A link to your GitHub repository
* A link to your deployed website

**Note:** Only one submission is required per group.

## Marking Criteria

| **Criteria** | **Missing** | **Needs Improvement** | **Good** | **Excellent** | **Marks** |
| --- | --- | --- | --- | --- | --- |
| **Working Code** | * Nothing submitted.   (0 marks) | * Single web application missing. * Most of the React components missing.   (10 marks) | * Most components are implemented. * Hooks don’t work properly. * Bugs exist in the code. * User can’t add items to the list of students or adding to the list causes errors.   (30 marks) | * All components are implemented. * No bugs exist in the code. * User can add entries to the student list without errors.   (40 marks) | /40 |
| **Style** | * Nothing submitted.   (0 marks) | * Indentation: inconsistent * Readability: poor variable names * Documentation: * No comments are included at the top. * No comments indicating major code sections or what they do.   (5 marks) | * Indentation: some inconsistencies * Readability: some variable names are not ideal * Documentation: * Comments at the top are missing or incomplete. * Comments indicating major code sections and what they do are incomplete.   (15 marks) | * Indentation: consistent * Readability: good variable names * Documentation: * Comments at the top are complete and include name, date and program description, including inputs, processing and output details  (4–5 sentences minimum). * Comments indicate major code sections and what they do.   (20 marks) | /20 |
| **Testing** | * Nothing submitted.   (0 marks) | * Data are not validated when user adds a new entry. * User cannot add new entries.   (5 marks) | * Data are partially validated when user adds new entries. * User can add new entries, but errors occur.   (15 marks) | * Data are validated when user adds new entries. * User can add new entries without errors. * Data validation is implemented.   (20 marks) | /20 |
| **Version Control (Group Work)** | * Nothing submitted. * No GitHub repository created.   (0 marks) | * Not all group members commit codes to GitHub.   (5 marks) | * GitHub repository is incomplete. * Git commit messages are incomplete.   (15 marks) | * GitHub repository is created. * All group members committed at least once.   (20 marks) | /20 |
| **Total** | | | | | **/100** |