## Project 2: React CRUD Assessment Rubric

	10-9	8-7	6-5	4-0
Functionality	Applications demonstrate comprehensive & robust evidence on the following:  Register a new user via a form.  User can login and logout.  API data requested from API resource groups using Axios.  Create, update & delete API data.  View API data in a table.  Incorrectly formatted form fields are handled using validation error messages.  API data paginated across several pages.  UI styled with Reactstrap.  Application deployed to Heroku.  Register, login and logout functionality tested using Cypress.	Applications demonstrate clear & detailed evidence on the following:  Register a new user via a form.  User can login and logout.  API data requested from API resource groups using Axios.  Create, update & delete API data.  View API data in a table.  Incorrectly formatted form fields are handled using validation error messages.  API data paginated across several pages.  UI styled with Reactstrap.  Application deployed to Heroku.  Register, login and logout functionality tested using Cypress.	Applications demonstrate evidence on the following:  Register a new user via a form. User can login and logout. API data requested from API resource groups using Axios. Create, update & delete API data. View API data in a table. Incorrectly formatted form fields are handled using validation error messages. API data paginated across several pages. UI styled with Reactstrap. Application deployed to Heroku. Register, login and logout functionality tested using Cypress.	Applications does not, or does not fully demonstrate evidence on the following:  Register a new user via a form.  User can login and logout.  API data requested from API resource groups using Axios.  Create, update & delete API data.  View API data in a table.  Incorrectly formatted form fields are handled using validation error messages.  API data paginated across several pages.  UI styled with Reactstrap.  Application deployed to Heroku.  Register, login and logout functionality tested using Cypress.

				T
	Applications thoroughly demonstrate code	Applications clearly demonstrate code	Applications demonstrate code elegance	Applications does not or does not fully
	elegance on the following:	elegance on the following:	on the following:	demonstrate code elegance on the
	<ul> <li>Appropriate use of control flow,</li> </ul>	<ul> <li>Appropriate use of control flow,</li> </ul>	<ul> <li>Appropriate use of control flow,</li> </ul>	following:
	data structures and in-built	data structures and in-built	data structures and in-built	<ul> <li>Appropriate use of control flow,</li> </ul>
	functions.	functions.	functions.	data structures and in-built
Elegance	<ul> <li>Sufficient code modularity.</li> </ul>	<ul> <li>Sufficient code modularity.</li> </ul>	<ul> <li>Sufficient code modularity.</li> </ul>	functions.
	<ul> <li>Components written as</li> </ul>	<ul> <li>Components written as</li> </ul>	<ul> <li>Components written as</li> </ul>	<ul> <li>Sufficient code modularity.</li> </ul>
	functional, not class.	functional, not class.	functional, not class.	Components written as
	<ul> <li>Adheres to client-server</li> </ul>	<ul> <li>Adheres to client-server</li> </ul>	<ul> <li>Adheres to client-server</li> </ul>	functional, not class.
	architecture.	architecture.	architecture.	Adheres to client-server
e P	<ul> <li>Header &amp; in-line comments</li> </ul>	Header & in-line comments	<ul> <li>Header &amp; in-line comments</li> </ul>	architecture.
Code	explain complex logic.	explain complex logic.	explain complex logic.	Header & in-line comments
	<ul> <li>Formatted code using Prettier &amp;</li> </ul>	<ul> <li>Formatted code using Prettier &amp;</li> </ul>	<ul> <li>Formatted code using Prettier &amp;</li> </ul>	explain complex logic.
	npm script.	npm script.	npm script.	<ul> <li>Formatted code using Prettier &amp;</li> </ul>
	Cypress & Prettier are installed	Cypress & Prettier are installed	Cypress & Prettier are installed	npm script.
	as dev dependencies.	as dev dependencies.	as dev dependencies.	Cypress & Prettier are installed
	No dead or unused code.	No dead or unused code.	No dead or unused code.	as dev dependencies.
				No dead or unused code.
	README file contains thorough evidence	README file contains clear evidence of:	README file contains evidence of:	README file does not or does not fully
e e	of:	<ul> <li>URL to application on Heroku.</li> </ul>	<ul> <li>URL to application on Heroku.</li> </ul>	contain evidence of:
Sag	<ul> <li>URL to application on Heroku.</li> </ul>	<ul> <li>How to setup the environment</li> </ul>	<ul> <li>How to setup the environment</li> </ul>	<ul> <li>URL to application on Heroku.</li> </ul>
Ď	<ul> <li>How to setup the environment</li> </ul>	for development, run Cypress	for development, run Cypress	<ul> <li>How to setup the environment</li> </ul>
Git	for development, run Cypress	tests & deploy the application.	tests & deploy the application.	for development, run Cypress
	tests & deploy the application.			tests & deploy the application.
જ		Git branches are mostly named with	Some git branches are named with	
ō	Git branches are thoroughly named with	convention & contain the correct code	convention & contain the correct code	Git branches are not or are not fully
ati	convention & contain the correct code	relating to the functional requirement.	relating to the functional requirement.	named with convention & do not or do not
r	relating to the functional requirement.			fully contain the correct code relating to
Documentation		Git commit messages are clearly	Git commit messages are formatted &	the functional requirement.
5	Git commit messages are comprehensively	formatted & reflect the functionality	reflect the functionality changes in detail.	
8	formatted & reflect the functionality	changes in substantial detail.		Git commit messages are not or are not
Ω	changes in succinct detail.			fully formatted & do not or do not reflect
				the functionality changes.

## Project 2: React CRUD App Marking Cover Sheet

Name:							
Date:							
Learner ID:							
Assessor's Name:							
Assessor's Signature:							
Criteria	Out Of	Weighting	Final Result				
Functionality	10	40					
Code Elegance	10	45					
Documentation & Git Usage	10	15					
Final Result		/100					
This assessment is worth 50% of the final mark for the Introductory Application Development							
Concepts course.							
Feedback:							
Functionality:							
Code Elegance:							

Documentation & Git Usage: