

## Project 2: React CRUD Assessment Rubric

	10-9	8-7	6-5	4-0
Functionality	<p>Application demonstrates comprehensive &amp; robust evidence on the following:</p> <ul style="list-style-type: none"> <li>• Register, login &amp; logout.</li> <li>• Requesting REST API data using Axios.</li> <li>• View REST API data.</li> <li>• Update &amp; delete REST API data.</li> <li>• Incorrectly formatted form fields are handled.</li> <li>• Paginate &amp; search REST API data.</li> <li>• UI designed with Reactstrap.</li> <li>• Deployed to Heroku.</li> <li>• End-to-end that ensures authentication is working as expected.</li> </ul>	<p>Application demonstrates clear &amp; detailed evidence on the following:</p> <ul style="list-style-type: none"> <li>• Register, login &amp; logout.</li> <li>• Requesting REST API data using Axios.</li> <li>• View REST API data.</li> <li>• Update &amp; delete REST API data.</li> <li>• Incorrectly formatted form fields are handled.</li> <li>• Paginate &amp; search REST API data.</li> <li>• UI designed with Reactstrap.</li> <li>• Deployed to Heroku.</li> <li>• End-to-end that ensures authentication is working as expected.</li> </ul>	<p>Application demonstrates evidence on the following:</p> <ul style="list-style-type: none"> <li>• Register, login &amp; logout.</li> <li>• Requesting REST API data using Axios.</li> <li>• View REST API data.</li> <li>• Update &amp; delete REST API data.</li> <li>• Incorrectly formatted form fields are handled.</li> <li>• Paginate &amp; search REST API data.</li> <li>• UI designed with Reactstrap.</li> <li>• Deployed to Heroku.</li> <li>• End-to-end that ensures authentication is working as expected.</li> </ul>	<p>Application demonstrates does not, or does not fully demonstrate evidence on the following:</p> <ul style="list-style-type: none"> <li>• Register, login &amp; logout.</li> <li>• Requesting REST API data using Axios.</li> <li>• View REST API data.</li> <li>• Update &amp; delete REST API data.</li> <li>• Incorrectly formatted form fields are handled.</li> <li>• Paginate &amp; search REST API data.</li> <li>• UI designed with Reactstrap.</li> <li>• Deployed to Heroku.</li> <li>• End-to-end that ensures authentication is working as expected.</li> </ul>
Code Elegance	<p>Application thoroughly demonstrate code elegance on the following:</p> <ul style="list-style-type: none"> <li>• Intermediate variables, idiomatic control flow, data structures &amp; in-built functions, &amp; sufficient modularity.</li> <li>• Functions &amp; variables are named appropriately.</li> <li>• Components are written as functional.</li> <li>• Adheres to a client-server architecture.</li> <li>• Filer header &amp; in-line comments.</li> <li>• Formatted code using Prettier.</li> <li>• Prettier &amp; Cypress are installed as dev dependencies.</li> <li>• No dead or unused code.</li> </ul>	<p>Application clearly demonstrate code elegance on the following:</p> <ul style="list-style-type: none"> <li>• Intermediate variables, idiomatic control flow, data structures &amp; in-built functions, &amp; sufficient modularity.</li> <li>• Functions &amp; variables are named appropriately.</li> <li>• Components are written as functional.</li> <li>• Adheres to a client-server architecture.</li> <li>• Filer header &amp; in-line comments.</li> <li>• Formatted code using Prettier.</li> <li>• Prettier &amp; Cypress are installed as dev dependencies.</li> <li>• No dead or unused code.</li> </ul>	<p>Application demonstrate code elegance on the following:</p> <ul style="list-style-type: none"> <li>• Intermediate variables, idiomatic control flow, data structures &amp; in-built functions, &amp; sufficient modularity.</li> <li>• Functions &amp; variables are named appropriately.</li> <li>• Components are written as functional.</li> <li>• Adheres to a client-server architecture.</li> <li>• Filer header &amp; in-line comments.</li> <li>• Formatted code using Prettier.</li> <li>• Prettier &amp; Cypress are installed as dev dependencies.</li> <li>• No dead or unused code.</li> </ul>	<p>Application does not or does not fully demonstrate code elegance on the following:</p> <ul style="list-style-type: none"> <li>• Intermediate variables, idiomatic control flow, data structures &amp; in-built functions, &amp; sufficient modularity.</li> <li>• Functions &amp; variables are named appropriately.</li> <li>• Components are written as functional.</li> <li>• Adheres to a client-server architecture.</li> <li>• Filer header &amp; in-line comments.</li> <li>• Formatted code using Prettier.</li> <li>• Prettier &amp; Cypress are installed as dev dependencies.</li> <li>• No dead or unused code.</li> </ul>

Documentation & Git Usage	<p>README file contains thorough evidence of:</p> <ul style="list-style-type: none"> <li>• URL to the application on Heroku.</li> <li>• How to setup the environment for development, run end-to-end Cypress tests &amp; deploy the application.</li> </ul> <p>Git branches are thoroughly named with convention &amp; contain the correct code relating to the functional requirement.</p> <p>Git commit messages are comprehensively formatted &amp; reflect the functionality changes in succinct detail.</p>	<p>README file contains clear evidence of:</p> <ul style="list-style-type: none"> <li>• URL to the application on Heroku.</li> <li>• How to setup the environment for development, run end-to-end Cypress tests &amp; deploy the application.</li> </ul> <p>Git branches are mostly named with convention &amp; contain the correct code relating to the functional requirement.</p> <p>Git commit messages are clearly formatted &amp; reflect the functionality changes in substantial detail.</p>	<p>README file contains evidence of:</p> <ul style="list-style-type: none"> <li>• URL to the application on Heroku.</li> <li>• How to setup the environment for development, run end-to-end Cypress tests &amp; deploy the application.</li> </ul> <p>Some git branches are named with convention &amp; contain the correct code relating to the functional requirement.</p> <p>Git commit messages are formatted &amp; reflect the functionality changes in detail.</p>	<p>README file does not or does not fully contain evidence of:</p> <ul style="list-style-type: none"> <li>• URL to the application on Heroku.</li> <li>• How to setup the environment for development, run end-to-end Cypress tests &amp; deploy the application.</li> </ul> <p>Git branches are not or are not fully named with convention &amp; do not or do not fully contain the correct code relating to the functional requirement.</p> <p>Git commit messages are not or are not fully formatted &amp; do not or do not reflect the functionality changes.</p>
---------------------------	--	--	--	---

# 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:**