

ID607001: Introductory Application Development Concepts

Project 2: React CRUD Assessment Rubric

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

Documentation & Git Usage	<p>Thorough use of project board & .gitignore.</p> <p>README file contains thorough evidence of:</p> <ul style="list-style-type: none"> • URL to the application on Heroku. • How to setup the environment for development, run UI tests & deploy the application. <p>Comprehensive use of Markdown syntax, i.e., headings, bold text & code blocks.</p> <p>Thorough spelling & grammar correctness.</p> <p>Git commit messages are comprehensively formatted & reflect the functionality changes in succinct detail.</p>	<p>Clear use of project board & .gitignore.</p> <p>README file contains clear evidence of:</p> <ul style="list-style-type: none"> • URL to the application on Heroku. • How to setup the environment for development, run UI tests & deploy the application. <p>Substantial use of Markdown syntax, i.e., headings, bold text & code blocks.</p> <p>Clear spelling & grammar correctness.</p> <p>Git commit messages are clearly formatted & reflect the functionality changes in substantial detail.</p>	<p>Use of project board & .gitignore.</p> <p>README file contains evidence of:</p> <ul style="list-style-type: none"> • URL to the application on Heroku. • How to setup the environment for development, run UI tests & deploy the application. <p>Use of Markdown syntax, i.e., headings, bold text & code blocks.</p> <p>Spelling & grammar correctness.</p> <p>Git commit messages are formatted & reflect the functionality changes in detail.</p>	<p>Does not or does not fully demonstrate use of project board & .gitignore.</p> <p>README file does not or does not fully contain evidence of:</p> <ul style="list-style-type: none"> • URL to the application on Heroku. • How to setup the environment for development, run UI tests & deploy the application. <p>Does not or does not fully demonstrate use of Markdown syntax, i.e., headings, bold text & code blocks.</p> <p>Does not or does fully demonstrate spelling & grammar correctness.</p> <p>Git commit messages are not or are not fully formatted & do not or do not reflect the functionality changes.</p>

ID607001: Introductory Application Development Concepts

Project 2: React CRUD 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: