ID607001: Introductory Application Development Concepts

Project 2: React CRUD Assessment Rubric

	10-9	8-7	6-5	4-0
Functionality	Application demonstrates comprehensive & robust evidence on the following: 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 & search REST API data. UI designed with Reactstrap. Deployed to Heroku. End-to-end that ensures authentication is working as expected.	Application demonstrates clear & detailed evidence on the following: 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 & search REST API data. UI designed with Reactstrap. Deployed to Heroku. End-to-end that ensures authentication is working as expected.	Application demonstrates evidence on the following: 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 & search REST API data. UI designed with Reactstrap. Deployed to Heroku. End-to-end that ensures authentication is working as expected.	Application demonstrates does not, or does not fully demonstrate evidence on the following: 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 & search REST API data. Ul designed with Reactstrap. Deployed to Heroku. End-to-end that ensures authentication is working as expected.

ID607001: Introductory Application Development Concepts

Project 2: React CRUD

Version 3, Semester One, 2022

elegance on the following: Intermediate variables, idiomatic control flow, data structures & inbuilt functions, & sufficient modularity. Functions & variables are named appropriately. Components are written as functional. Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. README file contains thorough evidence of: URL to the application on Heroku. How to setup the environment for development, run end-to-end elegance on the following: Intermediate variables, idiomatic control flow, data structures & inbuilt functions, & sufficient modularity. Intermediate variables, idiomatic control flow, data structures & inbuilt functions, & sufficient modularity. Intermediate variables, idiomatic control flow, data structures & inbuilt functions, & sufficient modularity. Intermediate variables, idiomatic control flow, data structures & inbuilt functions, & sufficient modularity. Functions & variables are named appropriately. Components are written as functional. Adheres to a client-server architecture. Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. No dead or unused code. README file contains thorough evidence of: URL to the application on Heroku. How to setup the environment for development, run end-to-end elegance on the following: Intermediate variables, idiomatic control flow, data structures & inbuilt functions, & sufficient modularity. Functions & variables are named appropriately. Components are written as functional. Functions & variables are named appropriately. Functions & variables are named appropriately. Functions & variables are installed as functional. Functions & variables are installed as functional. Functions & variables are installed as functional. No dead or unused code wit	s not or does not fully de elegance on the following: mediate variables, idiomatic ol flow, data structures & in-
Intermediate variables, idiomatic control flow, data structures & inbuilt functions, & sufficient modularity. Functions & variables are named appropriately. Components are written as functional. Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. README file contains thorough evidence of: URL to the application on Heroku. Intermediate variables, idiomatic control flow, data structures & in-built functions, & sufficient modularity. Intermediate variables, idiomatic control flow, data structures & in-built functions, & sufficient modularity. 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. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. README file contains thorough evidence of: URL to the application on Heroku. Intermediate variables, idiomatic control flow, data structures & in-built functions, & sufficient modularity. Functions & variables are named appropriately. Formatic ap	mediate variables, idiomatic ol flow, data structures & in-
control flow, data structures & inbuilt functions, & sufficient modularity. • Functions & variables are named appropriately. • Components are written as functional. • Adheres to a client-server architecture. • Filer header & in-line comments. • Formatted code using Prettier. • Prettier & Cypress are installed as dev dependencies. • No dead or unused code. README file contains thorough evidence of: • URL to the application on Heroku. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow, data structures & inbuilt functions, & sufficient modularity. control flow description for the built functions, & sufficient modularity. control flow description for the built functions, & sufficient modularity. control flow description on Heroku.	ol flow, data structures & in-
built functions, & sufficient modularity. Functions & variables are named appropriately. Components are written as functional. Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. README file contains thorough evidence of: URL to the application on Heroku. built functions, & sufficient modularity. Functions & variables are named appropriately. Components are written as functional. Adheres to a client-server architecture. Filer header & in-line comments. Filer header & in-line commen	•
modularity. Functions & variables are named appropriately. Components are written as functional. Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. Modularity. modularity. prod appropriately. Functions & variables are named appropriately. Adheres to a client-server architecture. Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. Modead or unused	
functional. Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. README file contains thorough evidence of: URL to the application on Heroku. Adheres to a client-server architecture. Adheres to a client-server architecture. Filer header & in-line comments. Filer header & i	functions, & sufficient
 Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. FEADME file contains thorough evidence of: URL to the application on Heroku. Adheres to a client-server architecture. Adheres to a client-server architecture. Filer header & in-line comments. Filer header & cypress are installed as dev dependencies. No dead or unused code. 	ılarity.
 Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. FEADME file contains thorough evidence of: URL to the application on Heroku. Adheres to a client-server architecture. Adheres to a client-server architecture. Filer header & in-line comments. Filer header & cypress are installed as dev dependencies. No dead or unused code. 	tions & variables are named
 Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. FEADME file contains thorough evidence of: URL to the application on Heroku. Adheres to a client-server architecture. Adheres to a client-server architecture. Filer header & in-line comments. Filer header & cypress are installed as dev dependencies. No dead or unused code. 	opriately.
functional. Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. README file contains thorough evidence of: URL to the application on Heroku. Adheres to a client-server architecture. Adheres to a client-server architecture. Filer header & in-line comments. Filer header & cypress are installed as dev dependencies. No dead or unused code. No dead or unused code. README file contains evidence of: URL to the application on Heroku. VAdheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Formatted code using Prettier. No dead or unused code using Prettier. Formatted code using	oonents are written as
Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. README file contains thorough evidence of: URL to the application on Heroku. Adheres to a client-server architecture. Filer header & in-line comments. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. No dead or unused code. README file contains evidence of: URL to the application on Heroku. URL to the application on Heroku. Adheres to a client-server architecture. Filer header & in-line comments. Filer header & in-line comments. Filer header & in-line comments. Formatted code using Prettier. Formatted code using Prettier. No dead or unused code on the prettier of the prettier of the prettier of the prettier of the prettier on the prettier of the prettier on the prettier of th	ional.
 Filer header & in-line comments. Formatted code using Prettier. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. 	res to a client-server
 Filer header & in-line comments. Formatted code using Prettier. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. 	tecture.
 Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. PREADME file contains thorough evidence of: URL to the application on Heroku. Formatted code using Prettier. Prettier & Cypress are installed as dev dependencies. No dead or unused code. Prettier & Cypress are installed as dev dependencies. No dead or unused code. 	header & in-line comments.
 Prettier & Cypress are installed as dev dependencies. No dead or unused code. Prettier & Cypress are installed as dev dependencies. No dead or unused code. Prettier & Cypress are installed as dev dependencies. No dead or unused code. Prettier & Cypress are installed as dev dependencies. No dead or unused code. No dead or unused code. Prettier & Cypress are installed as dev dependencies. No dead or unused code. No dead or unused code. Prettier & Cypress are installed as dev dependencies. No dead or unused code. No dead or unused code. No dead or unused code. URL to the application on Heroku. URL to the application on Heroku. 	atted code using Prettier.
dev dependencies. No dead or unused code. README file contains thorough evidence of: URL to the application on Heroku. dev dependencies. No dead or unused code. README file contains evidence of: URL to the application on Heroku. dev dependencies. No dead or unused code. README file contains evidence of: URL to the application on Heroku. URL to the application on Heroku.	ier & Cypress are installed as
● No dead or unused code. ● No	ependencies.
README file contains thorough evidence of: • URL to the application on Heroku. README file contains clear evidence of: • URL to the application on Heroku. README file contains evidence of: • URL to the application on Heroku. README file contains evidence of: • URL to the application on Heroku.	ead or unused code.
URL to the application on Heroku. URL to the application on Heroku. URL to the application on Heroku. Contain evidence.	
	•
How to setup the environment for development, run end-to-end development end run end-to-end end run end run end run end end run	
development, run end-to-end de	o the application on Heroku.
Cynress tests & denloy the Cynress tests & denloy test & Cynress tests & denloy test & Cynress tests & denloy the Cynress test & denloy test & Cynress & Cy	to setup the environment for
Cypress tests a deploy the	opment, run end-to-end
	ess tests & deploy the
	cation.
Comprehensive use of Markdown syntax, Substantial use of Markdown syntax, i.e., Use of Markdown syntax, i.e., headings, bold	
i.e., headings, bold text & code blocks. beadings, bold text & code blocks. beadings, bold text & code blocks. beadings, bold text & code blocks.	es not fully demonstrate use
of Markdown s	ntax, i.e., headings, bold text
Thorough spelling & grammar correctness. Clear spelling & grammar correctness. Spelling & grammar correctness. & code blocks.	
i.e., headings, bold text & code blocks. Thorough spelling & grammar correctness. Git commit messages are comprehensively formatted & reflect the functionality changes in succinct detail. i.e., headings, bold text & code blocks. Clear spelling & grammar correctness. Clear spelling & grammar correctness. Git commit messages are comprehensively reflect the functionality changes in substantial detail. Clear spelling & grammar correctness. Git commit messages are formatted & reflect the functionality changes in substantial detail. Cit commit messages are formatted & reflect the functionality changes in detail. Git commit messages in detail. Git commit messages in detail.	es fully demonstrate spelling
formatted & reflect the functionality changes reflect the functionality changes in the functionality changes in detail. & grammar cor	rectness.
in succinct detail. substantial detail.	
Git commit me	sages are not or are not fully
formatted & do	not or do not reflect the
functionality ch	

ID607001: Introductory Application Development Concepts

Project 2: React CRUD

Version 3, Semester One, 2022

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						
	/100							
This assessment is worth 50% of the final mark for the Introductory Application Development Concepts course.								
Feedback:								
Functionality:								
Code Elegance:								
Documentation & Git Usage:								

ID607001: Introductory Application Development Concepts

Project 2: React CRUD

Version 3, Semester One, 2022