

ID607001: Introductory Application Development Concepts

Practical: Node.js REST API Testing Research Assessment Rubric

	10-9	8-7	6-5	4-0
Functionality	<p>API tests demonstrate comprehensive & robust coverage on the following:</p> <ul style="list-style-type: none"> • Written using Mocha & Chai. • CRUD (create, read, update & delete) operations. • Authentication. • Validation rules. • Query parameters. • Status codes. • The shape of the data. • Code coverage using nyc. 	<p>API tests demonstrate clear & detailed coverage on the following:</p> <ul style="list-style-type: none"> • Written using Mocha & Chai. • CRUD (create, read, update & delete) operations. • Authentication. • Validation rules. • Query parameters. • Status codes. • The shape of the data. • Code coverage using nyc. 	<p>API tests demonstrate coverage on the following:</p> <ul style="list-style-type: none"> • Written using Mocha & Chai. • CRUD (create, read, update & delete) operations. • Authentication. • Validation rules. • Query parameters. • Status codes. • The shape of the data. • Code coverage using nyc. 	<p>API tests does not, or does not fully demonstrate coverage on the following:</p> <ul style="list-style-type: none"> • Written using Mocha & Chai. • CRUD (create, read, update & delete) operations. • Authentication. • Validation rules. • Query parameters. • Status codes. • The shape of the data. • Code coverage using nyc.
Code Elegance	<p>API tests thoroughly 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. • Filer header comments. • Formatted code using Prettier. • Prettier & nyc installed as dev dependencies. • No dead or unused code. • Database configured for testing environment. • Environment variables stored. 	<p>API tests clearly 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. • Filer header comments. • Formatted code using Prettier. • Prettier & nyc installed as dev dependencies. • No dead or unused code. • Database configured for testing environment. • Environment variables stored. 	<p>API tests 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. • Filer header comments. • Formatted code using Prettier. • Prettier & nyc installed as dev dependencies. • No dead or unused code. • Database configured for testing environment. • Environment variables stored. 	<p>API tests do not or do 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. • Filer header comments. • Formatted code using Prettier. • Prettier & nyc installed as dev dependencies. • No dead or unused code. • Database configured for testing environment. • Environment variables stored.

Documentation & Git Usage	README file contains thorough evidence of how to setup the environment for development & run the API tests.	README file contains clear evidence of how to setup the environment for development & run the API tests.	README file contains evidence of how to setup the environment for development & run the API tests.	README file does not or does not fully contain evidence of how to setup the environment for development & run the API tests.
	Comprehensive use of Markdown syntax, i.e., headings, bold text & code blocks.	Substantial use of Markdown syntax, i.e., headings, bold text & code blocks.	Use of Markdown syntax, i.e., headings, bold text & code blocks.	Does not or does not fully demonstrate use of Markdown syntax, i.e., headings, bold text & code blocks.
	Thorough spelling & grammar correctness.	Clear spelling & grammar correctness.	Spelling & grammar correctness.	
	Git commit messages are comprehensively formatted & reflect the functionality changes in succinct detail.	Git commit messages are clearly formatted & reflect the functionality changes in substantial detail.	Git commit messages are formatted & reflect the functionality changes in detail.	Does not or does not fully demonstrate spelling & grammar correctness. Git commit messages are not or are not fully formatted & do not or do not reflect the functionality changes.

ID607001: Introductory Application Development Concepts

Practical: Node.js REST API Testing Research Marking Cover Sheet

Name:

Date:

Learner ID:

Assessor's Name:

Assessor's Signature:

Criteria	Out Of	Weighting	Final Result
Functionality	10	60	
Code Elegance	10	30	
Documentation & Git Usage	10	10	
Final Result			/100
This assessment is worth 20% of the final mark for the Introductory Application Development Concepts course.			

Feedback:

Functionality:

Code Elegance:

Documentation & Git Usage: