Practical: Node.js REST API Testing Research Assessment Rubric

	10-9	**8-7**	**6-5**	**4-0**
Functionality	API tests demonstrate comprehensive & robust coverage on the following: • 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.	API tests demonstrate clear & detailed coverage on the following: • 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.	API tests demonstrate coverage on the following: • 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.	API tests does not, or does not fully demonstrate coverage on the following: • 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	API tests thoroughly demonstrate code elegance on the following: • 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.	API tests clearly demonstrate code elegance on the following: • 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.	API tests demonstrate code elegance on the following: • 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.	API tests do not or do not fully demonstrate code elegance on the following: • 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.

Documentation & Git Usage	README file contains thorough evidence of how to setup the environment for development & run the API tests. Git commit messages are comprehensively formatted & reflect the functionality changes in succinct detail.	README file contains clear evidence of how to setup the environment for development & run the API tests. Git commit messages are clearly formatted & reflect the functionality changes in substantial detail.	README file contains evidence of how to setup the environment for development & run the API tests. Git commit messages are formatted & reflect the functionality changes in detail.	README file does not or does not fully contain evidence of how to setup the environment for development & run the API tests. Git commit messages are not or are not fully formatted & do not or do not reflect the functionality changes.

Practical: Node.js REST API Testing Research

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: