Project 1: Laravel API Assessment Rubric

	10-9	8-7	6-5	4-0
Functionality	API contains comprehensive & robust evidence on the following: Application runs locally without modification. Models containing appropriate number of columns. Controller for each model which contain CRUD functionality. Custom validation when creating & updating data. Database tables seeded with JSON files. Appropriate status code & message returned when performing CRUD actions. Appropriate message returned when query does not return any data. Appropriate data returned using API Resources. Filter, sort & page API data from multiple models using query parameters. Data stored in & removed from the cache Protected routes using Sanctum. API rate limit set to 25 requests. Deployed to & usable on Heroku. API data stored in a MySQL development database & PostgreSQL production database.	API contains clear & detailed evidence of functionality on the following:	 API contains evidence on the following: Application runs locally without modification. Models containing appropriate number of columns. Controller for each model which contain CRUD functionality. Custom validation when creating & updating data. Database tables seeded with JSON files. Appropriate status code & message returned when performing CRUD actions. Appropriate message returned when query does not return any data. Appropriate data returned using API Resources. Filter, sort & page API data from multiple models using query parameters. Data stored in & removed from the cache Protected routes using Sanctum. API rate limit set to 25 requests. Deployed to & usable on Heroku. API data stored in a MySQL development database & PostgreSQL production database. 	 API does not, or does not fully contain evidence on the following: Application runs locally without modification. Models containing appropriate number of columns. Controller for each model which contain CRUD functionality. Custom validation when creating & updating data. Database tables seeded with JSON files. Appropriate status code & message returned when performing CRUD actions. Appropriate message returned when query does not return any data. Appropriate data returned using API Resources. Filter, sort & page API data from multiple models using query parameters. Data stored in & removed from the cache Protected routes using Sanctum. API rate limit set to 25 requests. Deployed to & usable on Heroku. API data stored in a MySQL development database & PostgreSQL production database.

Project 1: Laravel API

Version 2, Semester Two, 2021

API thoroughly demonstrates code API clearly demonstrates code elegance on API demonstrates code elegance on the API does not or does not fully elegance on the following: the following: following: demonstrate code elegance on the • Use of intermediate variables, Use of intermediate variables. Use of intermediate variables. following: i.e., no method calls as i.e., no method calls as i.e., no method calls as • Use of intermediate variables, i.e., no method calls as arguments. arguments. arguments. Idiomatic use of control flow, Idiomatic use of control flow, Idiomatic use of control flow, arguments. data structures and in-built data structures and in-built data structures and in-built Idiomatic use of control flow, functions. functions. functions. data structures and in-built Code Elegance Adheres to an OO architecture. Adheres to an OO architecture. Adheres to an OO architecture. functions. Adheres to an OO architecture. Efficient algorithmic approach, Efficient algorithmic approach, Efficient algorithmic approach, i.e., correct use of Eloquent. i.e., correct use of Eloquent. i.e., correct use of Eloquent. Efficient algorithmic approach, API resource groups named with API resource groups named with API resource groups named with i.e., correct use of Eloquent. a plural noun not verb. a plural noun not verb. a plural noun not verb. API resource groups named with In-line comments explain In-line comments explain In-line comments explain a plural noun not verb. complex logic. complex logic. complex logic. In-line comments explain Formatted code. Formatted code. Formatted code. complex logic. No dead or unused code. No dead or unused code. No dead or unused code. Formatted code. Databases configured for Databases configured for Databases configured for No dead or unused code. development & production development & production development & production Databases configured for environments. development & production environments. environments. environments. API documented in detail using Postman. API documented in succinct detail using API documented in substantial detail using API not or not full documented in detail Postman. Postman. using Postman. Git Usage README file contains evidence of: README file contains thoroughly evidence README file does not or does not fully README file contains clear evidence of: URL to API application on contain evidence of: URL to API application on Heroku. URL to API application on Heroku. URL to API documentation on • URL to API application on Heroku. **Documentation &** Heroku. URL to API documentation on Postman. URL to API documentation on Postman. How to setup the environment URL to API documentation on Postman. How to setup the environment for development & deploy the Postman. How to setup the environment for development & deploy the How to setup the environment application. for development & deploy the application. for development & deploy the Git commit messages formatted & reflect application. application. Git commit messages clearly formatted & the functionality changes in detail. Git commit messages comprehensively reflect the functionality changes in Git commit messages are not or are not formatted & reflect the functionality fully formatted & do not or do not reflect substantial detail. changes in succinct detail. the functionality changes.

Project 1: Laravel API 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 30% of the final mark for the Introductory Application Development Concepts course.

Functionality:

Code Elegance:

Documentation & Git Usage: