ID607001: Introductory Application Development Concepts

Project Marking Rubric

	10-9	8-7	6-5	4-0
Functionality – Your choice REST API	The Express REST API developed using Node.js contains comprehensive and robust evidence on the following functionality: development and production modification, models, data types, relationships, enum, files, messages, filtering, sorting, pagination, a 404 endpoint, validation, Swagger documentation, PostgreSQL database, deployment and scripts.	The Express REST API developed using Node.js contains clear and detailed evidence on the following functionality: development and production modification, models, data types, relationships, enum, files, messages, filtering, sorting, pagination, a 404 endpoint, validation, Swagger documentation, PostgreSQL database, deployment and scripts.	The Express REST API developed using Node.js contains evidence on the following functionality: development and production modification, models, data types, relationships, enum, files, messages, filtering, sorting, pagination, a 404 endpoint, validation, Swagger documentation, PostgreSQL database, deployment and scripts.	The Express REST API developed using Node.js does not or does not fully contain evidence on the following functionality: development and production modification, models, data types, relationships, enum, files, messages, filtering, sorting, pagination, a 404 endpoint, validation, Swagger documentation, PostgreSQL database, deployment and scripts.
	10-9	8-7	6-5	4-0
Functionality – OpenTDB API	The Express REST API developed using Node.js contains comprehensive and robust evidence on the following functionality: development and production modification, enums, models, admin user, basic user, validation, seeding, Helmet, CORS, rate limiting, compression, Swagger documentation, PostgreSQL database, deployment and scripts.	The Express REST API developed using Node.js contains clear and detailed evidence on the following functionality: development and production modification, enums, models, admin user, basic user, validation, seeding, Helmet, CORS, rate limiting, compression, Swagger documentation, PostgreSQL database, deployment and scripts.	The Express REST API developed using Node.js contains evidence on the following functionality: development and production modification, enums, models, admin user, basic user, validation, seeding, Helmet, CORS, rate limiting, compression, Swagger documentation, PostgreSQL database, deployment and scripts.	The Express REST API developed using Node.js does not or does not fully contain evidence on the following functionality: development and production modification, enums, models, admin user, basic user, validation, seeding, Helmet, CORS, rate limiting, compression, Swagger documentation, PostgreSQL database, deployment and scripts.

ID607001: Introductory Application Development Concepts

Project

Version 2, Semester Two, 2024

	The Express REST API developed using	The Express REST API developed using	The Express REST API developed using	The Express REST API developed using
	Node.js demonstrate comprehensive	Node.js demonstrate clear evidence on	Node.js demonstrate evidence on the	Node.js do not or do not fully demonstrate
st	evidence on the following:	the following:	following:	evidence on the following:
B	 Environment variables. 	Environment variables.	Environment variables.	Environment variables.
nd s	 Appropriate naming. 	 Appropriate naming. 	Appropriate naming.	 Appropriate naming.
luality an Practices	Idiomatic use.	Idiomatic use.	Idiomatic use.	Idiomatic use.
act alit	 Efficient algorithmic 	Efficient algorithmic approach.	Efficient algorithmic approach.	Efficient algorithmic approach.
ا پر ج	approach.	Sufficient modularity.	Sufficient modularity.	Sufficient modularity.
<u>e</u>	 Sufficient modularity. 	JSDoc header comment.	JSDoc header comment.	 JSDoc header comment.
Code Quality and Best Practices	 JSDoc header comment. 	Code is formatted.	Code is formatted.	 Code is formatted.
	 Code is formatted. 	No dead or unused code.	No dead or unused code.	No dead or unused code.
	 No dead or unused code. 			
	Comprehensive use of project board or	Clear use of project board or issues on	Use of project board or issues on GitHub.	Does not or does not full demonstrate use
	issues on GitHub.	GitHub.		of project board or issues on GitHub.
			README file contains evidence of:	
	README file contains comprehensive	README file contains clear evidence of:	 A URL to your REST APIs as web 	README file does not or does not fully
	evidence on the following:	 A URL to your REST APIs as web 	service on Render.	contain evidence of:
e e	 A URL to your REST APIs as 	service on Render.	 Setup the environment. 	A URL to your REST APIs as web
gg	web service on Render.	 Setup the environment. 	 Run your REST APIs locally. 	service on Render.
Usage	 Setup the environment. 	 Run your REST APIs locally. 	 Create and apply a migration. 	Setup the environment.
Git	 Run your REST APIs locally. 	 Create and apply a migration. 	 Reset the PostgreSQL database. 	Run your REST APIs locally.
0	 Create and apply a migration. 	 Reset the PostgreSQL database. 	Seed users.	 Create and apply a migration.
and	 Reset the PostgreSQL 	 Seed users. 	 Open Prisma Studio. 	 Reset the PostgreSQL database.
	database.	 Open Prisma Studio. 	Check your code.	Seed users.
Documentation	 Seed users. 	Check your code.	 Format your code. 	Open Prisma Studio.
ia i	 Open Prisma Studio. 	 Format your code. 	 An ERD of your REST APIs. 	Check your code.
ב	 Check your code. 	 An ERD of your REST APIs. 	 Use of Markdown. 	Format your code.
ΙĔ	 Format your code. 	 Use of Markdown. 	 Spelling and grammar 	An ERD of your REST APIs.
5	 An ERD of your REST APIs. 	 Spelling and grammar 	correctness.	 Use of Markdown.
Ŏ	 Use of Markdown. 	correctness.		Spelling and grammar
_	 Spelling and grammar 		Git commit messages are formatted and	correctness.
	correctness.	Git commit messages are clearly	reflect the changes in detail.	
		formatted and reflect the changes in		Git commit messages are not or are not
	Git commit messages are	substantial detail.		fully formatted and do not or do not fully
	comprehensively formatted and reflect			reflect the changes.
	the changes in concise detail.			

ID607001: Introductory Application Development Concepts

Project

Version 2, Semester Two, 2024

ID607001: Introductory Application Development Concepts Project Marking Cover Sheet

Name:			
Date:			
Learner ID:			
Assessor's Name:			
Assessor's Signature:			
Criteria	Out Of	Weighting	Final Result
Functionality	10	50	
Functionality Code Quality and Best Practices	10 10	50 40	
Code Quality and Best			
Code Quality and Best Practices Documentation and Git	10	40	/100
Code Quality and Best Practices Documentation and Git	10 10 Final Result	10	· · · · · · · · · · · · · · · · · · ·
Code Quality and Best Practices Documentation and Git Usage	10 10 Final Result	40 10 the Introductory Appli	· · · · · · · · · · · · · · · · · · ·

Feedback:

Functionality:

Code Quality and Best Practices:

Documentation and Git Usage:

ID607001: Introductory Application Development Concepts

Project

Version 2, Semester Two, 2024