Assessment Marking Criteria

ICTPRG553_554_AT2_MC_TQM_v3.docx

Student Name		**Student Number**	
Unit Code/s & Name/s	ICTPRG553 Create and develop REST APIs ICTPRG554 Manage data persistence using NoSQL data stores		
Cluster Name *If applicable*	Web data cluster		
Assessment Type	☐ Assignment ☐ Project ☐ Case Study ☒ Portfolio ☐ Third Party Report (Workplace) ☐ Third Party Report (Peer) ☐ Other		
Assessment Name	Website Information Architecture Portfolio	**Assessment Task No.**	2 of 2
Assessment Due Date		**Date Submitted**	/ /
Assessor Feedback:			

Attempt 1	Satisfactory ☐ Unsatisfactory ☐	**Date**	/ /
Assessor Name		**Assessor Signature**	
☐ **Student provided with feedback and reassessment arrangements** *(check box when completed)*		**Date scheduled for reassessment**	/ /
Attempt 2	Satisfactory ☐ Unsatisfactory ☐	**Date**	/ /
Assessor Name		**Assessor Signature**	
Note to Assessor: Please record below any reasonable adjustment that has occurred during this assessment e.g. written assessment given orally.			

Assessment Criteria / Benchmarks ***The evidence submitted demonstrates that the student has satisfactorily:***	**Attempt 1** **Date** **__/__/__**		**Attempt 2** **Date** **__/__/__**	
	Y	**N**	**Y**	**N**
PART 1, Task 1: NoSQL research and technology selection	☐	☐	☐	☐
1.1 Evidence that the scenario has been reviewed and analysed is presented in the three (3) reasons provided to justify NoSQl suitability.	☐	☐	☐	☐
1.2 a) Research on vertical (scale-up) and horizontal (scale-out) scaling methods and the reasons why horizontal scaling is better suited for the scenario presented has been provided.	☐	☐	☐	☐
b) Explanation of at least two (2) benefits of using horizontal scaling for the project provided.	☐	☐	☐	☐
1.3 Research and compare relational databases (SQL-based) and non-relational (NoSQL) databases evidence provided. Table completed.	☐	☐	☐	☐
1.4 Research and review at least three (3) NoSQL interfaces or vendor technologies provided. Table completed	☐	☐	☐	☐
1.5 REST API need has been justified for the given scenario. The justification addressed the following points: • Suitability to the scenario presented	☐	☐	☐	☐
• Flexibility and portability	☐	☐	☐	☐
• Scalability	☐	☐	☐	☐
• Cacheability	☐	☐	☐	☐
1.6 Reviewed Web API frameworks available for the chosen programming language and selected a framework that is suitable and framework selection justification provided.	☐	☐	☐	☐
PART 1, Task 2: Storage requirements and creation				
1.7 For the NoSQL interface selected in the previous task: • Identified and designed the data storage requirements for the type of data used in this project	☐	☐	☐	☐
• Calculated and determined the read and write throughputs in the NoSQL database, and provided details about how this can be managed.	☐	☐	☐	☐
• Determined the appropriate type of NoSQL data store to be used, describing it and its benefits.	☐	☐	☐	☐

• Reviewed the business needs presented and selected the most appropriate data (formats) types for the NoSQL datastore for this project. Table completed (with more tables added if needed).	☐	☐	☐	☐
• Created the necessary document/collection schema in MongoDB for the scenario presented. Documentation of the database structure and its notation presented.	☐	☐	☐	☐
• Sourced the data from the dataset provided and populated the datastore.	☐	☐	☐	☐
1.8 Determined indexing needs to suit the scenario presented and complete the following: • Configured, and created a single field index in a collection to optimise data retrieval.	☐	☐	☐	☐
• Configured and created either a multikey index or a compound index to optimise data retrieval.	☐	☐	☐	☐
1.9 Presented the completed MongoDB database setup to the manager, or relevant person in the organisation, for approval and signoff. Evidence provided.	☐	☐	☐	☐
Part 2, Task 1: Build the REST API project				
2.1 Created a RESTful API project, and screenshot provided.	☐	☐	☐	☐
2.2 Implemented the required database connection, ensuring the API project can communicate with the database. Provided a written description of the connection, and a screenshot(s) demonstrating the connection.	☐	☐	☐	☐
2.3 Defined methods within the API project to allow for each of the following functions, ensuring that the functionality meets the requirements of the project: • Retrieve a single record	☐	☐	☐	☐
• Retrieve multiple records	☐	☐	☐	☐
• Insert a single record	☐	☐	☐	☐
• Insert multiple records	☐	☐	☐	☐
• Update or replace an existing record	☐	☐	☐	☐
• Update or replace multiple records	☐	☐	☐	☐
• Delete a single record	☐	☐	☐	☐
• Delete multiple records	☐	☐	☐	☐
• Utilise projection on at least one occasion	☐	☐	☐	☐

2.4 Created endpoints in a RESTful API that: • Communicate directly with the NoSQL database, or use the methods created in 2.3.	☐	☐	☐	☐
• Contain appropriate validation to ensure that correct and incorrect requests are handled appropriately.	☐	☐	☐	☐
2.5 The created endpoints have been tested, demonstrating the results of correctly and incorrectly formatted requests to the created endpoints. Table completed.	☐	☐	☐	☐
2.6 Provided a detailed summary of the testing conducted. • Appropriate detail provided around how correct and incorrect data was determined. • Unexpected results (if any) have been described, along with adjustments made to handle these results.	☐	☐	☐	☐
2.7 Provided at least one screenshot demonstrating testing of each individual endpoint.	☐	☐	☐	☐
Part 3, Task 1: Configure CORS				
3.1 CORS for GET, POST, PUT, DELETE, and PATCH methods enabled. Screenshots of configuration provided.	☐	☐	☐	☐
3.2 Provided screenshots of requests made to endpoints, demonstrating the correct configuration of CORS for the endpoints.	☐	☐	☐	☐
3.3 Configured the API to allow for receiving and handling pre-flight requests.	☐	☐	☐	☐
3.4 Cross-origin requests have been tested on the client with at least five (5) instances. One per method is sufficient. Table completed.	☐	☐	☐	☐
Part 3, Task 2: Evaluate and secure REST API	☐	☐	☐	☐
3.5 Described the Authentication and Authorisation methods to secure the application, as per the business requirements provided.	☐	☐	☐	☐
3.6 Individual endpoints that will require authentication and authorisation as per the project requirements have been identified. Table completed.	☐	☐	☐	☐
3.7 Authentication and authorisation methods proposed have been implemented.	☐	☐	☐	☐
3.8 Authentication and authorisation methods have been tested and screenshots provided.	☐	☐	☐	☐
3.9 Provided screenshots that show the request and the API's response in the following situations: (at least once for each situation) • Authentication succeeding	☐	☐	☐	☐

• Authentication failed	☐	☐	☐	☐
• Authorisation succeeding	☐	☐	☐	☐
• Authorisation failed	☐	☐	☐	☐
Part 3, Task 3: Document REST API				
3.10 Compared and evaluated at least three (3) API documentation tools., and completed the table. At least one tool evaluated implements the OpenAPI specification and is compatible with the project	☐	☐	☐	☐
3.11 Using an OpenAI documentation tool, documented each REST API endpoint, and described the following: • Endpoint's purpose.	☐	☐	☐	☐
• Required parameters, and optional parameters.	☐	☐	☐	☐
• Expected responses from the endpoint.	☐	☐	☐	☐
3.12 Provided evidence of the documentation for each endpoint.	☐	☐	☐	☐
3.13 Presented the API documentation to the manager, or relevant person in the organisation, for approval and signoff. Evidence provided.	☐	☐	☐	☐
PART 4, Task 1: Database Tasks - configuration				
4.1 Determined and implemented a time-to-live (TTL) or special single-field index on a field in a collection. The TTL index automatically removes the document after a certain amount of time has passed or a specific clock time as per business requirements .	☐	☐	☐	☐
4.2 MongoDB has been configured to accept persistence of objects including objects of different data types. • Screenshots of the configuration file and options have been provided.	☐	☐	☐	☐
4.3 In accordance with the scenario provided: • At least two (2) triggers have been proposed and the corresponding events and notifications have been identified.	☐	☐	☐	☐
• Email to the manager or relevant stakeholder seeking confirmation of the triggers before implementing them has been provided as evidence.	☐	☐	☐	☐
4.4 The triggers have been implemented and tested. Screenshots of code and of testing are provided for each trigger.	☐	☐	☐	☐
4.5 Configured Authentication and Authorisation as per business requirements. Screenshots of the configuration process are provided, demonstrating the results of using both correct credentials and incorrect credentials.	☐	☐	☐	☐

4.6 A detailed description of how the encryption that has been used in this project meets the encryption requirements provided in the business requirements.	☐	☐	☐	☐
PART 4, Task 2: Database Tasks - Partitioning				
4.7 Identified at least one (1) collection that could benefit from partitioning - Determined the partition key and provided justification as to why it is suitable for partitioning the collection.	☐	☐	☐	☐
4.8 Created the partition based on the partition key determined in 4.7, Provided screenshots of the following process as evidence. • Configuring the database and collection for partitioning.	☐	☐	☐	☐
• Implementing the partition key.	☐	☐	☐	☐
• The results of the partitioning – showing the collection name, the number of chunks and shards, and the distribution of data.	☐	☐	☐	☐
4.9 Reviewed the partition created above and outlined a process to maintain or achieve balanced spread of data cross partitions, including optimising the data or the partition key	☐	☐	☐	☐
4.1T he completed Database Configuration and Partitioning tasks have been submitted to the manager, or relevant person in the organisation, for approval and signoff.	☐	☐	☐	☐
PART 5: Contingency task and knowledge concepts related to this project				
5.1 Contingency task: Some MongoDB documents are sometimes removed without any apparent operation being performed on that particular data store. Answer provided detailing how to investigate this hypothetical problem.	☐	☐	☐	☐
5.2 In relation to this project, identified the programming language used to interact with the NoSQL, and the data interchange format language. Assessed the performance and the suitability of the language selected for task.	☐	☐	☐	☐
5.3 In relation to the project completed in this portfolio: • Outlined the criteria that were used to partition a specific data store, and described the benefits achieved.	☐	☐	☐	☐
• What scalability method makes possible the distribution of data across partitions in NoSQL databases? Describe how the distribution process works. Appropriate answer provided.	☐	☐	☐	☐
• Explained how sort keys can be used in a partition to increase performance, and outlined their functions and features in a NoSQL implementation.	☐	☐	☐	☐

5.4 Described how TTL was used in the project, and described the features and functions implemented.	☐	☐	☐	☐
5.5 Described the implementation of transport encryptions, authentication, and authorisation in the REST API. Clarified how the methods and features used contribute to securing the application.	☐	☐	☐	☐
5.6 In relation to the debugging and testing of the project: • Identified the debugging and testing methodology used and described four (4) techniques used to debug and test the API.	☐	☐	☐	☐
• For each technique, evaluated its effectiveness in this project.	☐	☐	☐	☐
5.7 Identified and appraised the datastore format used for this portfolio project. Table completed showing a brief description of each datastore format presented.	☐	☐	☐	☐
5.8 For each data format listed below, appropriate answers were provided that explained the range of values allowed and provided an example of the code/notation used to declare them.	☐	☐	☐	☐
• numeric	☐	☐	☐	☐
• string	☐	☐	☐	☐
• Boolean	☐	☐	☐	☐
• complex	☐	☐	☐	☐
• date time	☐	☐	☐	☐