Assessment Task
Portfolio of Evidence

ICTPRG553_554_AT2_PE_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 Name	Website Information Architecture Portfolio	**Assessment Task No.**	2 of 2
Assessment Due Date		**Date Submitted**	/ /
Assessor Name			

Student Declaration: I declare that this assessment is my own work. Any ideas and comments made by other people have been acknowledged as references. I understand that if this statement is found to be false, it will be regarded as misconduct and will be subject to disciplinary action as outlined in the TAFE Queensland Student Rules. I understand that by emailing or submitting this assessment electronically, I agree to this Declaration in lieu of a written signature.

Student Signature		**Date**	/ /

PRIVACY STATEMENT: TAFE Queensland is collecting your personal information on this form for the purpose of assessment. In accordance with the Information Privacy Act 2009 (Qld), your personal information will only be accessed by staff employed by TAFE Queensland for the purposes of conducting assessment. Your information will not be provided to any other person or agency unless you have provided TAFE Queensland with permission, if authorised under our Privacy Policy (available at https://tafeqld.edu.au/global/privacy-policy.html) or disclosure is otherwise permitted or required by law. Your information will be stored securely. If you wish to access or correct any of your information, discuss how it has been managed or have a concern or complaint about the way the information has been collected, used, stored, or disclosed, please contact the TAFE Queensland Privacy Officer at privacy@tafeqld.edu.au

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

Instructions to Student	**General Instructions:** You are employed by Uptown IT as a Software/Web Developer. You have been assigned to a new project and your task is to develop a MongoDB REST API to interact with a large dataset of raw climate data. Your teacher/assessor will take on the role of the Project Manager assigned to this project by Uptown IT. Read the project documentation provided and familiarise yourself with the Project Scenario or Case Study before proceeding with portfolio tasks. Confirm anything you are not sure about the project with your manager (teacher/assessor). It is essential that you have a clear understanding of the scenario and tasks that you need to complete. This assessment instrument requires the student to complete a project portfolio that is divided into five (5) parts: PART 1 – Selecting and preparing the NoSQL database PART 2 – RESTful API selection, creation, and testing PART 3 – RESTful API Configuration PART 4 – Database Tasks PART 5 – Contingency task and knowledge concepts related to this project **Materials Required:** • Students are required to provide their own storage device. The recommendation for this qualification is an external SSD drive with at least 500 GB capacity, if you need to store a copy of the Virtual Machine (VM). For assessment files only, a 64 GB thumb drive will be sufficient. • Access to an integrated IDE environment for web development • Access to an online or local provider for a NoSQL Database, to allow for configuring security, access and triggers. • Access to open-source or commercial NoSQL database • Access to the Internet • Access to Connect (LMS) • Access to special-purpose tools, equipment and materials to complete the assessment, for example, editing and testing tools. **Online Delivery:** • Student to supply their own PC or laptop and peripherals and internet access

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

- Students should have sufficient permissions to be able to install software on their computers e.g. IDE software
- Students will require access to Microsoft Office or similar
- Students will require access to a web server or the ability to install a local web server on their computer (instructions for this will be provided)

Documentation:

- Uptown IT Scenario or Case Study
- Uptown IT REST API NoSQL Template

Details of location:

TAFE will provide a simulated work environment in the classroom. Research activities may be conducted in the classroom or at home.

If you are unable to attend a scheduled assessment activity, you must notify your teacher before the assessment is due and supply a doctor's certificate and approval from the team manager for an extension.

Time restrictions:

This assignment is designed to take place over 8 weeks or approximately 32 hours. The student is expected to attend classes as per timetable details and should be able to commit up to 3 hours per week of their own time to study or study related activities.

Interactions:

Teamwork skills are essential in the IT industry therefore you should work in teams to consult and collaborate on practical activities. However, each student must complete the assessment tasks individually (unless indicated).

Level of assistance permitted:

Staff cannot directly show students answers or solutions but support and guide them to complete tasks individually. Teachers and tutors should be available in class, and accessible by email for students working from home.

Work, Health and Safety:

The work environment should be assessed for safety prior to class. Special consideration should be taken regarding potential ICT related hazards such as tripping hazards, electromagnetic radiation, ergonomics, and posture. TAFE Queensland health and safety policies and procedures should be followed at all times.

Assessment Criteria:

To achieve a satisfactory result, your assessor will be looking for your ability to demonstrate the following key skills/tasks/knowledge to an acceptable industry standard:

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

	• Confirm RESTful API functionality • Build and test a RESTful API application • Build and test REST API endpoints and corresponding methods • Enable RESTful API cross-origin resource sharing (CORS) • Work with HTTP methods (GET, POST, PUT, DELETE, PATCH) • Secure RESTful API with authentication and authorisation • Document and test RESTful API • Understand NoSQL databases and select the best NoSQL option for each implementation • Work with NoSQL data storage options including partitioning and effective scale-out distribution • Determine indexing needs and types of indexing in NoSQL • Determine and implement time-to-live (TTL) operations.
Submission details (if relevant)	**Evidence Required to be Submitted:** Insert your details on the cover page and sign the Student Declaration. Include this template with your submission. **Submission via Connect:** Upload a single file into Assessment 2 (AT2) Assignment Folder in Connect. Multiple files can be compressed into a single file. **Name the file:** ICTPRG553_554_AT2_Surname_Student Number Assessment to be submitted via: • TAFE Queensland Learning Management System (Connect): *https://connect.tafeqld.edu.au/d2l/login* • **Username:** 9 digit student number • **For Password:** Reset password go to: *https://passwordreset.tafeqld.edu.au/default.aspx*
Instructions to Assessor	**Online Delivery:** Please revise and modify the Instructions to Student section if you are delivering online. **Specifications of assessment:** To be judged competent in this assessment item the student is required to

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

	demonstrate competence in all indicators shown in the marking guide. Gather evidence to demonstrate consistent performance in conditions that are safe and replicate the workplace. Noise levels, production flow, interruptions and time variances must be typical of those experienced in the web development field of work and include access to: • web development environment • project requirements Ensure that students read and familiarise themselves with the Project Scenario and the Client provided relevant files and/or resources before attempting the assessment. **Storage Devices:** Students are required to provide their own storage device. The recommendation for this qualification is an external SSD drive with at least 500 GB capacity, if you need to store a copy of the Virtual Machine (VM). For assessment files only, a 64 GB thumb drive will be sufficient. **Assessor to Provide:** • Students are required to provide their own storage device. The recommendation for this qualification is an external SSD drive with at least 500 GB capacity, if you need to store a copy of the Virtual Machine (VM). For assessment files only, a 64 GB thumb drive will be sufficient. • Access to an integrated IDE environment for web development • Access to an online or local provider for a NoSQL Database, to allow for configuring security, access and triggers. • Access to open-source or commercial NoSQL database • Access to the Internet • Access to Connect (LMS) • Access to special-purpose tools, equipment and materials to complete the assessment, for example, editing and testing tools. **Online Delivery:** • Students to supply their own PC or laptop and peripherals and internet access • Students should have sufficient permissions to be able to install software on their computers e.g. IDE software • Students will require access to Microsoft Office or similar

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

	• Students will require access to a web server or the ability to install a local web server on their computer (instructions for this will be provided) **Documentation:** • Uptown IT Scenario or Case Study • Uptown IT REST API NoSQL Template **Work, Health and Safety:** The work environment should be assessed for safety prior to class. Special consideration should be taken regarding potential ICT related hazards such as tripping hazards, electromagnetic radiation, ergonomics, and posture. TAFE Queensland health and safety policies and procedures should be followed at all times.
Note to Student	*An overview of all Assessment Tasks relevant to this unit is located in the Unit Study Guide.*

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

I. Project Scenario

You are employed by Uptown IT as a Software/Web Developer. You have been assigned to a new project and your task is to develop a MongoDB REST API to interact with a large dataset of raw climate data.

The client for this project is an educational institution that wants to use raw climate data to run student projects to analyse climate data collected in areas QLD using a distributed Internet of Things Sensor network.

The Interface for interacting with this data needs to be a RESTful Web API that will be developed to suit the data stored in the database, the functionality required around accessing and manipulating this data, and Authentication and Authorisation of requests.

The database must include a collection of Weather Data, as well as a collection of Users. The Users collection will be designed to store Authentication and Authorisation information for a user account that is used to interact with the **RESTful Web API**

The client has indicated that they will be designing their own client to interact with the developed RESTful Web API, and require all of the functionality outlined in this document to be provided through the API, including API documentation to allow for easily generating a client in a given language. The language and framework used to design the RESTful Web API is not important to the client, as long as conventions are followed when designing the API.

Your teacher/assessor will take on the role of the Project Manager assigned to this project by Uptown IT.

II. RAW data access

The project will use a simulated set of data, representing weather recording values from 3 different areas of the Sunshine Coast and is included in a zip file provided with this assessment: ICTPRG553_554_WeatherData_Readings_2022.zip

The climate data is provided as a single collection of data points in JSON format, with a variety of different data points relating to weather observations.

III. PROJECT REQUIREMENTS

MongoDB schema

The raw data must be structured into a suitable MongoDB schema containing the necessary documents and collections to satisfy the business project requirements.

In addition to the data provided you need to create user account to implement Authentication and Authorisation for the API Project

MongoDB data storage, partition, and scalability

The client has indicated that they will have access to multiple pieces of hardware that they would like to run the database from, utilising the hardware (similar to a Raspberry pi 4) efficiently so that no one database server is doing all of the work. It is your job to correctly determine and implement the mechanism that will allow this functionality across the clients' hardware.

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

MongoDB and API settings and data constraints

The following settings and constraints must be included:

Data **persistence** is required.

- **Indexing** – The client has indicated that most of their queries will be looking at temperature fluctuations over a period of time, and the combination of humidity and rainfall over periods of time. Ensure that there are indexes created that will assist in these queries
- **TTL** – The client has requested that the developed project is able to remove users with a 'student' role that have not accessed the dataset for a period of at least 30 days, to ensure there are no un-used user accounts with access to the database.
- **Triggers** – The client has requested that there are triggers created on the database, to automate a range of tasks, at least 2 of the proposed triggers below must be implemented in the database itself.
  - Remove documents on edit or creation that contain invalid weather readings:
    - the humidity is greater than 100%
    - the temperature is greater than 60 degrees Celsius
    - the temperature is less than -50 degrees Celsius
  - Log any deletions of weather data – ensuring that the deleted data is stored in a 'log' collection.
  - Update the last login date of a user, whenever a user successfully queries the database

MongoDB and API security requirements

Security requirements must include:

- The client requires that the data is **encrypted in transit** when data is sent across the network, and that the data is **encrypted at rest** when stored on the database server. Further encryption is not necessary due to the nature of the stored data.
- Individual User accounts need to be created to control authentication with the API endpoints, and to control Access to specific endpoints, this will be separate to the User created to provide security between the API and Database, specifically the following user accounts
  - **Teacher** accounts that will be able to Modify, Remove, Read and Add new Users and weather readings
  - **User** accounts that will be able to Read weather data only
  - **Sensor** accounts that will be able to Add new weather readings only
- The client has also noted that the connection between the web API and the database should be secure enough that the credentials used in the connection string cannot be used to delete or create new Databases or Collections on the target server.
  - A User account configured with appropriate Authorisation is required to be created on the Database Server to allow for managing access between the API and the Database

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

  - Multiple User accounts are required to be created for access to the API to control the actions a specific user is Authorised to perform

RESTful Web API Constraints and Conventions

- The developed web API must return status codes that clearly reflect the outcome of an operation, including but not limited to:
  - 400
  - 404
  - 200
  - 201
  - 500
- The developed web API must be documented well enough that any consumer of the API can determine the purpose, parameters and expected responses from each endpoint.
- The selected Web API Framework or language should be able to interact with the selected NoSQL provider without issue.

RESTful API MongoDB interactions and operations requirements

Although the client may run many and varied CRUD operations for the project, the following operations have been selected to test the application - **these must be functional and tested in the final product**:

- *Insert a new reading for a weather station:*

To allow for the connection of new sensors to the dataset through this endpoint, will receive weather data and validate this data before inserting into the database.

- *Insert a new user:*

Allow Administrators to create a new user account with a specified level of access

- *Insert multiple sensor readings for single station:*

To allow for batch inserts every day, rather than one-by-one as the data is recorded

- *Find the maximum precipitation recorded in the last 5 Months for a specific sensor, returning the sensor name, reading date / time and the precipitation value:*

To allow students to retrieve data that might provide insights into changes in weather patterns based on temperature fluctuations.

- *Find the temperature, atmospheric pressure, radiation, and precipitation recorded by a specific station at a given date and time.*
- *Find the maximum Temp(C) recorded for all stations for a given Date / Time range (start and finish date) returning the Sensor Name, reading date / time and the Temperature value:*
  - To allow students to retrieve data that might provide insights into changes in weather patterns based on temperature fluctuations.

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

- *Create a query that includes an index key:*
  - provide efficient querying by ensuring that an index is included for large or complex queries
- *Delete a user by Id:*
  - Remove a single user account - revoking access to Authorised actions
- *Delete multiple users that have the 'Student' role and last logged in between two given dates:*
  - Allow Administrators to remove a range of created user accounts based on a date range.
- *Update a specified entries' precipitation value to a specific value:*
  - To allow Admin users to correct errors in the dataset
- *Update access level for at least two users in the same query, based on a date range in which the users were created:*
  - To Allow Administrators to upgrade or downgrade multiple users that were created in batches (upgrading or downgrading an entire group of students)

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

PART 1 – Selecting and preparing the NoSQL database

Use the Uptown IT REST API and NoSQL Template provided to document all tasks in PART 1

TASK 1. NoSQL research and technology selection

1.1 Review and analyse the scenario presented with the aim of identifying the business data needs and confirming that NoSQL is the best option for this project.

Outline in detail at least three (3) reasons to justify the use of NoSQL for the scenario presented.

1.2 Research vertical (scale-up) and horizontal (scale-out) scaling methods and identify the reasons why horizontal scaling is better suited for the scenario presented. Review the business requirements. Explain in detail at least two (2) benefits of using horizontal scaling for the project.

1.3 Research and compare relational databases (SQL-based) and non-relational (NoSQL) databases. Complete the table below. Write your appraisal in the corresponding cells.
A "tick" will not be considered an appropriate answer.

TYPE of DATABASE	**SUITABILITY by DATA VOLUME**	**SUITABILITY by DATA TYPE** **Structured, Semi-Structured, Unstructured**	**QUERY COMPLEXITY**	**ACID COMPLIANCE**
Relational: SQL				
Non-relational: NoSQL				

1.4 Research and review at least three (3) NoSQL interfaces or vendor technologies. This information will be used by your manager and/or relevant personnel to discuss and select the most appropriate NoSQL solution.

NoSQL Interface	**MAIN FEATURES**	**DATA FORMATS**	**EASE OF USE**	**BEST FOR** **Types of Projects**
Add rows as necessary				

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

*You have been informed that the NoSQL interface selected for the project is **MongoDB**. Keep that in mind while completing the tasks below.*

1.5 Read and analyse the scenario presented and justify the **need** for a REST API solution for the given scenario. In the justification, address the following points:

- Suitability to the scenario presented
- Flexibility and portability
- Scalability
- Cacheability

1.6 Review and evaluate Web API frameworks and IDEs available for your current project, select a framework and IDE that you feel is suitable based on your chosen programming language and justify your selection by providing a detailed description of how this framework and IDE will meet the needs described in 1.5 and the scenario. Include an evaluation of at least one (1) other Web API framework and IDE that could be used.

Task 2. Storage requirements and creation

1.7 For the NoSQL interface selected in the previous task:

- Based on the data provided and the project scenario, determine the current and future (1 year) data storage requirements. This should factor in the amount of data that currently existed, and how this will grow over time.
- Calculate the average amount of data that may be input and output from the NoSQL database over a period,(in Kilobytes / Second). Provide details around how this can be maintained as the dataset grows, and options for handling the growth of the throughput of the API and Database project.
- Determine the most appropriate type of NoSQL Data store to be used in this project, describing this Data Store and the benefits of utilising this for the given scenario.
- Review the provided dataset and determine the best datatypes for each field, based on the selected NoSQL, provide this in a table showing the individual fields and the planned datatype for each field. Where there may be multiple collections, more tables should be added to help describe these collections.

Field Name	**Proposed Data type**
Add rows as necessary	

- Source the data from the dataset provided and populate the datastore.

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

1.8 Determine indexing needs to suit the scenario presented and complete the following, ensuring that a brief description of the suitability of each index and a screenshot of each index that clearly shows the sort key selected is provided.

a) Configure and create a single field index in a collection to optimise data retrieval.

b) Configure and create either a multikey index or a compound index to optimise data retrieval.

1.9 Present the completed MongoDB database setup to your manager or relevant person in the organisation for approval and signoff.

<table>
<tr><td colspan="3">Part 1 – Planning and Database SIGNOFF
Signing off on this document signifies that the MongoDB database setup and planning documentation presented complies with the Client’s Business requirements.</td></tr>
<tr><td>Project Manager or relevant stakeholder
Signature:
Date:</td><td colspan="2">Web Developer
Signature:
Date:</td></tr>
<tr><td colspan="3">Initial Review

(Feedback provided by Teacher)</td></tr>
<tr><td>APPROVAL</td><td>☐ Granted</td><td>☐ Not Granted</td></tr>
<tr><td colspan="3">Secondary Review (If adjustments are required in the Initial Review)

(Feedback provided by Teacher)</td></tr>
<tr><td>APPROVAL</td><td>☐ Granted</td><td>☐ Not Granted</td></tr>
</table>

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

PART 2 – RESTful API Creation

In this part, you are creating the RESTful API project, connecting to the database, defining database interaction methods, endpoints and testing the integration.

In order to comply with Uptown IT documentation procedures, a template has been provided to document the activities for PART 2.

Task 1. Build the REST API Project

2.1 Create a RESTful web API project using the selected Ide and framework described in Part 1. Provide a screenshot from within the selected IDE that demonstrates the empty project after creation.

2.2 Determine and implement the required database connection strategies and tools, ensuring that the API project can communicate with the Database created in Task 1.

- Provide a written description of the mechanism used to connect the API to the Database and include a screenshot demonstrating the connection string used to make the Database connection.

2.3 Define **methods** within the API project to interact with the Database, ensuring that **all of** the functionality required in the project can be achieved through these methods, this includes ensuring that the following methods exist and provide interaction through the API to the database:

The following HTTP methods must be correctly used at least once: GET, POST, PUT, OPTIONS.

  - Retrieve a single record.
  - Retrieve multiple records.
  - Insert a single record.
  - Insert multiple records.
  - Update or replace a single record.
  - Update or replace multiple records.
  - Delete a single record.
  - Delete multiple records.
  - A method that utilises database projection on at least one occasion to return a single, or multiple results
  - A method that utilises an index when querying data

2.4 Create **Endpoints** to provide the functionality outlined in the project, ensuring the following are completed:

- The endpoints can communicate with the Database either directly or through a series of methods created in 2.3
- The endpoints contain appropriate validation to ensure that the following situations are handled:
  - A Bad Request is made to the endpoint, returning a 4xx status code
  - A Correctly formed request is made to the endpoint return a 2xx status code
  - Any Server-Side errors return a 5xx status code

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

2.5 Test each endpoint Created in 1.4, ensuring that each endpoint returns the correct response code where the user provides a bad request **and** where the user provides a correctly formatted request. Ensure that all endpoints that will be presented in the final application are tested.

Endpoint (And parameters)	**Method**	**Result of correct data / parameters provided**	**Result of incorrect data / parameters provided**
/api/note/{id}	*GET*	*200 – Success*	*400 – Validation Error*

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

2.6 Provide a detailed summary of the testing conducted, including the following detail:

- How did you determine what was going to be correct and incorrect data to provide to each endpoint
- Describe any unexpected results, and the process required to adjust the API Endpoint or underlying methods to allow for providing expected results.

2.7 Provide at least one screenshot of testing each endpoint, include the Request and Response where possible.

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

PART 3 - RESTful API Configuration

Task 1. Configure Cross Origin Resource Sharing

3.1 Enable and configure CORS for all GET, POST, PUT and DELETE endpoints in preparation for requesting resources from the API from an origin different to that of the API.

- At least one allowed origin should be configured, this could be https://www.test-cors.org/ or another domain as discussed with your assessor.
- Any number of allowed Headers can be configured.
- Provide a screenshot of the configuration required to allow CORS on your server.

3.2 Provide screenshots of the request made to at least one endpoint for each method that clearly demonstrates the correct configuration of CORS for the endpoint. Ensure that these requests are sent from an origin different to that of the API and that the screenshot shows clear evidence that the request was from a different origin

3.3 Configure your API to allow for Pre-Flight requests (if necessary).

3.4 For each of the methods previously configured to accept CORS requests, send a request to each endpoint that will trigger a pre-flight request and response. Provide a screenshot of at least 5 instances of this succeeding (one for each type of HTTP Method listed in 3.2)

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

Task 2. Evaluate and secure REST API

3.5 Describe the method chosen to secure the application according to business needs, including both Authentication and Authorisation methods selected.

3.6 Complete the table below for each endpoint in your API, ensuring that the Authentication and Authorisation outlined for each method below aligns with the Project Requirements.

ENDPOINT	HTTP METHOD	Authentication Required?	Authorisation required (Role or access level)
/api/note{id}	*GET*	*YES*	*Super-User*
Add rows as necessary			

3.7 Implement the authentication and authorisation method proposed earlier in this task.

3.8 Test each method that requires authentication and authorisation to ensure that the implemented methods align with the project requirements.

3.9 Provide a screenshot for each of the following situations that clearly shows the Request and the Response from the API

- Authentication Succeeding
- Authentication Failed (incorrect Authentication details provided)
- Authorisation Succeeding
- Authorisation Failed (incorrect Permission for action)

Test the method and provide screenshots as evidence.

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

Task 3. Document REST API

3.10 Compare and evaluate at least three (3) API documentation tools that are currently used in the industry and could be used within, or alongside your developed API

Although your manager and/or relevant personnel have decided to use a tool that implements the Open API Specification to define and document the API, you need to compare and evaluate 3 options. Ensure that at least one tool evaluated above implements the Open API Specification and is suitable for use with your project

3.11 Use a tool that implements the Open API Specification to document each REST API endpoint, ensuring that the following is described:

- Endpoint Purpose
- Required and optional parameters
- Expected responses from the endpoint.

3.12 Provide screenshots that demonstrate this documentation for each endpoint. This could be individual screenshots or an export of the API document submitted alongside your project

3.13 Provide a written description of how the documentation aligns with the project requirements, and the benefits provided with including API documentation.

<table>
<tr><td colspan="3">Part 2 and 3 – REST API Development SIGNOFF
Signing off on this document signifies that the RESTful API and Documentation presented complies with the Client’s Business requirements.</td></tr>
<tr><td colspan="2">Project Manager or relevant stakeholder
Signature:
Date:</td><td>Web Developer
Signature:
Date:</td></tr>
<tr><td colspan="3">Initial Review

(Feedback provided by Teacher)</td></tr>
<tr><td>APPROVAL</td><td>☐ Granted</td><td>☐ Not Granted</td></tr>
<tr><td colspan="3">Secondary Review (If adjustments are required in the Initial Review)

(Feedback provided by Teacher)</td></tr>
<tr><td>APPROVAL</td><td>☐ Granted</td><td>☐ Not Granted</td></tr>
</table>

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

PART 4 – Database Tasks

Some of the following tasks will require the connection of your API to a cloud hosted instance of MongoDB (MongoDB Atlas). This incurs no cost but can present connection difficulties through corporate firewalls.

Task 1 – Configuration

4.1 Verify the NoSQL database is persisting objects of different types by providing a before and after screenshot of the same document after restarting your NoSQL Instance. If the document has not been persisted across the restart, configure the instance to persist data and provide the steps required to fix the persistence.

- Provide screenshots before and after Instance restart, test and fix persistence if it is not working (for any data stores).
- Provide a screenshot that demonstrates that the Database Server is configured to persist data.

Propose at least two (2) triggers for the project and identify the corresponding events and actions to be performed. Create an email to your manager or relevant stakeholder seeking confirmation of the triggers before implementing them.

4.3 Using a **cloud hosted** instance of MongoDB (Atlas) create and configure the selected triggers, ensuring the functionality of these triggers is tested.

- Provide a screenshot of the code used to define the triggers and the results of testing each trigger.

4.4 Utilising a **cloud hosted**, or local instance of MongoDB, configure Authentication and Authorisation between the API and Database Server to ensure that the API connecting to the Database is only capable of reading from and writing to the database that is relevant to this project.

- Provide screenshots of this configuration process, including the results of attempting to access the server with correct and incorrect credentials (these can be through the API, or through MongoDB Compass using the same connection string as the API)
- Describe the roles created within Atlas (or another relevant DBMS) and the process required to create the roles and user accounts.

4.5 Determine and implement a time-to-live (TTL) or special single-field index on a field in a collection. The TTL index will automatically remove the document after a certain amount of time has passed or a specific clock time. **Refer to the scenario presented for the required TTL actions.**

4.6 Review the Encryption requirements of the provided scenario and provide evidence that the encryption requirements are met. This could be achieved through screenshots of configuration, or a detailed description of the mechanisms that are enabling the required encryption.

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

Task 2 - Partitioning

This task requires the creation of a 'Sharded Cluster' when using MongoDB. For detailed instructions, and a script that can be used to create this, see the supplementary assessment files or contact your teacher. This can also be achieved through multiple physical devices, or via Docker Instances.

4.7 Review the scenario provided and the data imported into your database and determine at least one (1) collection that could benefit from partitioning.

- Select an appropriate partition key for this collection, ensure that you provide clear justification as to why this partition key is suitable for the collection to be partitioned

4.8 Utilising simulated architecture representing a sharded cluster:

- Import the dataset used previously in this project.
- Configure the database and collection for partitioning.
- Implement the selected partition key, ensuring that the collection is partitioned.
- Provide screenshots that clearly show the collection name, the number of chunks, shards, and the distribution of data over these different shards.

4.9 Review the results of the partitioning completed in 4.4 and provide a clear description of a process that could be used to ensure a continued even distribution of data across partitions, or to achieve a better distribution across the partitions.

4.10 Review the completed project and document, ensuring that all of the required steps have been completed. Attend a brief review meeting with your Manager to review the completion of the Database Tasks using the Signoff section below.

<table>
<tr><td colspan="3">Part 4 – Database Tasks SIGNOFF
Signing off on this document signifies that the Final Database Task Documentation and Procedures presented comply with the Client's Business requirements.</td></tr>
<tr><td colspan="2">Project Manager or relevant stakeholder
Signature:
Date:</td><td>Web Developer
Signature:
Date:</td></tr>
<tr><td colspan="3">Initial Review

(Feedback provided by Teacher)</td></tr>
<tr><td>APPROVAL</td><td>☐ Granted</td><td>☐ Not Granted</td></tr>
<tr><td colspan="3">Secondary Review (If adjustments are required in the Initial Review)

(Feedback provided by Teacher)</td></tr>
<tr><td>APPROVAL</td><td>☐ Granted</td><td>☐ Not Granted</td></tr>
</table>

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

PART 5 – Contingency task and knowledge concepts related to this project

REFERENCING YOUR WORK

Part 5 requires you to carry out research and provide answers to a number of questions. Provide the answers in your own words. Plagiarism is a form of academic misconduct and will not be tolerated. Include references for all your sources using a formal referencing style such as APA or Harvard.

Q1 Contingency task: Assume that you are notified that some MongoDB documents are sometimes removed without any apparent operation being performed on that particular data store. Where would you start looking for the source of the problem?

Q2 In relation to the project completed in this portfolio, identify the **programming language** used to interact with the NoSQL and the **data interchange format language** or notation. Assess the performance and suitability of the language selected for the task.

Q3 In relation to the project completed in this portfolio:

- Outline the criteria that you used to **partition** a specific data store and the benefits achieved.
- What scalability method makes possible the **distribution of data across partitions** in NoSQL databases? Describe how the distribution process works.
- Explain how **sort keys** can be used in a partition to increase performance and outline their **functions and features** in a NoSQL implementation**.**

Q4 In relation to the project completed in this portfolio, review the requirements that needed to be reviewed before the implementation of a TTL index on a field, and the functionality provided by this TTL

Q5 In relation to the project completed in this portfolio:

- Examine how you implemented **transport encryptions**, **authentication**, and **authorisation** in the REST API. Clarify how the methods and **features** you used contributed to securing the application.

Q6 In relation to the project completed in this portfolio, identify the debugging and testing methodology used and delineate **2 techniques** used to **debug** and **test** the API and **2 techniques** used to **debug** and **test** the **Database** (consider triggers, indexes and Projections).

Q7 Identify and appraise the **datastore format** used for this portfolio project. Provide a brief description of each datastore format presented in the table below.

DATA FORMAT	FEATURES	WHEN TO USE	EXAMPLE PROVIDER
Key-value			
Document-based			
Column-based			
Graph-based			

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.

Q8 For each data format listed below, explain the range of values allowed in each datatype. For this exercise, provide a screenshot of a single document created in a NoSQL technology of your choice that demonstrates each of the data types below.

- numeric
- string
- Boolean
- complex
- date time

Q9 For each HTTP method listed below, explain the features and uses cases of each method.

(Consider how each method differs from one another).

- GET
- POST
- PUT
- OPTIONS

End of Assessment

© TAFE Queensland. Uncontrolled copy if printed. Refer to TAFE Queensland intranet for current version of document.