|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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* | Data Cluster | | | |
| Assessment Name | Website Information Architecture Portfolio | Assessment Task No. | | 2 of 2 |
| Assessment Due Date | Week 10 | Date submitted | | / / |
| Assessor Name | REST API and NoSQL Portfolio | | | |
| **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 | / / |

|  |  |
| --- | --- |
| **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 four (4) parts:   * PART 1 – Selecting and preparing the NoSQL database * PART 2 – RESTful API selection, creation, and testing * PART 3 – RESTful API and MongoDB Integration * PART 4 – 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 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 * 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   **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. Demonstrated ability to:   * 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   Refer to the marking criteria for specific details:  ICTPRG553\_554\_AT2\_MC\_TQM\_V1  **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.  **Reasonable Adjustments:**  Reasonable adjustments are available to students for a variety of reasons, including: disability, language, literacy and numeracy (LLN) problems or extenuating circumstances. Talk to your teacher, counsellor, or disability officer if you require extra support or an extension based on the conditions identified.  **Number of Attempts:**  You will receive up to two (2) attempts at this assessment task. Should your 1st attempt be unsatisfactory (U), your teacher will provide feedback and discuss the relevant sections / questions with you and will arrange a due date for the submission of your 2nd attempt. If your 2nd submission is unsatisfactory (U), or you fail to submit a 2nd attempt, you will receive an overall unsatisfactory result for this assessment task. Only one re-assessment attempt may be granted for each assessment task.  ***For more information, refer to the Student Rules.***  **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. |
| **Submission details** | **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  TAFE Queensland Learning Management System (Connect)  **Accessing Connect:**  Connect URL: https://connect.tafeqld.edu.au/d2l/login  Username: 9 digit student number  Password: <your password>  For password reset 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 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 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 * 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   **Level of Assistance Permitted:**  Teachers and tutors should be available in class, and accessible by email for students working from home. Staff cannot directly show students answers but support and guide them to complete tasks individually. Students with disability will receive reasonable adjustments.  **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).  **Contingencies:**  Reasonable adjustment is available to students for a variety of reasons, including: disability, language, literacy and numeracy (LLN) problems or extenuating circumstances.  **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.  If you have any question or need help regarding this assessment item please contact your teacher/tutor through email or during face-to-face sessions. |

# 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, to control access to actions throughout the application. In particular, the client would like a teacher account that can modify and delete all database entries, and student accounts that can query the database for information.

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

# 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: 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.

# 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.

Be aware that not all raw data may be required for the project. Make sure that you understand the project requirements before discarding any data.

In addition to the data provided you need to create user accounts.

# MongoDB data storage, partition, and scalability

To ensure maximum efficiency of storage and retrieval across the distributed horizontal setup, an appropriate data field is to be selected and used to effectively partition the current and future data

# MongoDB and API settings and data constraints

The following settings and constraints must be included:

* Data **persistence** is required.
* **Indexing** - Based on the resulting schema, determine indexing needs to suit the scenario presented and create a single field index in a collection to optimise data retrieval and either a multikey index or a compound index to optimise data retrieval.
* TTL - Ability to automatically remove documents if the following condition occurs.
  + A user has not logged in for 30 days
* Triggers – Any two of the potential triggers below need to be created. (other triggers may be viable if discussed with your teacher during assessment)
  + Remove documents on edit or creation that contain invalid weather readings:
    - the humidity is greater than 100%
    - the temperature is greater than 60 degrees Celcius
    - the temperature is less than -50 degrees Celcius
  + Update the location value of a document in the dataset if the Longitude and Latitude values are updated
  + 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
* 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
  + 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 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:

1. Insert a new reading for a weather station (single)
   1. 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
2. Insert a new user (single)
   1. Allow Administrators to create a new user account with a specified level of access
3. Insert multiple sensor readings for single station
   1. To allow for batch inserts every day, rather than one-by-one as the data is recorded
4. 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 (single / projection)
   1. To allow students to retrieve data that might provide insights into changes in weather patterns based on temperature fluctuations
5. Find the temperature, atmospheric pressure, radiation and precipitation recorded by a specific station at a given date and time (hour) (multiple / projection)
6. 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(multiple / projection)
   1. To allow students to retrieve data that might provide insights into changes in weather patterns based on temperature fluctuations
7. Create a query that includes an index key
   1. provide efficient querying by ensuring that an index is included for large or complex queries
8. Delete a user by Id (single)
   1. Remove a single user account - revoking access to Authorised actions
9. Delete multiple users that have not logged in for more than 30 days (multiple)
   1. Allow Administrators to remove inactive users that are not also Administrators
10. Update a specified entries’ precipitation value to a specific value
    1. To allow Admin users to correct errors in the dataset
11. Update access level for at least two users in the same query, based on a date range in which the users were created (multiple)
    1. To Allow Administrators to upgrade or downgrade multiple users that were created in batches (upgrading or downgrading an entire group of students)

**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 |  |  |  |  |

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

**Task 2. Storage requirements and partitions**

1.5 For the NoSQL interface selected in the previous task:

a) 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

b) Review the business needs presented, and select the most appropriate data (formats) types for the NoSQL datastore for this project.

c) Source the data from the dataset provided and populate the datastore.

d) Create the necessary document/collection schema in MongoDB for the scenario presented. Document the database structure and its notation, e.g. JSON. It can be presented as a combination of text and graphics.

e) Identify at least one (1) collection that could benefit from partitioning. Determine the partition key and the sort key to control the order in which the data will be stored in the partition. Document your solution.

f) Create the partition ensuring that the partition key and the sort key match the business requirements. Provide screenshots as evidence.

g) Review the partition created above and outline a strategy to optimise data and achieve an effective distribution of storage across the partition.

**Task 3. Indexing and TTL in NoSQL**

1.6 Calculate the average amount of data that may be input and output from the NoSQL database over a selected period of time, provide details around how this can be maintained as the dataset grows, and improved where necessary

1.7 Determine indexing needs to suit the scenario presented and complete the following:

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.

1.8 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**.

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

|  |  |
| --- | --- |
| **MongoDB database SIGNOFF**  Signing off on this document signifies that the **MongoDB database setup** presented **complies** with the Client’s Business **requirements.** | |
| Project Manager or relevant stakeholder  Signature:  Date: | Web Developer  Signature:  Date: |
| **Documentation NOT APPROVED**  Please provide feedback on the changes needed. | |
| **APPROVAL**  Granted  Not Granted | |

**PART 2 – RESTful API selection, creation, and testing**

In this part, you are selecting and setting up and testing the RESTful API that will integrate with the MongoDB datastore in PART 3.

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

**Task 1. Project requirements**

2.1 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:

1. Suitability to the scenario presented
2. Flexibility and portability
3. Scalability
4. Cacheability

**Task 2. Selecting REST API framework and IDE**

2.2 Review Web API frameworks available for your chosen programming language, select a framework that you feel is suitable and justify your selection by providing a detailed description of how this Framework will meet the needs described in 2.1.

**Task 3. Build the REST API: Endpoints and methods**

2.3 Develop a RESTful Web API that provides at least the endpoints required to meet the functionality requirements provided in the scenario, ensuring that an appropriate level of validation is implemented these endpoints.

The initial development of the endpoints does not require a connection to the database, but should be able to return different responses based on the results of validation of the provided HTTP Request

The endpoints created to meet the clients functionality requirements must also include at least one endpoint that meets each of the following criteria:

1. Return a single value, using the GET method
2. Return a collection of values, using the GET method
3. Add a value to a collection using the POST method
4. Update an existing document using the PUT method
5. Update an existing value within a document using the PATCH method
6. Remove a specified item from a collection using the DELETE method

Each developed endpoint should return a status code and message that describes the result of an invalid request, no record found, or a server error.

2.4 Test the methods/endpoints used in 2.3 and check that the response status codes obtained match the business requirements/needs.

If necessary, modify and re-test until all tests pass. Complete the table below and provide screenshots as evidence.

### Initial Tests:

|  |  |  |  |
| --- | --- | --- | --- |
| Endpoint (And parameters) | Method | Response Status Code and Message | Matches Requirements  Yes/No |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Add rows as necessary |  |  |  |

### Further Tests after corrections (Where a response did not match the requirements)

|  |  |  |  |
| --- | --- | --- | --- |
| Endpoint (And parameters) | Method | Response Status Code and Message | Matches Requirements  Yes/No |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Add rows as necessary |  |  |  |

**Task 4. Enable CORS**

2.5Enable and configure CORS for all GET, POST, PUT & DELETE methods in preparation for requesting resources from the API from an origin different to that of the API.

2.6 Provide screenshots of the request made to at least one endpoint for each method (at least 4 endpoints) 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

2.7 For each of the methods previously configured to accept CORS requests, send a request to each endpoint including a custom header in the request to trigger a pre-flight request and response. Provide a screenshot of at least 4 instances of this succeeding (one for each type of HTTP Method listed in 2.6)

If configuration was required to allow this to succeed, include a screenshot of that configuration alongside the request and response screenshots.

**Task 5. Evaluate and secure REST API**

2.8 Describe the method chosen to secure the application, as well as the individual endpoints that will require authentication and authorisation as per the business requirements provided.

|  |  |  |  |
| --- | --- | --- | --- |
| ENDPOINT NAME | HTTP METHOD | Authentication Required? | Authorisation required (Role or access level) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Add rows as necessary |  |  |  |

2.9 Implement the authentication and authorisation method proposed in the previous task.

Test the method and provide screenshots as evidence.

**Task 6. Documenting REST API endpoints**

* 1. Compare and evaluate at least three (3) API documentation tools that are currently used in the industry and may be relevant to documenting the developed API

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TOOLS | MAIN FEATURES | COST  Free/Paid | EASE OF USE | BEST FOR  Type of Project | OVERALL RATING  (L)1-5 (H) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  | Add rows as necessary |  |  |  |

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

* 1. 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 and expected responses from the endpoint.

Provide screenshots that demonstrate this documentation for each endpoint.

* 1. Validate documentation against endpoints, ensuring that each endpoint is documented, and that the documentation aligns with the endpoint’s purpose, parameters and responses. Provide a written description of how the documentation aligns with the project requirements. Submit this description and your screenshots to your manager or relevant person in the organisation for approval.

|  |  |
| --- | --- |
| **REST API Endpoints Documentation SIGNOFF**  Signing off on this document signifies that the **documentation** presented **complies** with the Client’s Business **requirements.** | |
| Project Manager or relevant stakeholder  Signature:  Date: | Web Developer  Signature:  Date: |
| **Documentation NOT APPROVED**  Please provide feedback on the changes needed. | |
| **APPROVAL**  Granted  Not Granted | |

**PART 3 – RESTful API and MongoDB Integration**

*In order to comply with Uptown IT documentation procedures, a template has been provided to document the activities for PART 3.*

**Task 1. API interaction with NoSQL**

3.1 Using the REST API designed, created and tested in PART 2, interact with the NoSQL database (MongoDB) for this project. Connect to datastore.

3.2 Perform the operations listed below. All the operations must be as per the business requirements **specified in the scenario presented**. If you completed some of these queries in 2.3, while testing the API, you can reuse them in this section.

For each operation, use the corresponding method and endpoint(s). Provide a screenshot of each request to an endpoint and the returned result.

1. Insert a new reading for a weather station (single)
2. Insert 5 new weather readings for single station
3. 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 (single / projection)
4. Find the temperature, atmospheric pressure, radiation and precipitation recorded by a specific station at a given date and time
5. 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
6. Send a request to an endpoint that utilises an index in the underlying query
7. Delete a single user by Id
8. Delete multiple users
9. Update a specified entries’ precipitation value to a specific value
10. Update access level for at least two users in the same query, based on a date range in which the users were created

**Task 2. Further queries or object interactions**

3.3 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, and if necessary configuration changes made to ensure persistence.

3.4 In accordance with **the scenario provided**:

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

|  |  |  |
| --- | --- | --- |
| Trigger | Event | Actions to be performed |
|  |  |  |
|  |  |  |

1. Modify triggers, if necessary, based on feedback received. Then, implement the triggers and test them. For each trigger, provide screenshots of the relevant data before and after each trigger is activated.

3.5 Review and confirm that the following features are working according to business access requirements. Modify and retest as necessary. Test each one of them and provide screenshots.

a) Data encryption

b) Database authentication

c) Database authorisation

3.6 Use the Uptown IT REST API and NoSQL template to finalise and document the project. Using the template ensures compliance with Uptown IT organisational procedures for documentation.

3.7 Present the completed REST API and MongoDB integration to your manager or relevant person in the organisation for approval and signoff.

|  |  |
| --- | --- |
| **REST API MongoDB integration SIGNOFF**  Signing off on this document signifies that the **REST API MongoDB integration** **complies** with the Client’s Business **requirements.** | |
| Project Manager or relevant stakeholder  Signature:  Date: | Web Developer  Signature:  Date: |
| **Documentation NOT APPROVED**  Please provide feedback on the changes needed. | |
| **APPROVAL**  Granted  Not Granted | |

**PART 4 – Contingency task and knowledge concepts related to this project**

**REFERENCING YOUR WORK**

*Part 4 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:

1. Outline at least four (4) **benefits** achieved from the implementation of a NoSQL database that would not have been possible in a relational database implementation.
2. In terms of functionality, outline two (2) **functions** that were not possible in a relational database.

Q3 In relation to the project completed in this portfolio, examine the features of **scaling out** and **scaling up** approaches, and explain two (2) **features** of **scaling out** that are essential to this project.

Q4 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.

Q5 In relation to the project completed in this portfolio:

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

Q6 In relation to the project completed in this portfolio, appraise how you used **TTL** and discuss the **features** and **functions** implemented. Could you have taken advantage of other functions and features? Outline them.

Q7 In relation to the project completed in this portfolio:

1. Explain the method and process you used to implement **authentication** and **authorisation** procedures in the NoSQL database and justify the number of **access** levels created.
2. 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.

Q8 In relation to the project completed in this portfolio, identify the debugging and testing methodology used and delineate four (4) **techniques** used to **debug** and **test** the API. For each technique evaluate its effectiveness in this specific project.

Q9 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 |
| **Key-value** |  |  |  |
| **Document-based** |  |  |  |
| **Column-based** |  |  |  |
| **Graph-based** |  |  |  |

Q10 For each data format listed below, explain the range of values allowed and provide an example of the code/notation used to declare them. For the purpose of 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

1. numeric
2. string
3. Boolean
4. complex
5. date time

**End of Assessment**