Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**ASSESSMENT**

**Task 2**

**ICTPRG554 Manage data persistence using noSQL data stores**

**This unit forms part of the following qualification:**

**ICT50120 Diploma of Information Technology**

**Assessment Task 2: Project Portfolio**

## **Pre-assessment checklist**

## **Purpose**

The pre-assessment checklist helps students determine if they are ready for assessment. The Trainer/Assessor must review the checklist with the student before the student attempts the assessment task. If any items of the checklist are incomplete or not clear to the student, the Trainer/Assessor must provide relevant information to the student to ensure they understand the requirements of the assessment task. The student must ensure they are ready for the assessment task before undertaking it.

## 

## **Information for students**

* Please make sure you have completed the necessary prior learning before attempting this assessment.
* Please make sure your Trainer/Assessor has clearly explained the assessment process and tasks to be completed.
* Please make sure you understand what evidence is required to be collected and how.
* Please make sure you know your rights and the complaints and appeal process.
* Please make sure you discuss any special needs or reasonable adjustments to be considered during the assessment (refer to the Reasonable Adjustments Strategy Matrix and negotiate these with your Trainer/Assessor).
* Please make sure that you have access to a computer and the internet (if you prefer to type the answers).
* Please ensure that you have all the required resources needed to complete this Unit Assessment Task (UAT).
* Due date of this assessment task is according to your timetable.
* In exceptional (compelling and compassionate) circumstances, an extension to submit an assessment can be granted by the Trainer/Assessor.
* Evidence of the compelling and compassionate circumstances must be provided together with your request for an extension to submit your assessment work.
* Request for an extension to submit your assessment work must be made before the due date of this assessment task.

## **Reasonable adjustments**

* If student has requested a reasonable adjustment, then complete the reasonable adjustment form included in the unit information pack.

## **Student declaration**

* I confirm that Trainer/Assessor has provided all the information related to the assessment task as included in the information for student section and I am ready for the assessment.

|  |  |  |  |
| --- | --- | --- | --- |
| Student signature | Marcelo Augusto Bueno Dos Santos | Date | **05/06/2024** |

**Assessment task instructions**

**Assessment type:**

* Unit Project

**Instructions provided to the student:**

**Assessment task description:**

* This is the second (2) unit assessment task that student has to successfully complete to be deemed competent in this unit of competency.
* The student is required to complete six activities.
  + Activity 1: Prepare report
  + Activity 2: Practical task in computer lab
  + Activity 3: Indexes configuration
  + Activity 4: Queries update
  + Activity 5: Updating and deleting objects
  + Activity 6: Testing event trigger
* The student must attempt all criteria to the required level, e.g., Assessment criteria mentioned in the performance checklist to be deemed satisfactory in this task.

**Applicable conditions:**

* Activity 1 and 2 is untimed and conducted as an open book test (this means the student can refer to textbooks or other learner materials during the test).
* Activity 3, 4, 5 and 6 are timed.
* The student must complete the task independently.
* No marks or grades are allocated for this assessment task. The outcome of the task will be Satisfactory or Not Satisfactory.
* Trainer /Assessor must assess student’s practical skills, technique, and knowledge as he/she completes this assessment task.
* The Trainer/Assessor may ask the student relevant questions on this assessment task to ensure that this is his/her own work.

**Resubmissions and reattempts:**

* Where a student’s performance is deemed not satisfactory after the first attempt, a resubmission attempt will be allowed. Assessor must note any such submissions.
* Student may speak to their Trainer/Assessor if the student has any difficulty in completing this task and requires a reasonable adjustment (e.g., can be given as an oral assessment).
* For more information, please refer to the RTO Student Handbook.

**Location:**

* This assessment task may be completed in (tick the relevant box):

|  |  |
| --- | --- |
| Learning Management System | Classroom |
| Simulated learning environment | Workplace |
| Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |

**Purpose of the assessment:**

* Purpose of this project is to check skills and knowledge related to performance criteria and performance evidence of the unit.

**General Instructions for attempting the project:**

* Student must complete all the activities of this assessment task.
* Answers must demonstrate an understanding and application of relevant concepts, critical thinking, and good writing skills.
* The student must complete all the provided templates (if any).

**Resources required to complete the assessment task:**

* Access to learner guide and other learning materials.
* Computer
* Internet
* MS Word
* Printer or e-printer
* Unit Assessment Task
* External hardware devices
* Internal hardware devices
* Application program softwares
* Operating system
* NoSQl database
* Internet

**Scenario**

Future IT is an IT company that provides different IT services such as application development, technical issues resolution, software support, Internet services, database services, and others. It operates all across Australia with branches in major cities such as Melbourne, Sydney, Perth, and Brisbane.

It has around 133,240 clients and is providing services to them for the last five years. The company has been consistent with providing services and is giving tough competition in the industry.

Future IT manages data for the different organisation as a third party and provide protection to the data. It helps organisation to save their databases for internal purposes and also protects them if any data loss occurs, then Future IT is responsible for that.

Future IT is using the NOSQL database and specifically MongoDB to manage the data of different organisations. The reason for using a noSQL database is given below:

* Provide fast performance and horizontal scalability
* Provide big capability
* Easy to implement as compared to RDBMS
* Different developer languages and platforms are supported

Future IT is also using horizontal and vertical scaling to manage the database. Horizontal and vertical databases provide the following benefits:

Horizontal scaling benefits

* Easy to upgrade
* Easy to run fault-tolerance
* Easily scalable

Vertical scaling benefits

* Reduce software costs
* Easy to implement
* Less power consumption

Future IT is aware that they can use different NoSQL technologies, which are:

* Document databases
* Key-value stores
* Column-oriented databases
* Graph databases

To store the data, Future IT can use the following different types of data stores:

* Document databases
* Key-value stores
* Column-oriented databases
* Graph databases

You have been working as a database manager for Future IT. You need to go through the scenario and complete the following activities.

|  |
| --- |
| **Workplace health and safety (WHS) processes and procedures** |
| This policy:   * shows the commitment of management and workers to health and safety * aims to remove or reduce the risks to the health, safety and welfare of all workers, contractors and visitors, and anyone else who may be affected by our business operations * aims to ensure all work activities are done safely. |
| Management (the supervisor and/or manager) is responsible for providing and maintaining:   * a safe working environment * safe systems of work * plant and substances in a safe condition * facilities for the welfare of all workers * any information, instruction, training and supervision needed to make sure that all workers are safe from injury and risks to their health |
| Employees are responsible for:   * ensuring their own personal health and safety, and that of others in the workplace * complying with any reasonable directions (such as safe work procedures, wearing personal protective equipment) given by management for health and safety   **Monitor**   * Incorrect positioning of the screen and source documents can result in awkward postures. Adjust the monitor and source documents so that your neck is in a neutral and relaxed position. * Centre the monitor directly in front of you above your keyboard. * Set the screen distance to approximately an arm’s length away (350mm-750mm) * Set the height of the monitor so that the top third of the screen is at eye level, and the bottom of the screen can be read without having to incline your head. (If you wear bifocals, lower the monitor to a comfortable reading level.) * Reduce glare by careful positioning of the screen. * Place screen at right angles (90 degrees) to windows * Adjust curtains or blinds as needed * Adjust the vertical screen angle and screen controls to minimise glare from overhead lights * Other techniques to reduce glare include the use of glare filters, light filters, or secondary task lights * Position source documents directly in front of you, between the monitor and the keyboard using a stand.   **Keyboard**   * The keyboard should be detachable from the screen * Place the keyboard at the front edge of the desk so that it is within range of your hands without straightening your elbows. * Place the keyboard in a position that allows your forearms to be at a 90-degree angle and the wrists to be straight (hand in line with the forearm) * Determine what section of the board you use most frequently and readjust the keyboard so that section is centred with your body. * The keyboard angle should sit flat on the desk to help maintain a straight wrist position. * Wrist rests can help to maintain neutral postures and pad hard surfaces. However, the wrist rest should only be used to rest the palms of the hands between keystrokes. Resting on the wrist rest while typing is not recommended. Wrist rests are primarily for periods of rest between bouts of typing.   **Mouse**   * Position the mouse as close as possible to the keyboard. Avoid reaching over the keyboard to use your mouse. * Work with your shoulder relaxed. * Keep your arm close by your side. * Keep your wrist in a straight or neutral position. * Lightly supporting your forearm on your armrest or desk can help you keep your shoulder relaxed. * Do not twist or move your wrist from side to side, or up and down when working.  Keep your hand relaxed. Move the mouse with the whole arm, using the elbow as a pivot point, keeping your wrist in a straight neutral position. Do not hold your pointing device with a tight grip. * When not using the pointing device, let it go. * Keep your fingers relaxed on the buttons without inadvertently pressing them. * Keep your thumb relaxed. Do not keep your thumb in a bent position when using the mouse. * Consider alternating between the left and right hand to reduce the workload and strain on the one hand. Use caution when switching hands, and make sure the device is made for the hand you are using. Give yourself time to get used to the change.   **Slips, trips and falls**  Electric cables across passageways, mats, unexpected steps in poorly-lit areas, slippery surfaces (eg marble floors in foyers) and storage that requires workers to use step ladders or reach above shoulder height can all raise the risk of people slipping, tripping and falling. Serious injuries can be sustained as a result, so the office should be checked for such hazards, and any risks of slips, trips and falls should be controlled.  **Thermal comfort and air quality**  A range of issues may need to be addressed, including temperature control, air-conditioning, air quality including ozone levels, and the risk of conditions such as Legionnaire’s disease and ‘sick building syndrome’.  **Electrical safety**  Electrical appliances, power boards and power cords should be checked and maintained in good conditions and without overloading, as needed. |

You need to complete the following activities to complete this task.

**Activity 1: noSQL options review and selection**

|  |  |
| --- | --- |
| **Your task** | In this activity, you need to prepare a report on noSQL options and provide information about horizontal and vertical scaling, noSQL technologies and vendor guidelines for using noSQL technology.  In the report you need to cover the following points:   * Business requirements from noSQL * Need of noSQL * Horizontal and vertical scaling comparison * Benefits of horizontal and vertical scaling * noSQL technologies * Comparison between noSQL technologies and traditional relational data models (SQL) * Selection of specific noSQL technology   Once the report is completed, you need to send an email to the manager and vendor to confirm the use of the noSQL technology according to business requirements and confirm the benefits of horizontal and vertical scaling. |
| **You will be required to complete and/or attach.** | Complete the following Template 1 and submit it to your Trainer/ Assessor. |

|  |
| --- |
| **Template 1: Review and select noSQL technology** |
| **Introduction**  ⦁ Future IT is an IT company that offers a variety of services including application development, technical issue resolution, software support, and more. Operating across Australia, it has branches in key cities like Melbourne, Sydney, Perth, and Brisbane. With approximately 133,240 clients served over the past five years, the company has maintained consistency and competitiveness in the industry. They need a new database, and this database must have flexibility, accessibility, security, scalable and they want to upgrade the previous database, the previous data base needs to have more security, new features and the maintenance cost should be less. |
| Future IT is using the NOSQL database and specifically MongoDB to manage the data of different organisations. The reason for using a noSQL database is given below:   * Provide fast performance and horizontal scalability * Provide big capability * Easy to implement as compared to RDBMS * Different developer languages and platforms are supported |
| Future IT is also using horizontal and vertical scaling to manage the database. Horizontal and vertical databases provide the following benefits:  Horizontal scaling benefits   * Easy to upgrade * Easy to run fault-tolerance * Easily scalable   Vertical scaling benefits   * Reduce software costs * Easy to implement * Less power consumption |
| **Horizontal Scaling (Scaling Out)** **Benefits:**   1. **Increased Capacity**: Horizontal scaling involves adding more machines (nodes) to handle increased load. This can significantly increase the overall capacity of the database system, enabling it to process more requests simultaneously. 2. **Fault Tolerance**: By distributing the data across multiple nodes, horizontal scaling can enhance fault tolerance. If one node fails, others can take over, minimizing downtime. 3. **Cost-Effectiveness**: In many cases, adding commodity hardware for horizontal scaling can be more cost-effective than investing in more powerful single machines. 4. **Flexibility**: Horizontal scaling allows for incremental growth. Businesses can add more nodes as needed without disrupting existing services.  **Vertical Scaling (Scaling Up)** **Benefits:**   1. **Simplicity**: Vertical scaling involves upgrading the existing hardware (e.g., adding more CPU, RAM, or storage). This approach is simpler as it doesn’t require changes to the application logic or data distribution methods. 2. **Performance**: Upgrading to more powerful hardware can provide significant performance improvements, making it suitable for applications requiring high performance with minimal latency. 3. **Consistency**: Vertical scaling maintains a single database instance, ensuring consistency and reducing the complexity of managing distributed data. 4. **Compatibility**: Many legacy systems and applications are designed to run on a single server. Vertical scaling can enhance their performance without the need for significant re-engineering. |
| To store the data, Future IT can use the following different types of data stores:   * Document databases * Key-value stores * Column-oriented databases * Graph databases |
| Comparison between noSQL technologies and traditional relational data models   |  |  | | --- | --- | | **noSQl** | **Traditional relational data models (SQL)** | | **Characteristics:**  **Scalability**: NoSQL databases are designed to scale out horizontally, making them well-suited for handling large volumes of unstructured or semi-structured data across distributed systems.  **Flexibility**: Schemeless design allows for dynamic changes to data structures without requiring a predefined schema. This is ideal for applications with evolving data models.  **Variety**: NoSQL includes several types of databases such as document stores (e.g., MongoDB), key-value stores (e.g., Redis), column-family stores (e.g., Cassandra), and graph databases (e.g., Neo4j), each optimized for specific use cases.  **Performance**: Optimized for specific data access patterns, NoSQL databases can deliver high performance for read and write operations, particularly in big data and real-time web applications. | **Characteristics:**  **ACID Compliance**: Relational databases (RDBMS) ensure atomicity, consistency, isolation, and durability, making them reliable for transactions requiring strong consistency and integrity.  **Structured Data**: They use a predefined schema and structured query language (SQL) for defining and manipulating data, making them suitable for applications with well-defined data relationships and schemas.  **Joins and Complex Queries**: RDBMS excel in handling complex queries involving multiple tables through joins, which are crucial for transactional systems like banking and ERP.  **Mature Ecosystem**: Relational databases have been around for decades, with a mature ecosystem of tools, support, and established best practices. | |
| MongoDB |

|  |
| --- |
| **Email to the manager and vendor** |
| Subject: Proposal to Implement MongoDB in Our Company    Dear Jonatas and Rajesh,    I hope this message finds you well. As we continue to seek ways to enhance our data management capabilities, I propose the implementation of MongoDB within our company.    MongoDB offers significant advantages, including flexible schema design, horizontal scalability, and superior performance in handling large volumes of unstructured data. These features align perfectly with our needs for dynamic data structures and the ability to scale out efficiently.    By transitioning to MongoDB, we can improve our data processing speed and reliability, thereby enhancing our overall operational efficiency. Additionally, MongoDB's robust community support and comprehensive documentation will facilitate a smooth integration and maintenance process.    I recommend scheduling a meeting to discuss this proposal further and outline the implementation plan. Your input and collaboration will be invaluable in ensuring a successful transition.    Thank you for considering this proposal. I look forward to your feedback.    Best regards,    Marcelo Augusto Bueno Dos Santos  database manager  Marcelo.augusto9999@hotmail.com |

**Activity 2: Designing and creating a datastore**

**This activity is in continuation with the previous activity.**

|  |  |
| --- | --- |
| **Your task** | In the previous activity, you have selected the noSQL technology which meets the organisational requirements. In this activity, you need to prepare a report providing information about the data storage requirements and how data stores will be created for the noSQL database. You also need to provide information about the different noSQL data stores. You also need to create a partition key and distribute the storage across partitions.  In the report you need to cover the following points:   * Data storage requirements for noSQL * Design data store for noSQL * Types of noSQL data stores * Select noSQL datastore * Create partition key * Highlight storage place for the data items in the partition key * Check the distribution of the storage across partition key * Discuss partition key and its requirements   Once the report is completed, you need to send an email to select the noSQL data store in which the partition key will be created, and storage will be assigned. |
| **You will be required to complete and/or attach.** | Complete the following Template 2 and submit it to your Trainer/ Assessor. |

|  |
| --- |
| **Template 2: Datastores information** |
| NoSQL databases have unique data storage requirements due to their flexibility and scalability. Unlike traditional relational databases, NoSQL databases can handle unstructured and semi-structured data, which necessitates a storage system that supports varied data formats such as JSON, BSON, or XML. They often require high-capacity storage solutions to manage large volumes of data distributed across multiple nodes, facilitating horizontal scaling.    Data storage systems for NoSQL must ensure low latency and high throughput to handle real-time data processing needs. Additionally, they should provide robust data replication and backup mechanisms to ensure data availability and fault tolerance. As NoSQL databases prioritize availability and partition tolerance, their storage systems should also support eventual consistency models, allowing for distributed data synchronization over time. Thus, scalable, flexible, and reliable storage infrastructure is critical for effectively leveraging NoSQL databases. |
| Designing a data store for NoSQL involves selecting the appropriate database type based on application requirements. For example:  1. Document Store (e.g., MongoDB): Ideal for applications needing flexible schema and JSON-like document storage. Suitable for content management systems and user profiles.  2. Key-Value Store (e.g., Redis): Best for caching and session management, where fast read/write operations are essential.  3. Column Store (e.g., Cassandra): Excellent for handling large-scale, time-series data and analytical queries.  4. Graph Store (e.g., Neo4j): Optimal for applications with complex relationships, like social networks and recommendation engines.  Ensure the data store supports horizontal scaling, high availability, and data partitioning to meet the demands of modern, scalable applications. |
| NoSQL data stores come in four primary types, each optimized for different use cases:  **Document Stores** (e.g., MongoDB, CouchDB): Store data as JSON-like documents, providing flexibility with dynamic schemas. Ideal for content management and user data storage.  **Key-Value Stores** (e.g., Redis, DynamoDB): Use simple key-value pairs, making them highly performant for caching, session management, and real-time analytics.  **Column-Family Stores** (e.g., Cassandra, HBase): Organize data in columns rather than rows, suitable for handling large volumes of data and write-heavy workloads, such as log data and time-series data.  **Graph Databases** (e.g., Neo4j, OrientDB): Designed for storing and querying data with complex relationships, making them perfect for social networks, recommendation systems, and fraud detection. |
| **MongoDB**: Ideal for applications needing flexible, JSON-like document storage with dynamic schemas. It's great for content management systems, e-commerce platforms, and user profiles. |
| * **Enable Sharding on the Database:** First, you need to enable sharding for the database you want to shard.   sh.enableSharding("Future\_IT")   * **Create a Shard Key for the Collection:** Choose a shard key and shard the collection. The shard key is a field that MongoDB uses to distribute the documents.   sh.shardCollection("Future\_IT.Future\_IT\_employees, { "userId": 1 })  In this example, the userId field is chosen as the shard key. |
| Check the distribution of the storage across partition key |
| A partition key is a critical concept in distributed databases like MongoDB, used for horizontal scaling and data distribution across multiple servers or partitions. It determines how data is distributed and stored across these partitions. The choice of partition key is crucial as it directly impacts the performance, scalability, and efficiency of the database.    Several requirements guide the selection of a suitable partition key. Firstly, it should exhibit high cardinality, ensuring a wide range of values to distribute data evenly across partitions. Additionally, the partition key should exhibit uniformity in data access patterns to prevent hot spots and ensure balanced load distribution.    Moreover, the partition key must be immutable or relatively stable to avoid frequent data movements between partitions, which can degrade performance. Lastly, it should align with the application's querying and data access patterns to facilitate efficient data retrieval and manipulation. By meeting these requirements, a well-chosen partition key can optimize database performance and scalability in distributed environments. |

|  |
| --- |
| **Email to the manager and vendor** |
| Subject: Implementation of MongoDB NoSQL Data Store - Future\_IT Project    Dear Jonatas and Rajesh,    I hope this email finds you well. I am writing to inform you about an exciting development in our IT infrastructure project at Future\_IT. After careful consideration and evaluation, we have decided to implement MongoDB as our NoSQL data store solution.    MongoDB offers numerous advantages, including flexibility, scalability, and ease of use, making it an ideal choice for our project's requirements. With MongoDB, we anticipate significant improvements in data management and processing capabilities, leading to enhanced performance and efficiency.    As part of this implementation, I would like to highlight that we will be creating a collection named "Future\_IT" within the MongoDB database. Furthermore, a partition key will be carefully chosen to ensure optimal data distribution and storage across partitions. This key will play a crucial role in maintaining balanced load distribution and maximizing resource utilization.    Additionally, storage allocation will be strategically managed to accommodate the growing data needs of our project while ensuring cost-effectiveness and scalability.    We believe that this transition to MongoDB will greatly benefit our project and align with our long-term strategic goals. We appreciate your support and cooperation throughout this process and look forward to achieving success together.    Should you have any questions or concerns regarding this implementation, please feel free to reach out to me. Thank you for your attention to this matter.    Best regards,    Marcelo Augusto Bueno Dos Santos  database manager  Marcelo.augusto9999@hotmail.com |

**Activity 3: Indexes configuration**

**This activity is in continuation with the previous activity.**

|  |  |
| --- | --- |
| **Your task.** | In this activity, you need to select the sort key which will be implemented on the noSQL data stores and configure read and write throughputs as per business requirements. You also need to create and configure at least two indexes which will be used to optimise the queries used to retrieve the data. You also need to optimise data queries and configure time-to-live on the data objects.  In this activity, you need to complete the following tasks:   * Determine different sort key * Select sort key to meet business requirements * Calculate read and write throughputs of the datastore * Configure read and write through-puts * Create and configure at least two indexes * Optimise data queries and retrievals for indexes * Configure TTL on data objects   You need to perform this activity in a simulated environment and keep a record of the steps which you have performed by providing screenshots. **The information needs to be submitted to the trainer in a word file under the name StudentID\_UnitCode\_activity3.**  You also need to prepare a small report containing the information about different sort keys, requirements of the business from sort key, how to optimise queries using indexes. |
| **You will be required to complete.** | You need to complete the template provided and perform the tasks.  Your trainer will observe your performance and complete the following performance criteria checklist. |
| **Timeframe** | You need to complete this activity in 1-2 hours, and your trainer will provide you additional time to complete the template. |

**Activity 4: Updating queries**

**This activity is in continuation with the previous activity.**

|  |  |
| --- | --- |
| **Your task.** | In this activity, you need to update the queries for the noSQl database. You need to select API for the noSQL. You need to connect the API with the noSQL data store to perform the queries on the database.  In this activity, you need to perform the following tasks:   * Select API clients for interacting with noSQL * Connect API client to noSQL datastore * Insert single data to noSQL datastore * Insert multiple items with a single query * Select the single object with the help of a query * Retrieve multiple objects using the query * Use query against the index * Use query to select multiple attributes * Use query to obtain project results |
| **Database requirements and entries** | To perform the different queries in MongoDB, students need to create an entry for user name, user ide, contact details and address. Using the queries, the student needs to put the following information in the database.   |  |  |  |  | | --- | --- | --- | --- | | **User id** | **User name** | **Contact details** | **Country of birth** | | 1 | Alex | 123456789 | England | | 2 | Sally | 987456321 | Australia | | 3 | Rohan | 153264789 | India |   The student needs to enter the details of Alex as a single entry using the appropriate query and then enter the multiple data, which is Sally and Rohan, in the database using the queries.  Once the information is entered into the database, the student needs to retrieve the information and select different data. |
| **You will be required to complete.** | **You need to perform the tasks given in the activity, and your trainer will observe your performance and complete the following performance criteria checklist.** |
| **Timeframe** | You need to complete this activity in 1-2 hours, and your trainer will provide you additional time to complete the template. |

**Activity 5: Activity 5: Updating and deleting objects in data stores**

**This activity is in continuation with the previous activity.**

|  |  |
| --- | --- |
| **Your task.** | In this activity, you need to use different queries to perform specific functions on the noSQL database.  In this activity, you need to perform the following task using different queries:   * Delete a single object from the database * Delete multiple objects from the database * Update single object in the database * Update multiple objects in the database * Persist objects in data stores * Configure change event trigger and notification * Confirm the configuration of change event trigger and notification |
| **Database requirements and entries** | For this activity, the student needs to delete the contact of Alex as a single object. After deleting contact details, the student needs to delete all the details of Alex as multiple objects.  After deleting this information, the student needs to update the contact details of Sally to 456321789. And after that student needs to update the user id for both Sally and Rohan. |
| **You will be required to complete.** | **You need to perform the tasks given in the activity, and your trainer will observe your performance and complete the following performance criteria checklist.** |
| **Timeframe** | You need to complete this activity in 1-2 hours, and your trainer will provide you additional time to complete the template. |

**Activity 6: Testing event trigger and data persistence**

**This activity is in continuation with the previous activity.**

|  |  |
| --- | --- |
| **Your task.** | In this activity, you need to test the configuration of the event and fix the changes if required. You also need to test the data persistence and check if the data is encrypted, authorised and authenticated. You also need to document the final results ad changes made in the event trigger and data persistence.  In the activity, you need to perform the following tasks:   * Test the event responses and triggers notifications * Fix the event responses and triggers * Check if data is encrypted, authorised and authenticated * Test data persistence * Fix data persistence * Document and finalise the result |
| **You will be required to complete.** | **You need to take the screenshot of the results and submit it to the trainer by following the organisation guidelines. The screenshots need to be stored in a word file under the name StudentID\_Unitcode\_Activity6.**  **You need to perform the tasks given in the activity, and your trainer will observe your performance and complete the following performance criteria checklist.** |
| **Timeframe** | You need to complete this activity in 1-2 hours, and your trainer will provide you additional time to complete the template. |

