Assignment-4

Virtual Machine, Cloud DB, and REST API

1. Setup Azure Virtual Machine:

   - Created a Virtual Machine on Azure with appropriate specifications.
   - Configured network settings to allow inbound traffic on required ports for API access.

2. Install Node.js and Additional Tools:

   - Installed Node.js on the Virtual Machine to run the server-side code.
   - Installed necessary tools such as npm for package management.
   - Installed a text editor (e.g., Visual Studio Code) for code development.

3. Implementation of REST API with Express:

   - Developed a Node.js program using Express.js to implement the REST API.
   - Defined endpoints to handle CRUD operations for managing inventory.
   - Endpoints included:
     - POST /products to create a new product.
     - GET /products to retrieve all products with optional query parameters for filtering.
     - GET /products/:id to retrieve details of a specific product by ID.
     - PUT /products/:id to update details of an existing product.
     - DELETE /products/:id to delete a product by its ID.

4. Creation of SQL Database on Azure:

   - Created an SQL Database on Azure.
   - Configured a table named "products" with columns: name, price, and quantity.
   - Ensured appropriate indexing and constraints for efficient data management.

5. Securing Database Access:

- Configured network settings to allow access to the database only from the Virtual Machine.
- Implemented firewall rules to restrict access from unauthorized sources.
- Ensured encryption of data transmission between the Virtual Machine and the database.

6. Testing API Using Postman:

- Utilized Postman for testing the API endpoints.
- Made requests to the public IP address of the Virtual Machine.
- Validated responses for CRUD operations on the inventory:
  - Created new products and verified their addition to the database.
  - Retrieved all products and confirmed the accuracy of the returned data.
  - Retrieved individual products by ID and ensured correct details were returned.
  - Updated existing products and verified the changes reflected in the database.
  - Deleted products and confirmed their removal from the inventory.

Conclusion:

- Successfully implemented a RESTful API for inventory management using Node.js and Express.
- Ensured secure access to the SQL Database from the Virtual Machine only.
- Thoroughly tested the API using Postman, validating functionality and data integrity for CRUD operations.