### Project Report: Customer Management System (CMS)

#### Overview

The Customer Management System (CMS) has been successfully developed to meet all specified requirements. This system allows users to manage customer profiles through a web interface, focusing on clean design, efficient data handling, and robust security features.

#### Technologies Used

1. **Frontend:**
   * **Framework:** Next.js 14 (React)
   * **Features:**
     + User Interface for listing all customers
     + Forms for adding and editing customer details
     + Option to delete customers
     + Responsive design for both desktop and mobile
     + Client-side routing with Next.js
2. **Backend:**
   * **Framework:** ASP.NET 8 Web API
   * **API Endpoints:**
     + GET /customers - Retrieve all customers
     + POST /customers - Add a new customer
     + PUT /customers/{id} - Update an existing customer
     + DELETE /customers/{id} - Delete a customer
   * **Security:** JWT (JSON Web Tokens) for secure API access
3. **Database:**
   * **System:** SQL Server
   * **Schema:**
     + Customers Table:
       - CustomerId (Primary Key, Auto-Generated, Integer)
       - FirstName (String)
       - LastName (String)
       - Email (String, Unique)
       - Phone (String)
       - Address (String)
   * **Data Seeding:** SQL scripts replaced with a Seed class within the project to populate the database with a default user and sample customers.

#### Code Quality

The code follows best practices with meaningful variable names, comments on complex logic, and modular, reusable functions. The project adheres to Separation of Concerns (SoC).

#### Version Control

GitHub was used for version control, with a .gitignore file configured for Visual Studio and node\_modules. Commits were made incrementally with descriptive messages.

#### Security

The API is secured with JWT authentication, and server-side validation is implemented to prevent SQL Injection and XSS attacks.

#### Deliverables

1. **Source Code:** Available on GitHub.
2. **Database Script:** Replaced with a Seed class for initial data population.
3. **Setup Documentation:** README file included.
4. **Postman Collection:** Attached to the project for easy API testing.

#### Seeding

Instead of executing an SQL query like:

INSERT INTO Customers (FirstName, LastName, Email, Address, Phone)

VALUES

('bob', 'faraj', 'bob@bob.com', 'Dubai, UAE', '+971000000000'),

('hani', 'hamad', 'hani@hani.com', 'Sharjah, UAE', '+971000000000');

I used C# code to insert the data directly into the database using ORM tools like Entity Framework or through custom code. You can find this code in the SeedData method of the Seed class, which is called in Program.cs.

#### Conclusion

All requirements were successfully completed, and the project is now available on GitHub for review. The system is fully functional, secure, and ready for deployment.

**Note:** Please review the source code, unit tests, README, and Postman collection via the provided GitHub <https://github.com/raedalifarhan/customer-management-system.git>.