

1 System Overview and Configuration

1.1 System Requirements

1. Frontend

1. HTML:

- I have used HTML5 to structure the web page with a grid/table to display tasks and a dialog for task creation/editing.
- I have included the “Add”, “Edit”, “Delete” buttons at the top of the grid for creating tasks and performing actions like edit and delete.

2. CSS:

- I have used CSS3 to style the page to make it visually appealing and responsive, and to ensure the dialog is cantered and styled appropriately.

3. JavaScript:

- I have used JavaScript to handle task creation, editing, and deletion at the front end and to display warning dialogs for deletion.
- I have used JavaScript to manage the display of tasks in the table.
- I have used JavaScript to implement event listeners for form submission and button clicks.

2. Backend

3. ASP.NET Core Web API:

- I have used ASP.NET Core Web API to create endpoints for CRUD operations (Create, Read, Update, Delete).
- I have used SQL Database to store created tasks
- I have used ASP.NET Core Web API to define a Task model with the required attributes.

4. SQL Database:

- I have used SQL Database define a schema for the Tasks table.
- I have used SQL Database implement data persistence using SQL scripts.

1.1.1 The .NET Framework (ASP.NET Core MVC)

The Model-View-Controller Architecture

The Task application is developed on the basis of the *Model – View – Controller* architecture. The **MVC** is an architectural pattern that separates an application into three main logical components: the **model**, the **view**, and the **controller**. Each of these components are built to handle specific development aspects of the application making the application scalable and extensible.

1.1.2 Microsoft SQL Server

Microsoft SQL Server is the requisite database environment used by the Task App. SQL Server makes it easier and more cost effective to build high-performance, mission critical applications, enterprise ready Big Data assets, and BI solutions that help employees make better decisions, faster. These solutions have the flexibility of being deployed on premises, in the cloud or in a hybrid environment, and can be managed through a common and familiar tool set.

1.1.3 C# Programming Language

the Task App is developed primarily using the C# programming language. C# (pronounced as see sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines.

1.1.3.1.1 Software Tools and Equipment to run the application in Development Environment

We shall procure the following software tools for the development of the application:

- Visual Studio 2019 Community Edition
- Microsoft SQL Server 2017 or Higher

1.2 Registering the Web Application

For the Task App to be visible as an application, it has to be registered on the Internet Information Services (IIS) to run. To register the Task App with Internet Information Services (IIS), proceed as follows:

- i. Copy the published source code to a folder on the server (or on the network).
- ii. **Open IIS on the server:**
 - Select Windows Start - Control Panel
 - Select System and Security - Administrative Tools

- Select Internet Information Services (IIS)
- Navigate the tree view to Server Name - Web Sites - Default Web Site

iii. **Create a Virtual Site:**

- Right-click default web site and select - New Application. The application creation wizard window will open.
- Enter the Alias – for example www.itc-tasks.un.org and click next to continue.

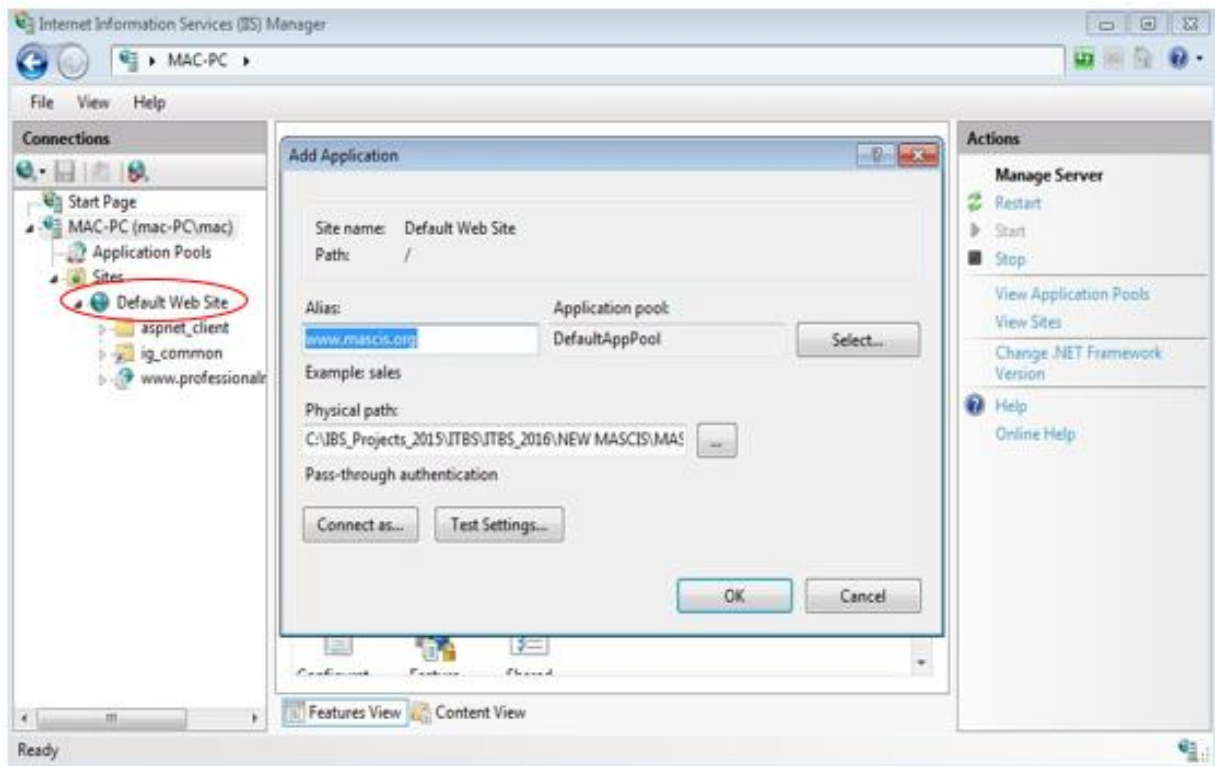
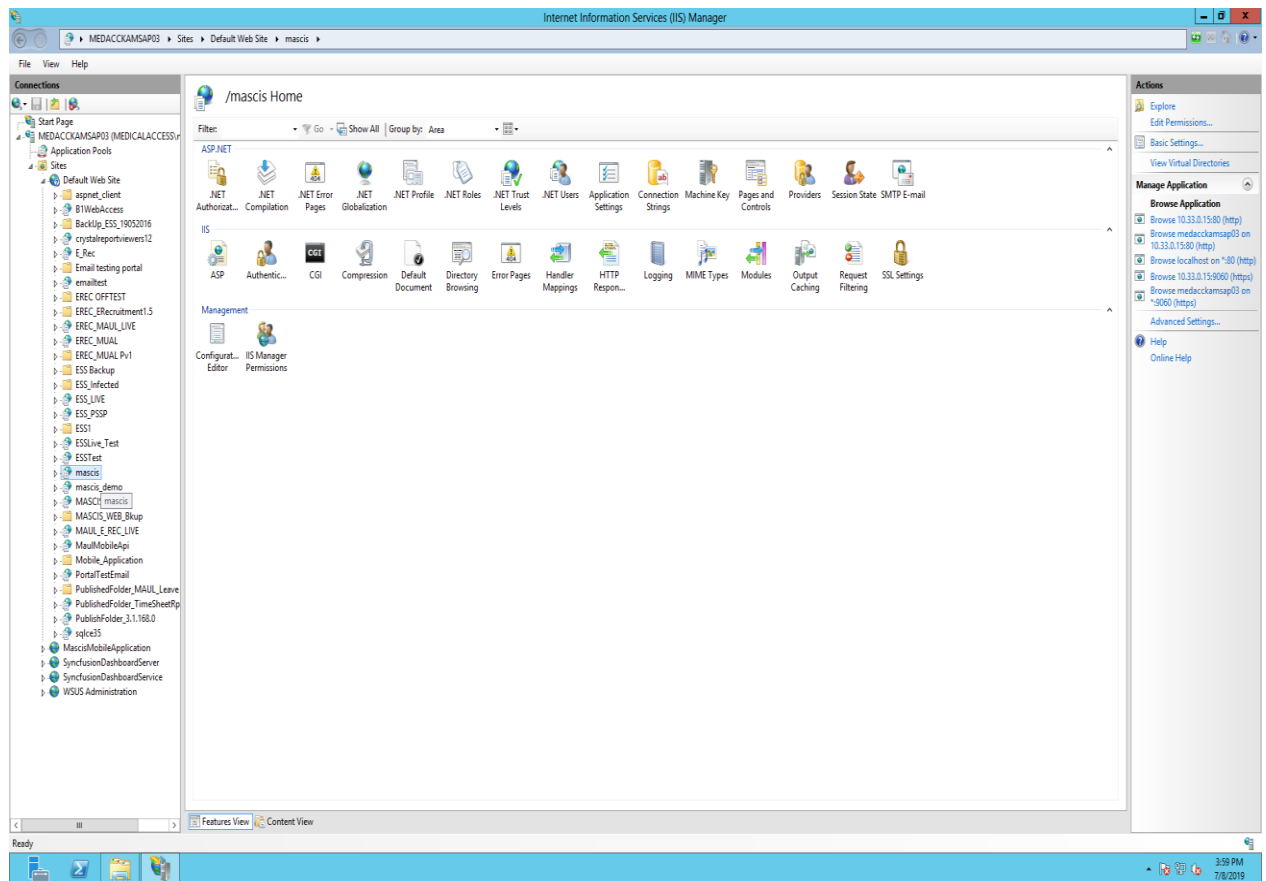


Figure 1: Application creation wizard window

- Enter the Directory – for example C:\Tasks and click next to continue.
- Leave the default security permissions (i.e. Read on and Run Scripts on) and click next to continue.
- Click (**OK**) to complete the registration of the web site.

Figure 2: Registering the Web Application with IIS



2 database Design

2.1 Setting up the Database

To setup the database of the Tasks App, Microsoft SQL Server 20 should have been installed. Running an SQL Script

Script the entire database and run the script from a file as follows:

- i. Establish the script file location
- ii. Open SQL Server Management Studio as an Administrator.
- iii. Navigate the tree view to File - Open - File...
- iv. Click on the Execute button (**Figure 4 below**).

Figure 3: Navigating to the script location

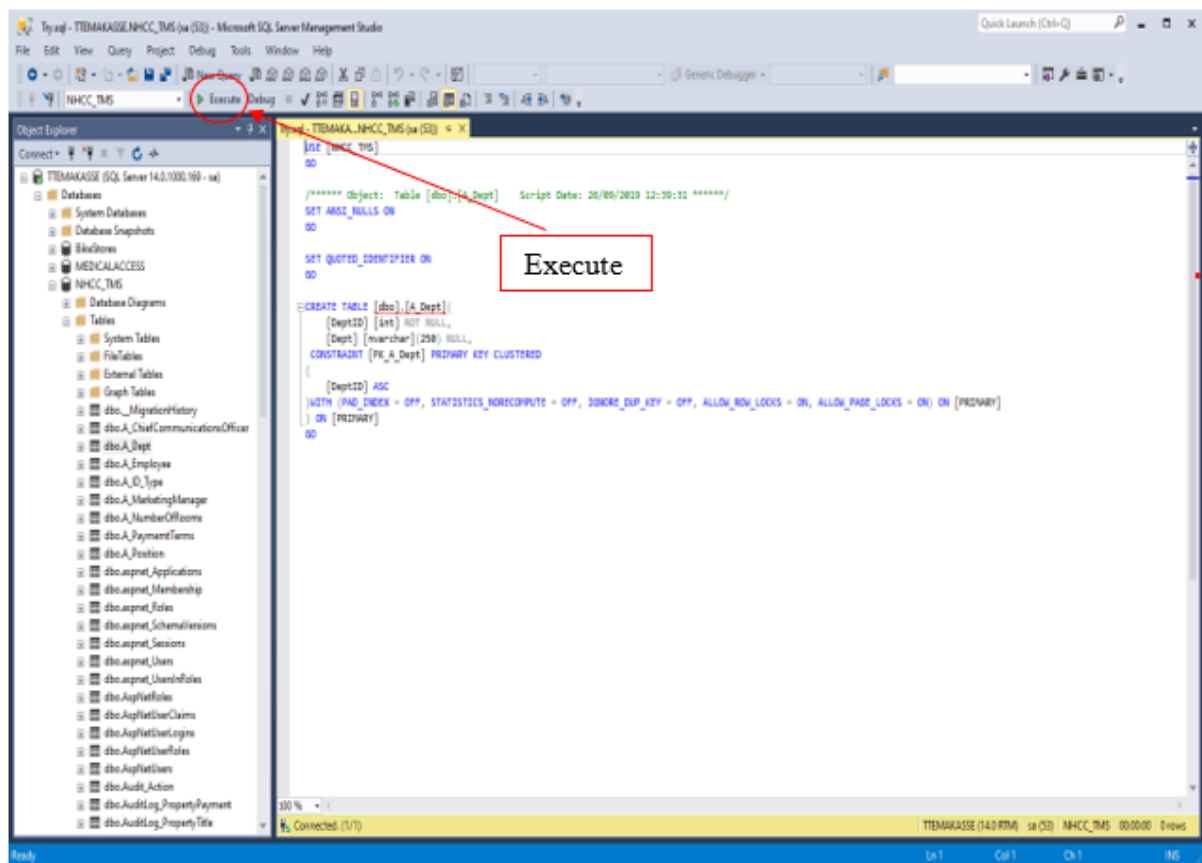
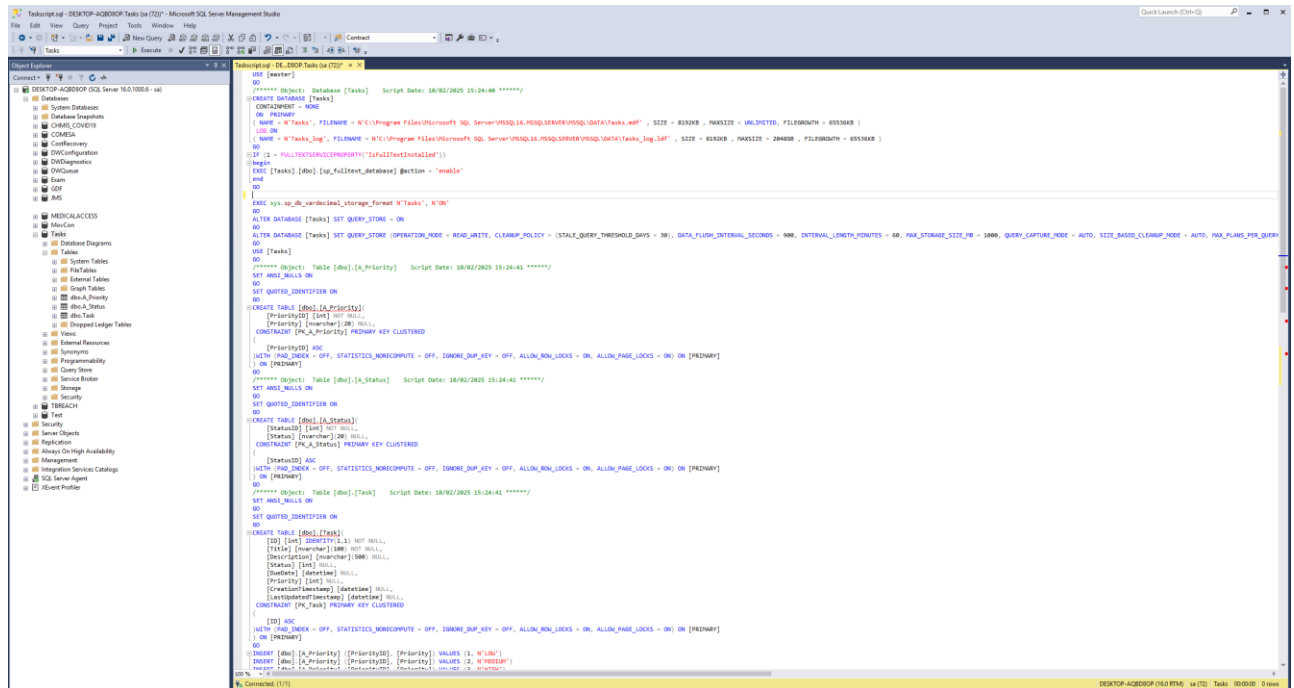


Figure 4: Executing the script in SQL Server Management Studio



2.2 Summary

The Tasks App has database access functions which implement the basic CRUD operations. All these database access functions come with standard error handling. Enums are used to encode the record field mappings – which helps reduce coding errors and helps makes database schema changes easier to manage. The database access layer leverages the data access application block and exception management application block from Microsoft. Using these standard architecture building blocks helps ensure that the application is easy to follow and maintain.