

# **HealthCare Information Management Systems (HIMS)**

**Web Application using MVC Design Pattern, Visual Studio, and SQL Server**

## **Minor Project Report**

**Short-Term Project**

**For practical experiences and understanding of the Software Project Requirement  
and Development**

By:

**Rohan Chauhan: 209202126**

Under the guidance of:  
Dr. Sunil Kumar



**MANIPAL UNIVERSITY  
JAIPUR**

Jan-Feb 2022

---

**Department of Computer and Communication Engineering**

School of Computing and Communication Engineering, Manipal University Jaipur,

Dehmi Kalan, Jaipur, Rajasthan, India- 303007

# STUDENT DECLARATION

I hereby declare that this project **HealthCare Information Management System (HIMS)** is my own work to the best of my knowledge and belief.

This is solely done by me to gain practical experiences and understanding of the Software Project Requirement and Development under the guidance of esteemed professor Dr. Sunil Kumar

Place:

Date:

Rohan Chauhan

(209202126)

B.Tech (CCE) 4<sup>th</sup> Semester

**Department of Computer and Communication Engineering**  
School of Computing and Communication Engineering, Manipal University  
Jaipur, Dehmi Kalan, Jaipur, Rajasthan, India- 303007

Date:

## **CERTIFICATE FROM GUIDE**

This is to certify that the work entitled “**HealthCare Information Management System (HIMS)**” accomplished by Rohan Chauhan (209202126) at Manipal University Jaipur to gain practical experiences and understanding of the Software Project Requirement and Development while studying Bachelor of Technology in Computer and Communication Engineering. This work is carried out by him under my supervision and guidance from Jan 2022 to Feb 2022

Dr. Sunil Kumar

Department of Computer and Communication Engineering  
Manipal University Jaipur

# ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to Dr. Sunil Kumar, my mentor for his invaluable guidance, support, and encouragement throughout my project on **“Healthcare Information Management System”**.

Dr. Kumar’s expertise in the field of Web based Applications, IOT and Insightful inputs have been instrumental in shaping my understanding of developing Web based Application and its intricate workings. His constant motivation and feedback have helped me stay focused and driven towards achieving the project’s objectives.

I would also like to thank the member of our research team for their contributions and cooperation, which have made this project a success. Finally, I would like to acknowledge the support of my family and friends, whose unwavering belief in me has been a source of strength and inspiration.

Once again, Thank you! Dr. Sunil Kumar for your unwavering support, guidance, and mentorship, which have been invaluable to the success of this project.

# Table of Contents

ABSTRACT .....	6
MOTIVATION .....	8
STATEMENT OF PROBLEM.....	9
TECHNOLOGY USED .....	10
DESIGN PATTERN .....	12
DATABASE OBJECT .....	13
<b>TABLE .....</b>	<b>13</b>
[dbo].[Patient].....	13
<b>STORED PROCEDURE.....</b>	<b>14</b>
[dbo].[sproc_GetAllPatient] .....	14
[dbo].[sproc_GetPatientByID].....	15
[dbo].[sproc_InsertPatient] .....	15
[dbo].[sproc_UpdatePatientByID] .....	17
[dbo].[sproc_DeletePatientByID] .....	18
ASP.NET WEB APPLICATION .....	18
<b>Setting up project for ASP.NET Web Application.....</b>	<b>18</b>
<b>Open “Web.config” to add Database Connection Details .....</b>	<b>21</b>
<b>Creation of Model Class for Patient .....</b>	<b>21</b>
<b>Creation of Controller Class for Patient .....</b>	<b>23</b>
<b>Creation of MVC View for Controller Class .....</b>	<b>24</b>
<b>Creation of MVC View for GET Action - Index.....</b>	<b>25</b>
<b>Creation of MVC View for GET Action - Details .....</b>	<b>26</b>
<b>Creation of MVC View for GET Action - Create .....</b>	<b>27</b>
<b>Creation of MVC View for GET Action - Edit .....</b>	<b>27</b>
<b>Creation of MVC View for GET Action - Delete .....</b>	<b>28</b>
<b>Creation Data Access Layer (DAL) Class .....</b>	<b>28</b>
<b>Creation of “PatientController.cs” .....</b>	<b>35</b>
CONCLUSION .....	38
REFERENCES .....	39
APPENDIX .....	39

# **ABSTRACT**

Online Management of HealthCare Information is a promising and efficient approach to manage HealthCare Information that allows HealthCare professional to manage their patient activities, their appointments, diagnosis, prescription, doctor information etc.

By using Web Based and Database Technologies, Online Portal is being developed to record health-related data from patients. HealthCare providers can manage patient health and provide more proactive care. The collected data is then stored securely in the Database. This approach not only enhances the security and privacy of the patient's data but also enables healthcare providers to track and analyze patient health data over time.

In terms of technical aspects, the proposed system would require a robust architecture that can handle large amount of data. This architecture would need to include a data collection user or web interface and data storage module.

Overall, the proposed system has the potential to revolutionize the healthcare industry by providing a secure, efficient, and patient-centric approach to healthcare monitoring.

# INTRODUCTION

The need for proactive and personalized healthcare has been on the rise for the past decade, and healthcare management and monitoring systems have been at the forefront of addressing this need. The healthcare industry has seen a surge in the use of management and monitoring patient health, making it easier for healthcare providers to intervene before critical health issues arise. However, the collection and storage of sensitive patient data has also raised concerns about privacy, security, and the accuracy of the data.

To address these concerns, Web-based and database technology has been proposed as a secure and transparent means of storing and sharing patient data. The proposed system collects data online from Web Application securely stores in Database. This ensures that the data is accessible only to authorized parties and is protected from unauthorized access or tampering. Further, the system has potential to apply machine learning algorithms to analyze the data, providing healthcare insights into patient health. This enables healthcare providers to make informed decisions about patient care, identify potential health risks, and provide personalized treatment options.

The proposed system's technical aspects include an architecture that can handle the large amounts of data collected by the Web Based Application. This architecture comprises a data collection Web Based User Interface module, a Database-based storage module. Additionally, appropriate security measures such as encryption and access control, are employed to protect patient data from unauthorized access and tampering.

Overall, the proposed Healthcare Information Management System has the potential to revolutionize the way healthcare is delivered, improve the quality of care, reduce medical errors, and enhance patient outcomes. By leveraging the benefits of Web and Database Technologies, the system provides a secure, efficient, and patient-centric approach to healthcare management and monitoring.

In conclusion, the integration of Web based Technologies and Database in Healthcare Management and Monitoring systems has the potential to revolutionize the healthcare industry by improving the quality and efficiency of healthcare services.

# MOTIVATION

The motivation behind HealthCare Information Management System (HIMS) using Web Based Technologies and Databased technologies is to address some of the major challenges facing the healthcare industry. These challenges include the need for secure and transparent sharing of patient data, the need for remote patient management and monitoring, and the need for more personalized and efficient healthcare services.

Traditionally, patient data has been siloed within healthcare institutions, making it difficult for healthcare providers to access and share patient data across different healthcare organizations. This can result in duplication of tests, delayed diagnoses, and inefficient healthcare processes.

By using a Web-based Online System, healthcare providers can securely store and share patient data, which can improve the accuracy of diagnoses and treatment plans and reduce costs.

The use of Online HealthCare Information Management Systems enables healthcare providers to collect real-time patient data, which can be used to diagnose and treat patients remotely. This is particularly important for patients with chronic conditions, who require regular monitoring, as it can reduce the need for hospital visits and enable patients to receive more personalized and efficient healthcare services.

Moreover, the use of Online Portal in healthcare can reduce the administrative burden on healthcare providers, enabling them to focus more on patient care. By automating many healthcare processes, such as insurance claims and payments, healthcare providers can reduce costs and improve the efficiency of the healthcare system.

Overall, the motivation behind HealthCare Information Management Systems using Web Based and Database technologies is to improve the quality and efficiency of healthcare services, while addressing some of the major challenges facing the healthcare industry. By leveraging these cutting-edge technologies, healthcare providers can provide more personalized, efficient, and secure healthcare services to patients, thereby improving patient outcomes and the overall healthcare experience.



## **STATEMENT OF PROBLEM**

The current healthcare system is facing several challenges that impact patient outcomes, healthcare costs, and data privacy. These challenges include fragmented healthcare systems, high healthcare costs, limited access to care, and data privacy concerns.

The integration of Web Based Technologies and Database technologies offers a potential solution to these challenges by creating a Smart HealthCare Information Management System (HIMS) that can enhance patient outcomes, reduce healthcare costs, and protect patient data privacy.

# TECHNOLOGY USED

Healthcare Information & Management Systems using Web Application and Database employs technologies to record of patient data, securely and centralized data storage. Some of the key technologies used in these systems include:

1. **Microsoft Visual Studio 2022 Community Edition**: Microsoft Visual Studio is a fully featured, extensible, free IDE (Integrated Development Environment) for creating modern applications for Android, iOS, Windows, as well as web applications and cloud services.

- a. System Requirement for Installing Microsoft Visual Studio

Component	Requirement
Storage	Minimum of 1GB of available hard drive space.
Memory	Minimum: 4 GB
Processor Speed	Minimum: x64 Processor, ARM64

- b. Download Microsoft Visual Studio 2022 Community Edition

<https://learn.microsoft.com/en-us/visualstudio/releases/2022/release-notes>

2. **Microsoft SQL Server 2022**: SQL Server 2022 is ideal for development and production for desktop, web, and small server applications.

- a. System Requirement for Installing Microsoft SQL Server

Component	Requirement
Storage	Minimum of 6 GB of available hard drive space.
Memory	Minimum: Express Editions: 512 MB; All other editions: 1 GB
Processor Speed	Minimum: x64 Processor: 1.4 GHz Recommended: 2.0 GHz or faster
Processor Type	x64 Processor: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support

- b. Download Microsoft SQL Server 2022

<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

3. Operating System

- a. System Requirement

Component	Requirement
Operating system	Windows 10 or greater; Windows Server 2016 or greater.
.NET Framework	Minimum operating systems include minimum .NET framework.

<b>Component</b>	<b>Requirement</b>
Network Software	Supported operating systems for SQL Server have built-in network software. Named and default instances of a stand-alone installation support the following network protocols: Shared memory, Named Pipes, and TCP/IP.

By leveraging these technologies, Online HealthCare Information Management System (HIMS) provide secure and efficient management of patient data, enabling improved healthcare outcomes and reduced healthcare costs.

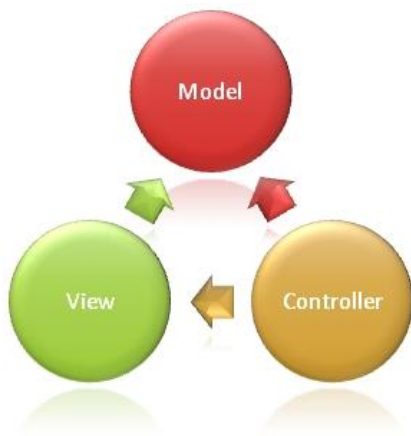
# DESIGN PATTERN

## Model-View-Controller

The Model-View-Controller (MVC) architectural pattern separates an application into three main components: The **Model**, The **View**, and The **Controller**. The ASP.NET MVC framework used for creating MVC-based Web applications.

The MVC framework is defined in the **System.Web.Mvc** namespace and is a fundamental, supported part of the **System.Web** namespace.

The MVC framework includes the following components:



**Models:** Model is used to implement the logic for the Application Data Domain such as properties of database table's columns. Often, model objects retrieve and saves information into the database. For example, a Patient model object retrieve information from a database and saves Patient's updated information back to a Patient table in SQL Server Database.

**Views:** Views is used to display information on Web Or Applications User Interface (UI).

**Controllers:** Controllers is used to handle user requests and based on the request it calls associated model and then it calls associated view to displays the output on UI.

# DATABASE OBJECT

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS).

## TABLE

Tables are database objects that contain all the data in a database. In tables, data is logically organized in a row-and-column. Each row represents a unique record, and each column represents a field in the record.

### [dbo].[Patient]

Name of Database	HIMS
Name of Table	[dbo].[Patient]
Purpose of Table	This table is created to store Patient's information such as their name, DOB and other demographics information.
Definition of Table	<pre>IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Patient]') AND type in (N'U')) DROP TABLE [dbo].[Patient] GO  CREATE TABLE [dbo].[Patient](     [PatientID] [int] IDENTITY(1,1) NOT NULL,     [Firstname] [nvarchar](100) NOT NULL,     [LastName] [nvarchar](100) NOT NULL,     [DOB] [datetime] NOT NULL,     [Address] [nvarchar](500) NULL,     [City] [nvarchar](100) NOT NULL,     [State] [nvarchar](100) NOT NULL,     [Country] [nvarchar](100) NOT NULL,     [Zip] [nvarchar](20) NOT NULL,     [MobileNumber] [varchar](10) NOT NULL,     [IsActive] [bit] NOT NULL,     [DateAdded] [datetime] NOT NULL,     [AddedBy] [nvarchar](100) NOT NULL,     [DateLastModified] [datetime] NOT NULL,     [ModifiedBy] [nvarchar](100) NOT NULL,     PRIMARY KEY CLUSTERED     ([PatientID] ASC )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY] ) ON [PRIMARY] GO  ALTER TABLE [dbo].[Patient] ADD DEFAULT ((1)) FOR [IsActive] GO</pre>

	<pre> ALTER TABLE [dbo].[Patient] ADD DEFAULT (getdate()) FOR [DateAdded] GO ALTER TABLE [dbo].[Patient] ADD DEFAULT ('Admin') FOR [AddedBy] GO ALTER TABLE [dbo].[Patient] ADD DEFAULT (getdate()) FOR [DateLastModified] GO ALTER TABLE [dbo].[Patient] ADD DEFAULT ('Admin') FOR [ModifiedBy] GO </pre>
--	--

## STORED PROCEDURE

[dbo].[sproc\_GetAllPatient]

Name of Database	HIMS
Name of Stored Procedure	[dbo].[sproc_GetAllPatient]
Purpose of Stored Procedure	This procedure is to fetch entire patient's records from database.
Definition of Stored Procedure	<pre> CREATE PROC [dbo].[sproc_GetAllPatient] AS BEGIN  SELECT     [PatientID]     ,[Firstname]     ,[LastName]     ,[DOB]     ,[Address]     ,[City]     ,[State]     ,[Country]     ,[Zip]     ,[MobileNumber]     ,[IsActive]     ,[DateAdded]     ,[AddedBy]     ,[DateLastModified]     ,[ModifiedBy] FROM [dbo].[Patient] WITH (NOLOCK)  END GO </pre>

[dbo].[sproc\_GetPatientByID]

Name of Database	HIMS
Name of Stored Procedure	[dbo].[sproc_GetPatientByID]
Purpose of Stored Procedure	This procedure is to fetch information of a single patient's record from a database.
Definition of Stored Procedure	<pre>CREATE PROC [dbo].[sproc_GetPatientByID] @PatientID INT AS BEGIN  SELECT     [PatientID]     ,[Firstname]     ,[LastName]     ,[DOB]     ,[Address]     ,[City]     ,[State]     ,[Country]     ,[Zip]     ,[MobileNumber]     ,[IsActive]     ,[DateAdded]     ,[AddedBy]     ,[DateLastModified]     ,[ModifiedBy] FROM [dbo].[Patient] WITH (NOLOCK) WHERE [PatientID]=@PatientID  END GO</pre>

[dbo].[sproc\_InsertPatient]

Name of Database	HIMS
Name of Stored Procedure	[dbo].[sproc_InsertPatient]
Purpose of Stored Procedure	This procedure is to insert patient's information into the table.

<p>Definition of Stored Procedure</p>	<pre> CREATE PROC [dbo].[sproc_InsertPatient] @Firstname [nvarchar](100), @LastName [nvarchar](100), @DOB [datetime], @Address [nvarchar](500), @City [nvarchar](100), @State [nvarchar](100), @Country [nvarchar](100), @Zip nvarchar(20), @MobileNumber varchar(10), @IsActive [bit] AS BEGIN  INSERT INTO [dbo].[Patient] (     [Firstname]     ,[LastName]     ,[DOB]     ,[Address]     ,[City]     ,[State]     ,[Country]     ,[Zip]     ,[MobileNumber]     ,[IsActive] ) VALUES (     @Firstname     ,@LastName     ,@DOB     ,@Address     ,@City     ,@State     ,@Country     ,@Zip     ,@MobileNumber     ,@IsActive )  END GO </pre>
---	--



[dbo].[sproc\_UpdatePatientByID]

Name of Database	HIMS
Name of Stored Procedure	[dbo].[sproc_UpdatePatientByID]
Purpose of Stored Procedure	This procedure is to update patient's information into the table.
Definition of Stored Procedure	<pre>CREATE PROC [dbo].[sproc_UpdatePatientByID]     @PatientID int,     @Firstname [nvarchar](100),     @LastName [nvarchar](100),     @DOB [datetime],     @Address [nvarchar](500),     @City [nvarchar](100),     @State [nvarchar](100),     @Country [nvarchar](100),     @Zip nvarchar(20),     @MobileNumber varchar(10),     @IsActive [bit] AS BEGIN  UPDATE [dbo].[Patient] SET     [Firstname]=@Firstname     ,[LastName]=@LastName     ,[Address]=@Address     ,[City]=@City     ,[State]=@State     ,[Country]=@Country     ,[Zip]=@Zip     ,[MobileNumber]=@MobileNumber     ,[IsActive]=@IsActive     ,[DateLastModified]=getdate()     ,[ModifiedBy]='Admin' WHERE PatientID=@PatientID  END GO</pre>

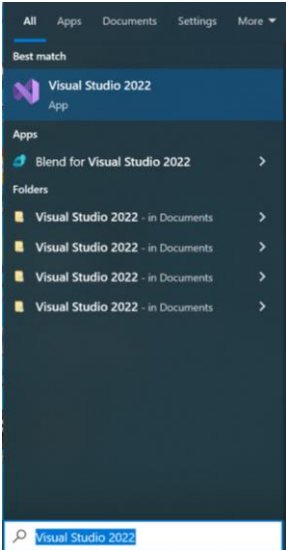
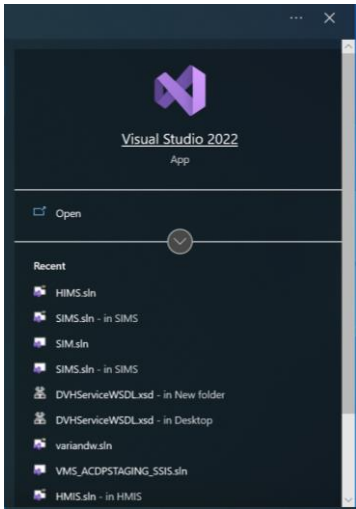

[dbo].[sproc\_DeletePatientByID]

Name of Database	HIMS
Name of Stored Procedure	[dbo].[sproc_DeletePatientByID]
Purpose of Stored Procedure	This procedure is to delete patient's information from the table.
Definition of Stored Procedure	<pre>CREATE PROC [dbo].[sproc_DeletePatientByID] @PatientID int AS BEGIN  DELETE FROM [dbo].[Patient] where PatientID=@PatientID  END GO</pre>

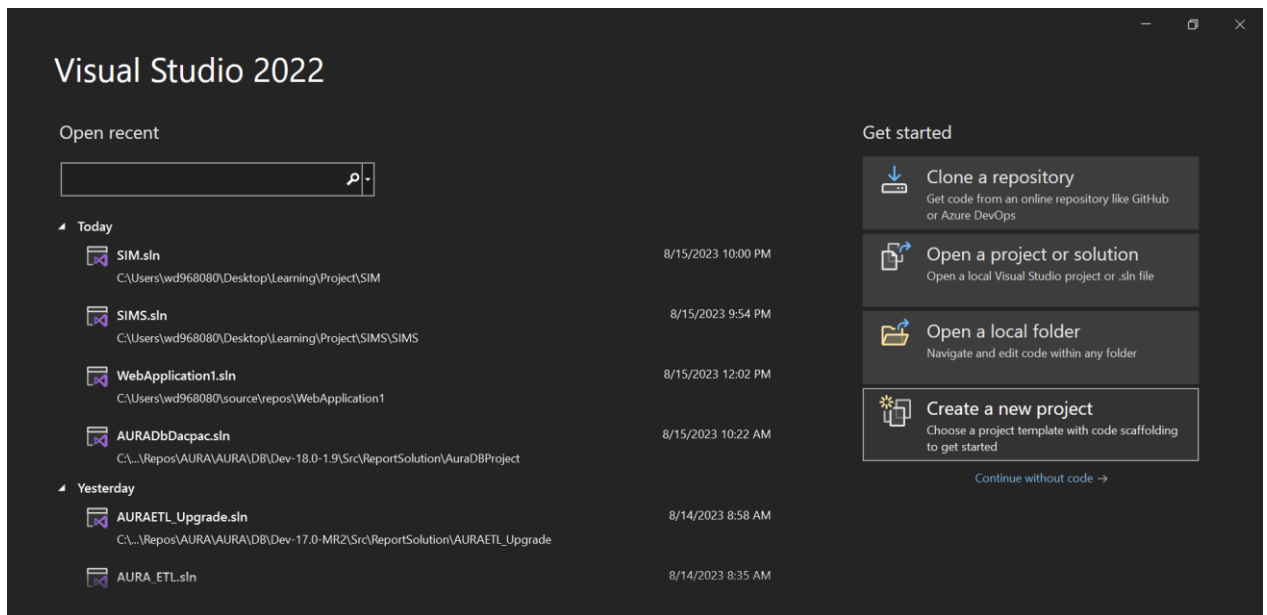
## ASP.NET WEB APPLICATION

Setting up project for ASP.NET Web Application

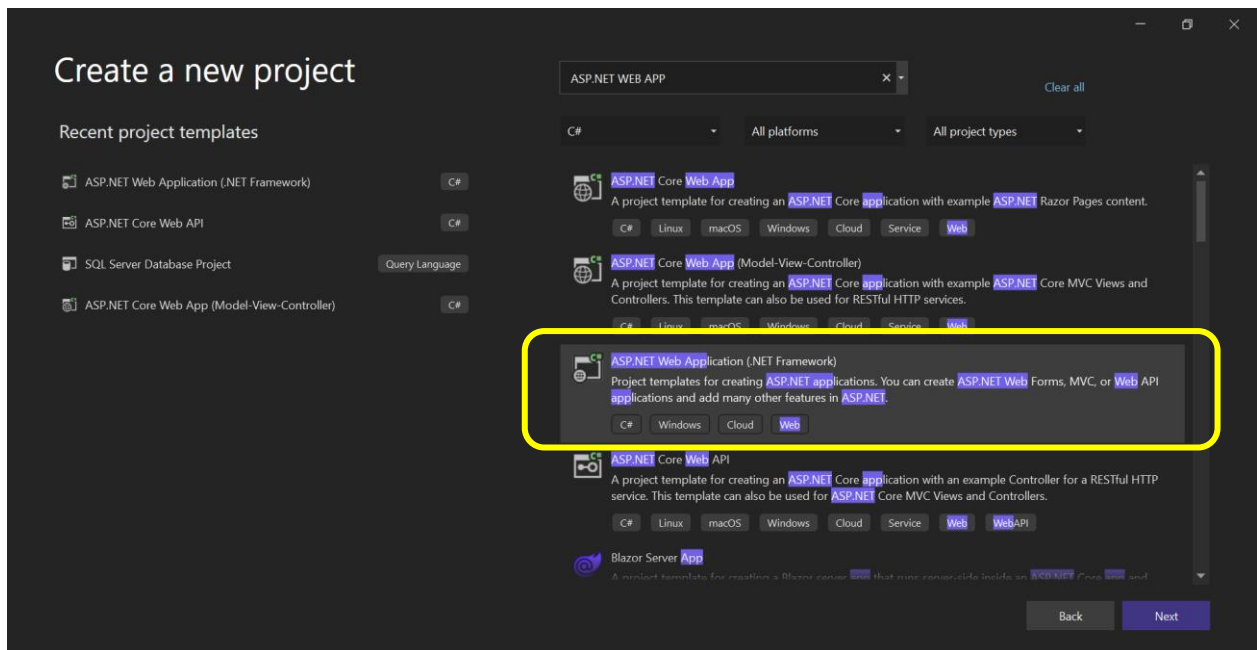
Launch Visual Studio 2022 > Click "Visual Studio 2022" App

<p>Search for Launch Visual Studio 2022</p> 	<p>Click "Visual Studio 2022" App</p> 	<p>Visual Studio 2022 Launches</p> 
---	---	---

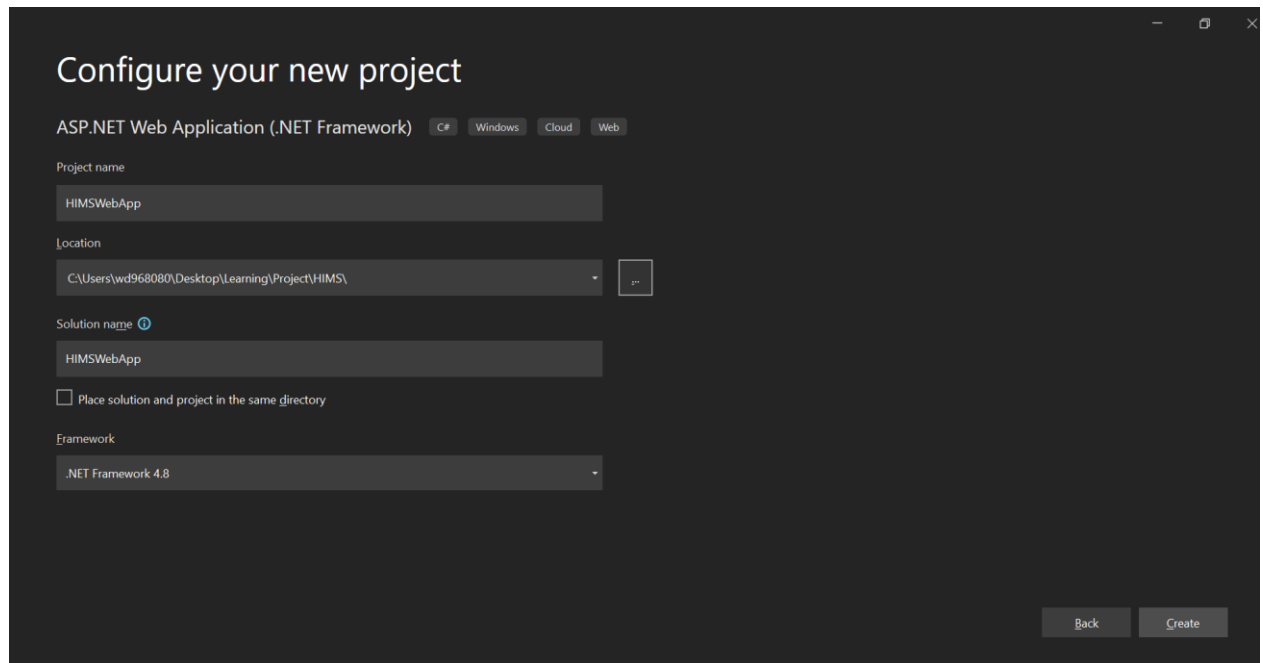
Click on Create New Project



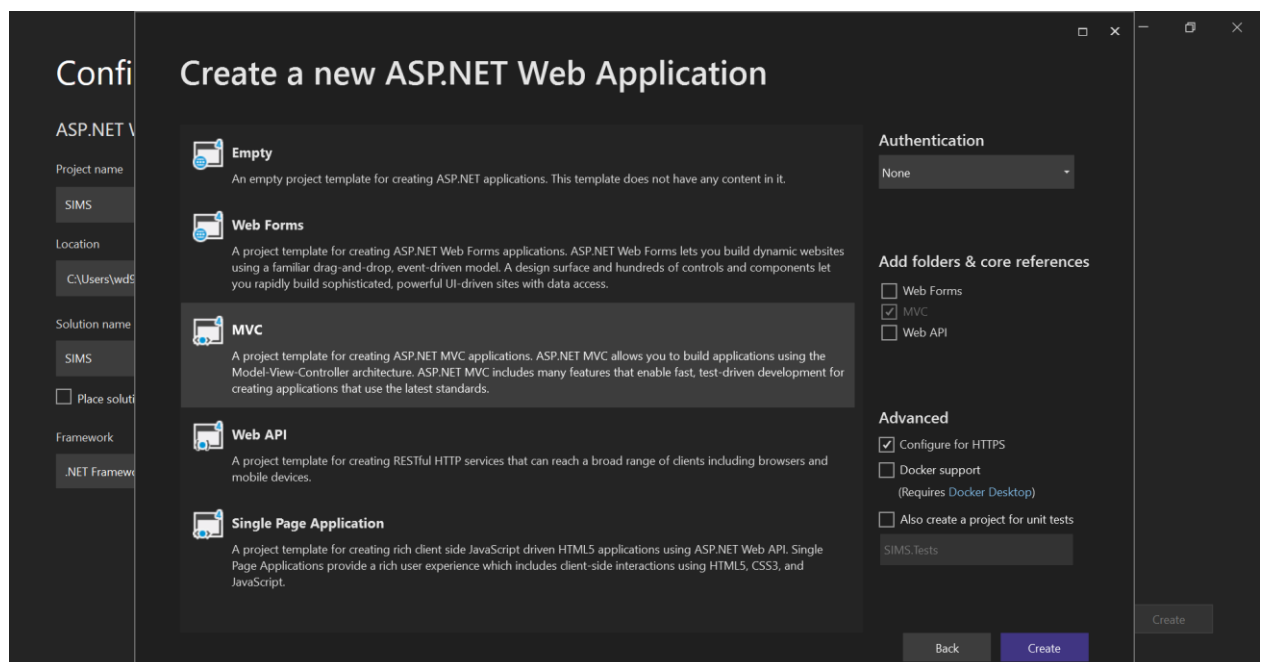
## Select Project Template “ASP.NET Web Application (.Net Framework)”



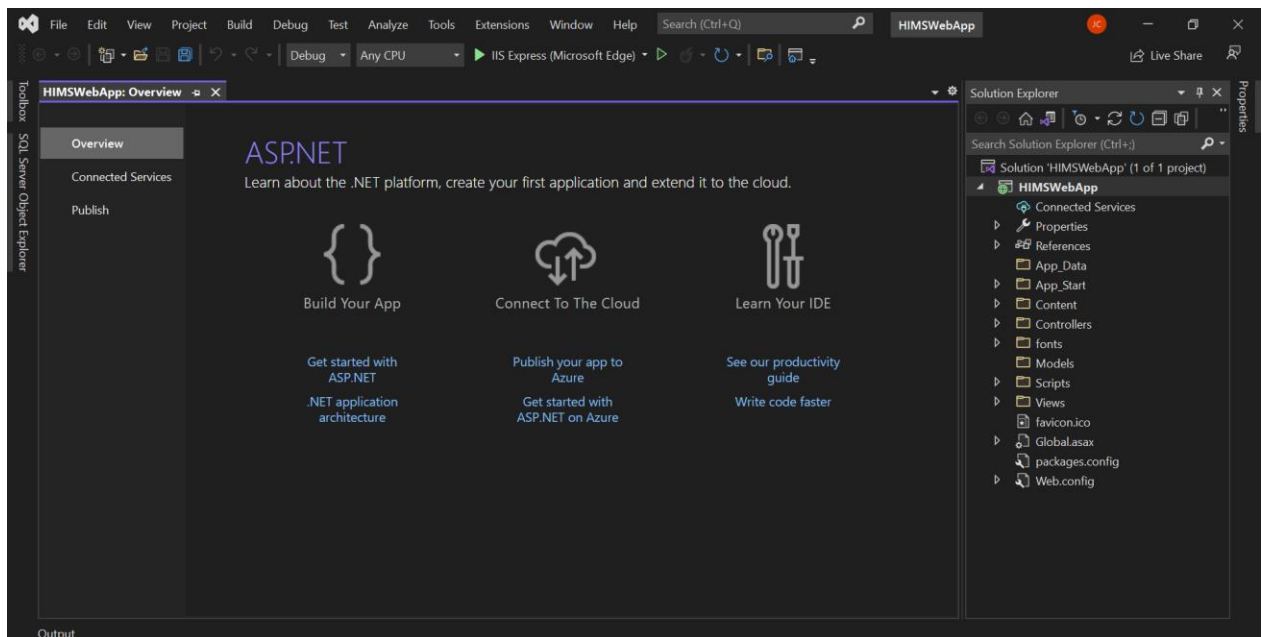
Project Name is given as “HIMSWebApp”



Select MVC

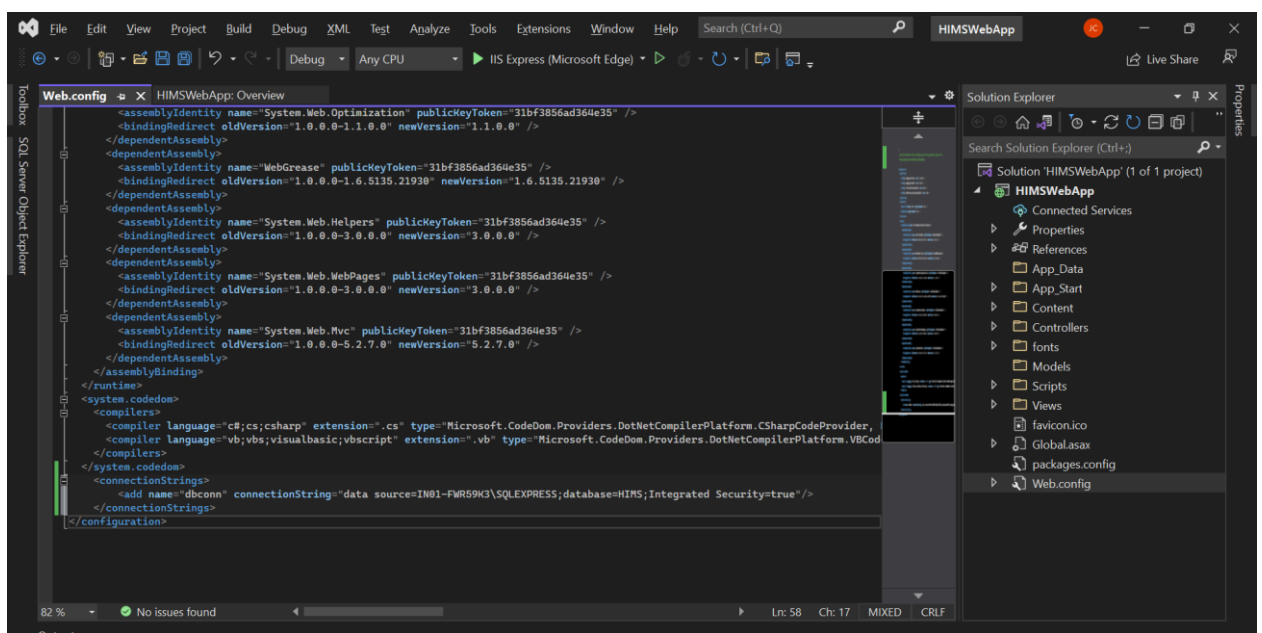


Initial Project Setup looks like as follow



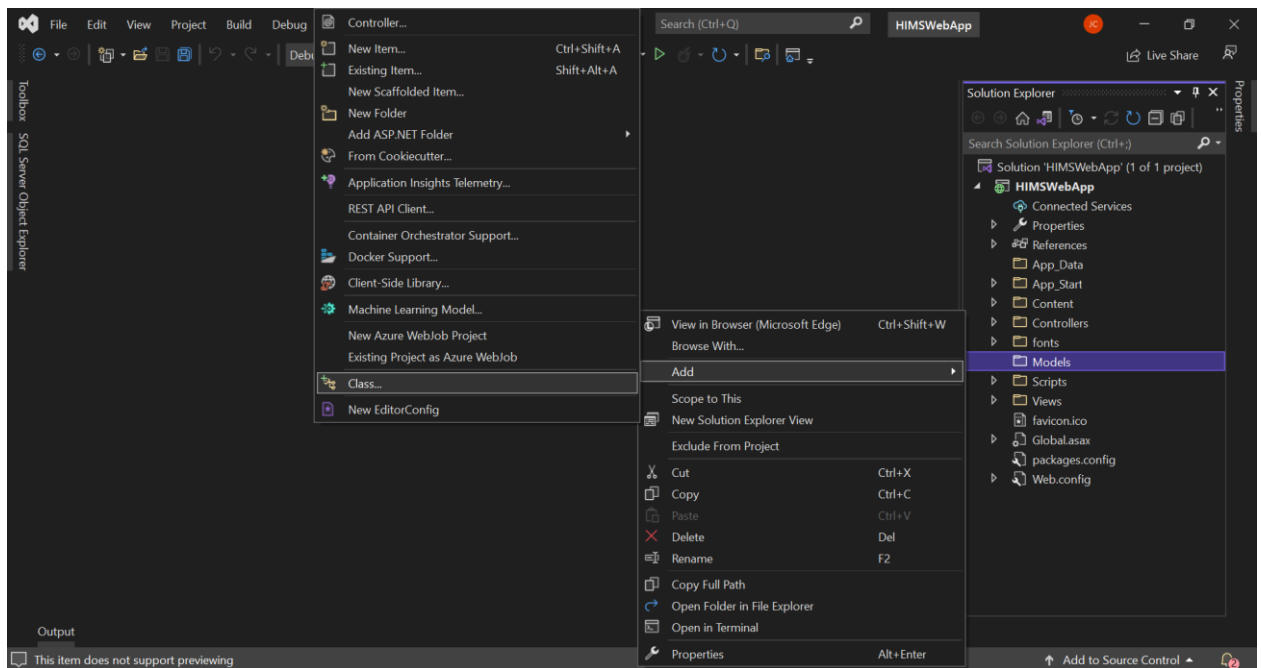
## Open “Web.config” to add Database Connection Details

Open “Web.config” file to provide database connection string such as Name of Database Server, Database Name, Database User etc.

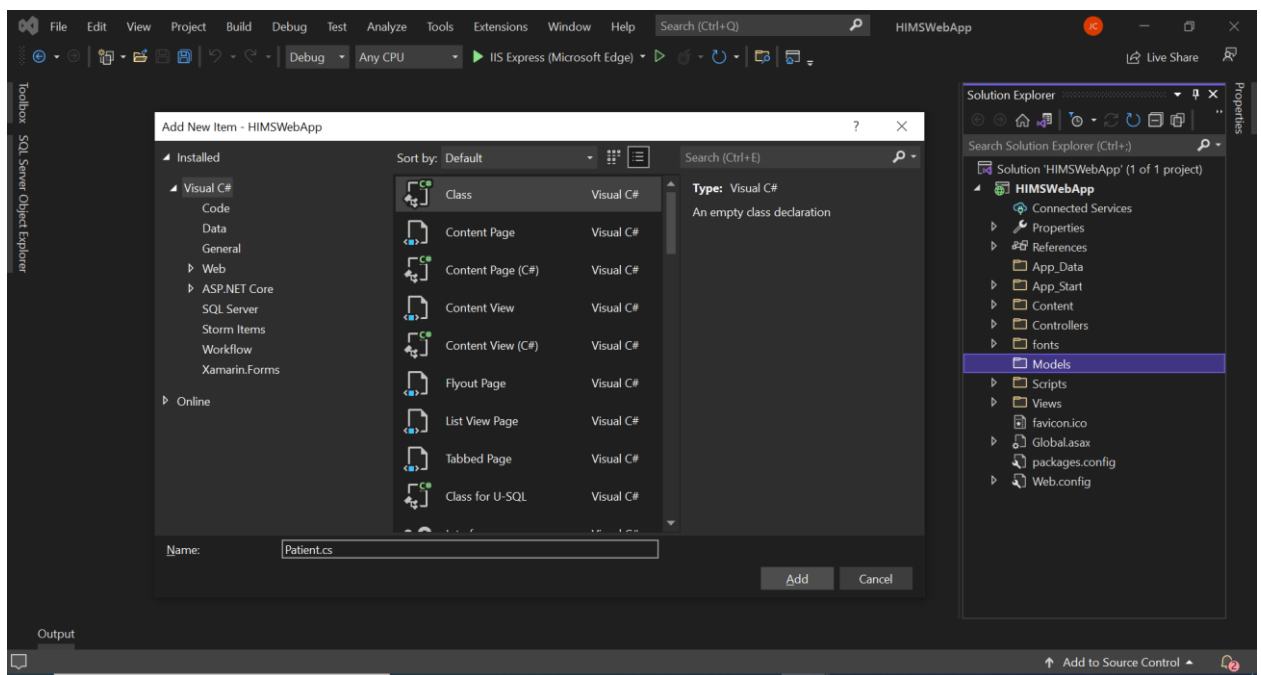


## Creation of Model Class for Patient

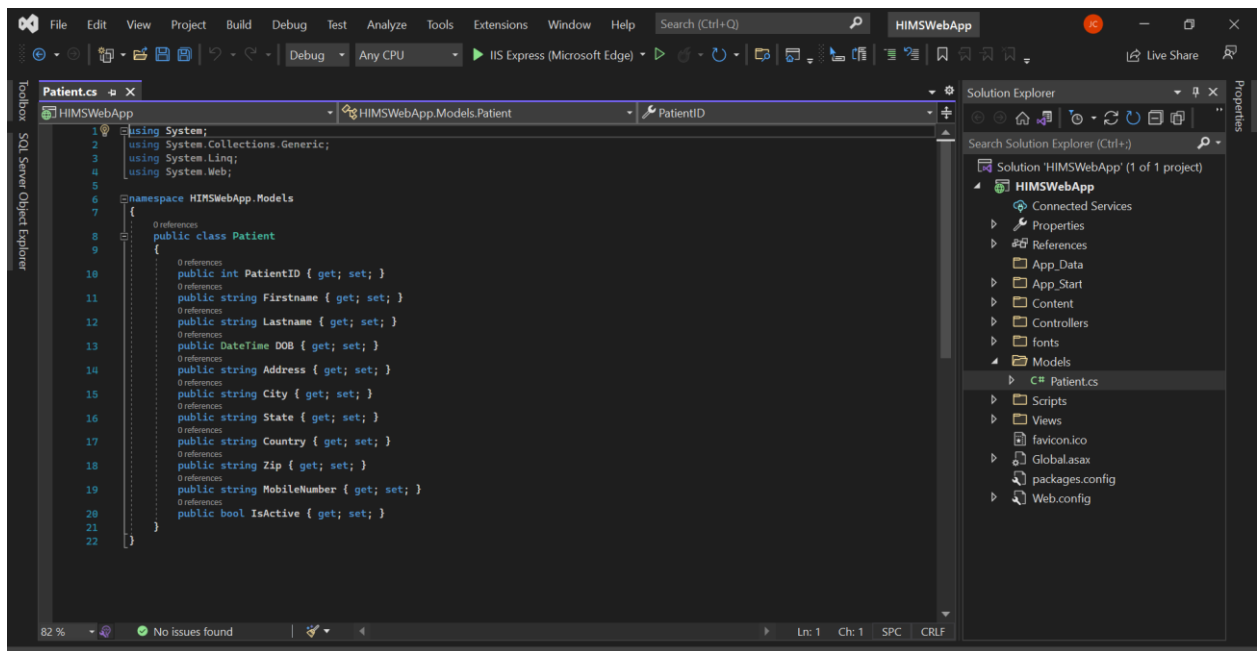
Right Click on “Models” folder > Click “Add” > Click “Class...”



Select “Class” of type Visual C# and then provide name as “Patient.cs” > Click “Add”



Double click to open “Patient.cs” and then declare Properties of Patient like the column names created for “Patient” table in the database.

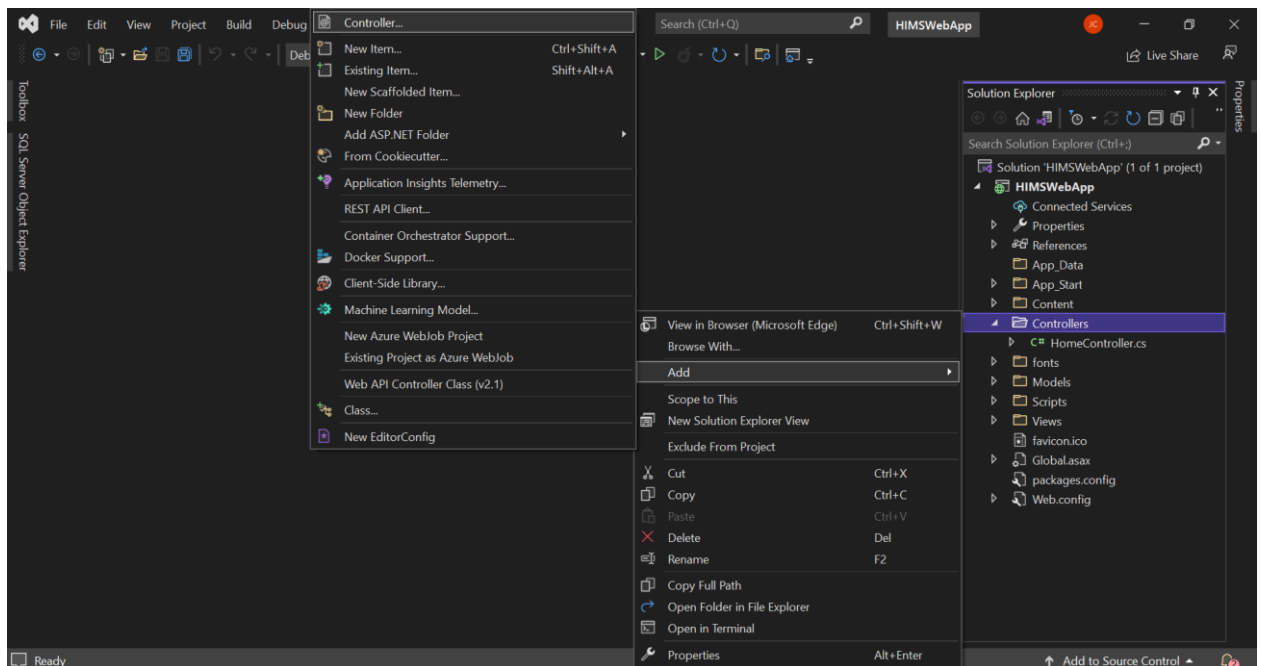


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace HIMSWebApp.Models
{
    public class Patient
    {
        public int PatientID { get; set; }
        public string Firstname { get; set; }
        public string Lastname { get; set; }
        public DateTime DOB { get; set; }
        public string Address { get; set; }
        public string City { get; set; }
        public string State { get; set; }
        public string Country { get; set; }
        public string Zip { get; set; }
        public string MobileNumber { get; set; }
        public bool IsActive { get; set; }
    }
}
```

## Creation of Controller Class for Patient

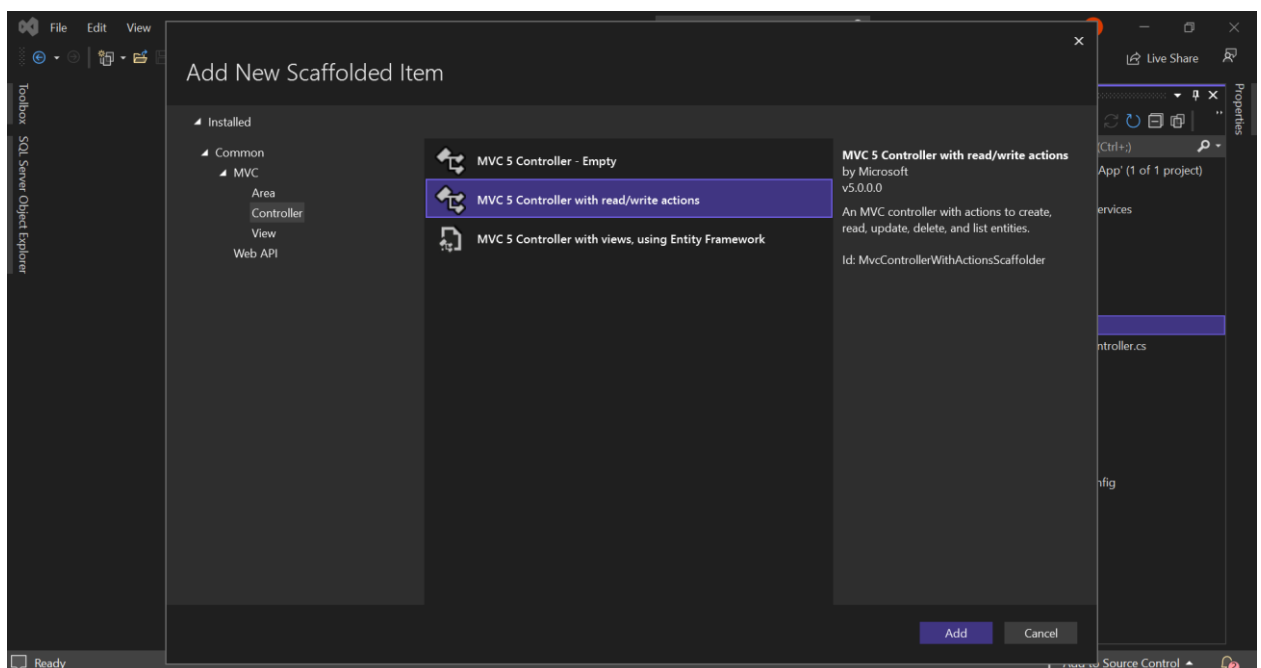
Right click “Controllers” folder > Click “Add” > Click “Controller...”



## Creation of MVC View for Controller Class

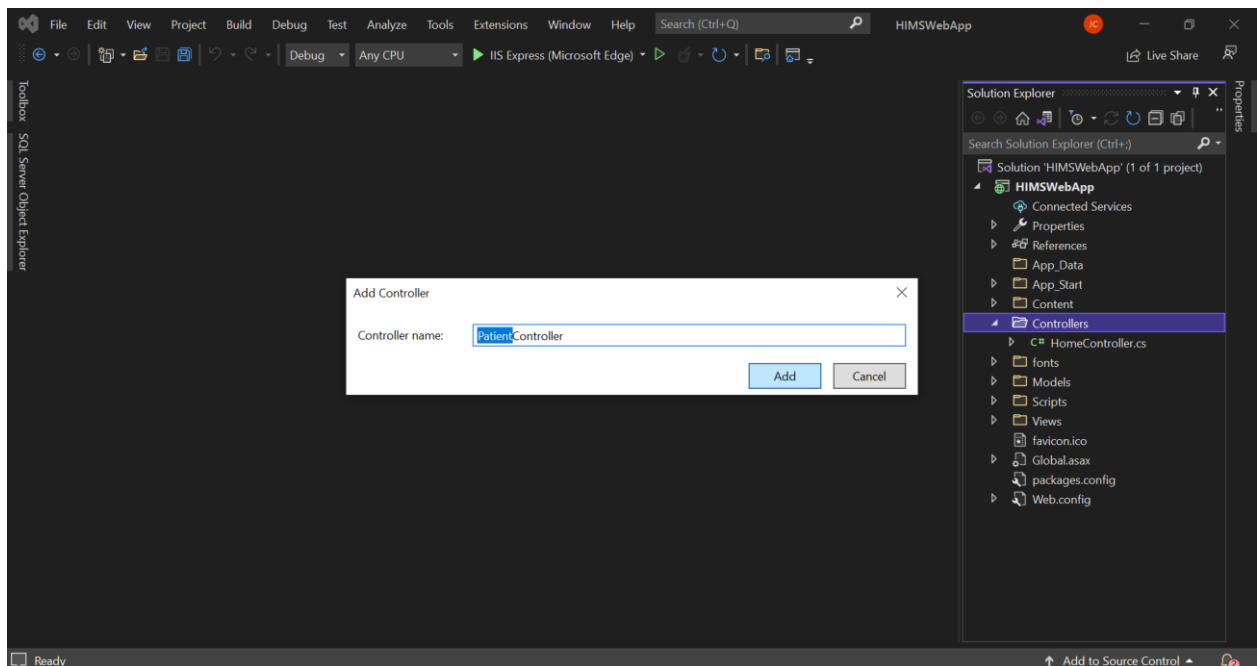
This is used to create MVC View for Controller Class using “MVC 5 Controller with read/write Actions”.

This helps to provides actions to Create, Read, Update, Delete (CRUD) operation.

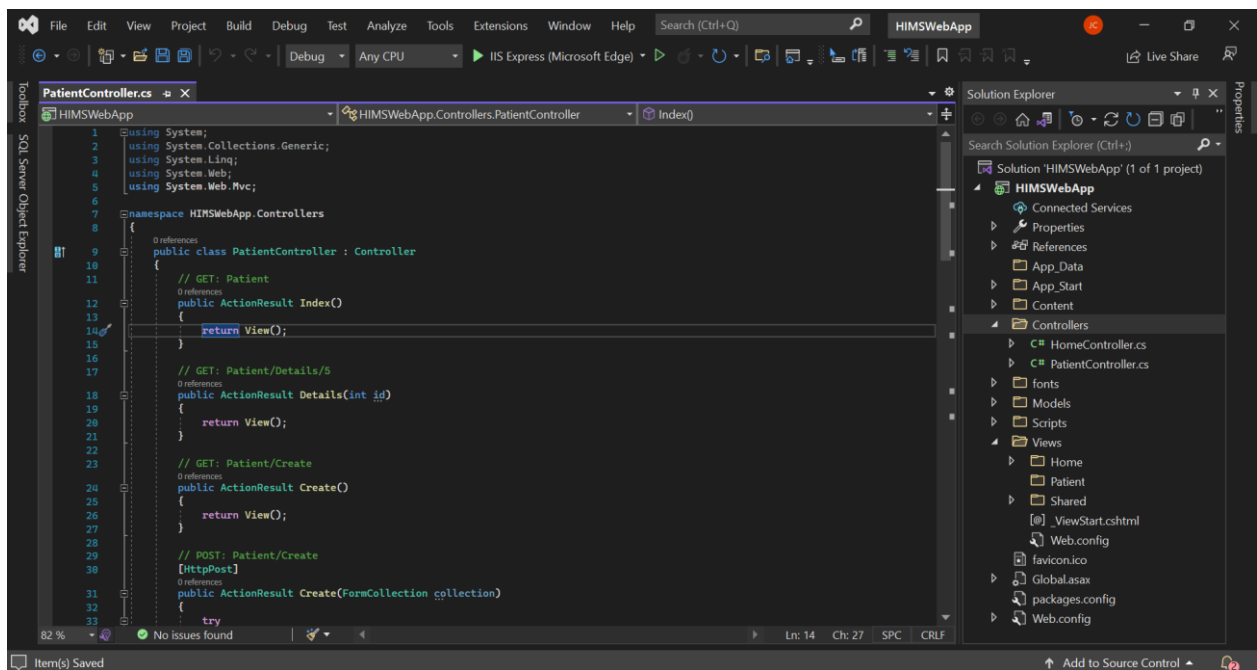


Type name of Controller as “PatientController” > Click “Add”



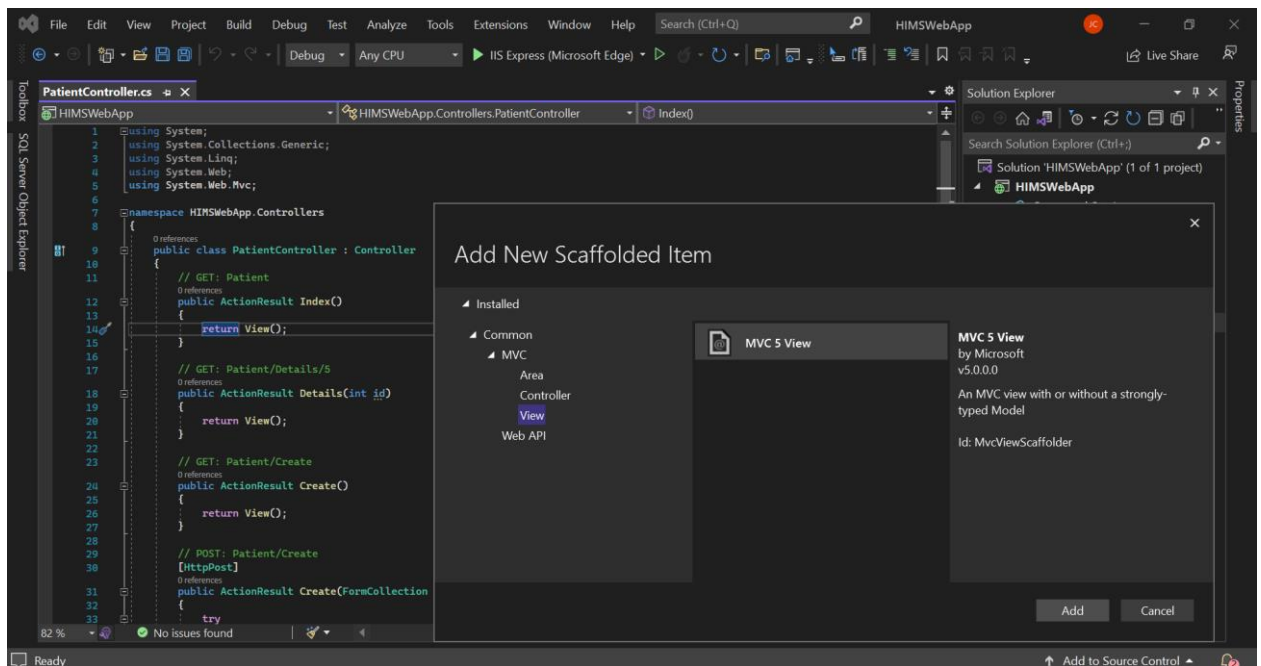


Creation of MVC View for each of the GET Actions - Index, Details, Create, Edit, Delete

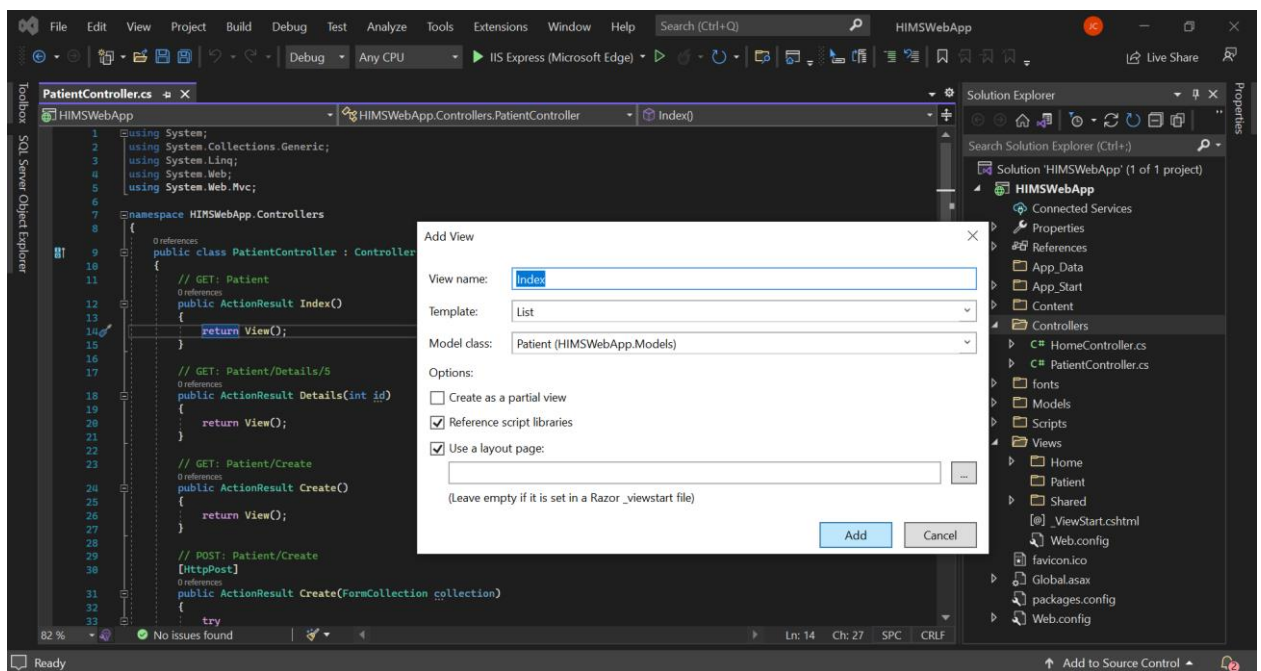


Creation of MVC View for GET Action - Index

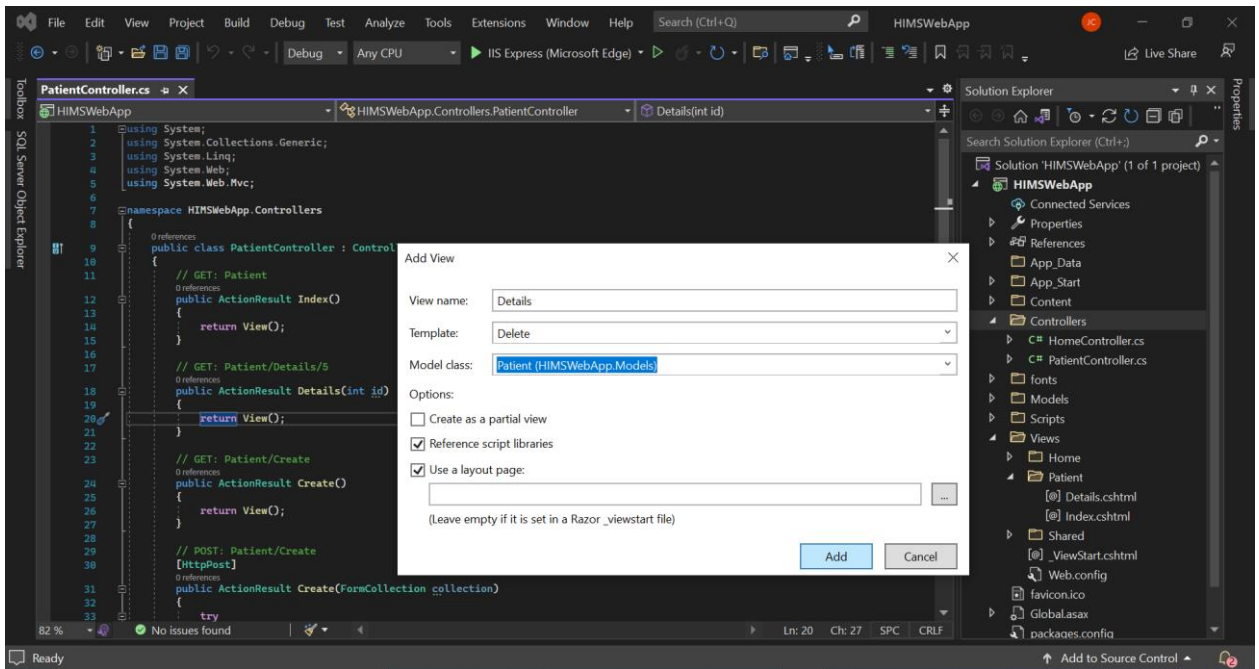
Right click anywhere inside "Index()" code block > Dialog box popup as "Add New Scaffolder Item" > Select MVC 5 View > Click Add



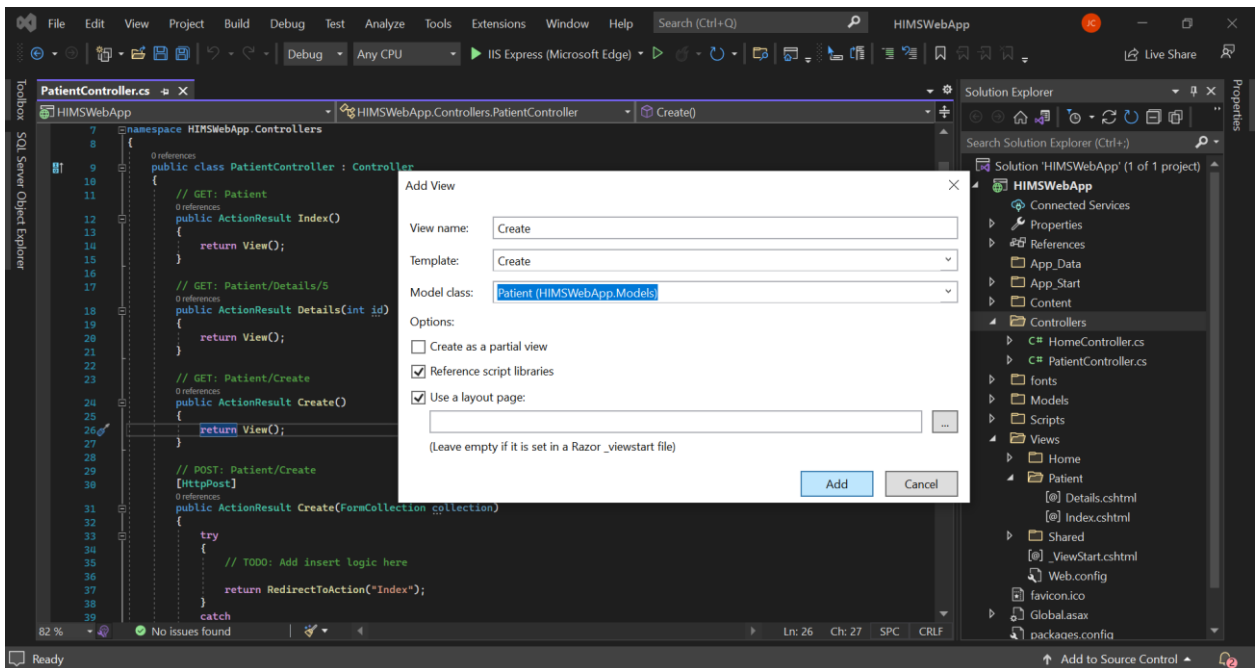
Select View name as "Index" > Select template as "List" > Select model class as "Patient (HIMSWebApp.Models)"



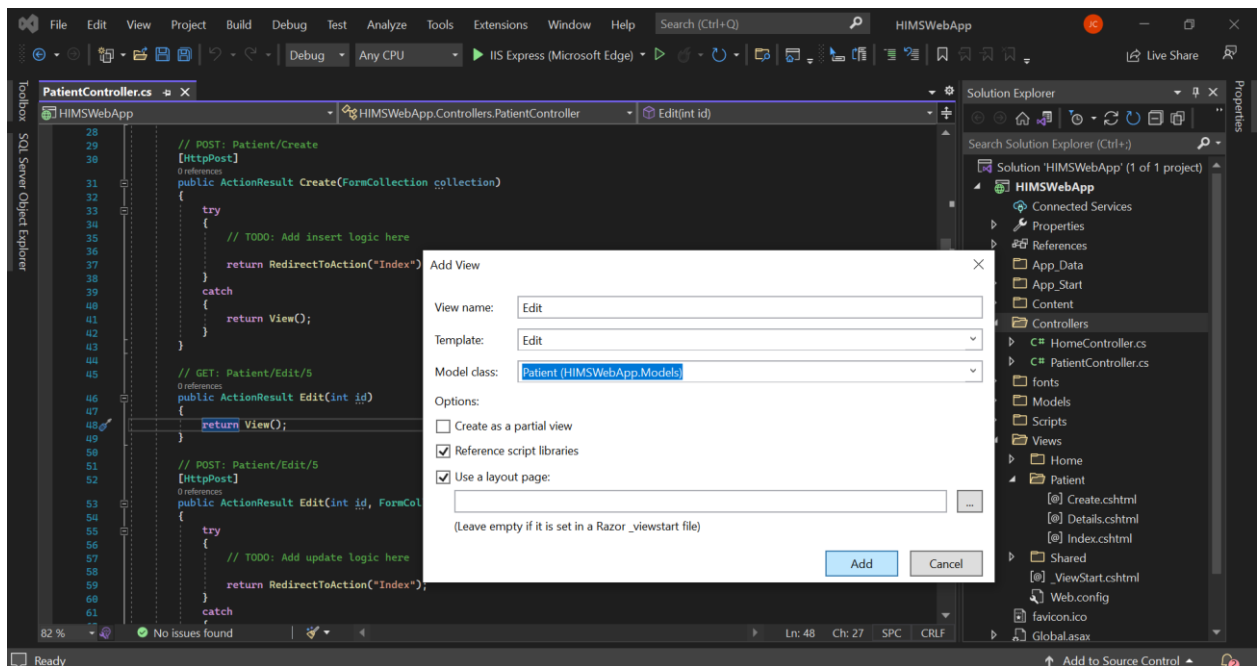
Creation of MVC View for GET Action - Details



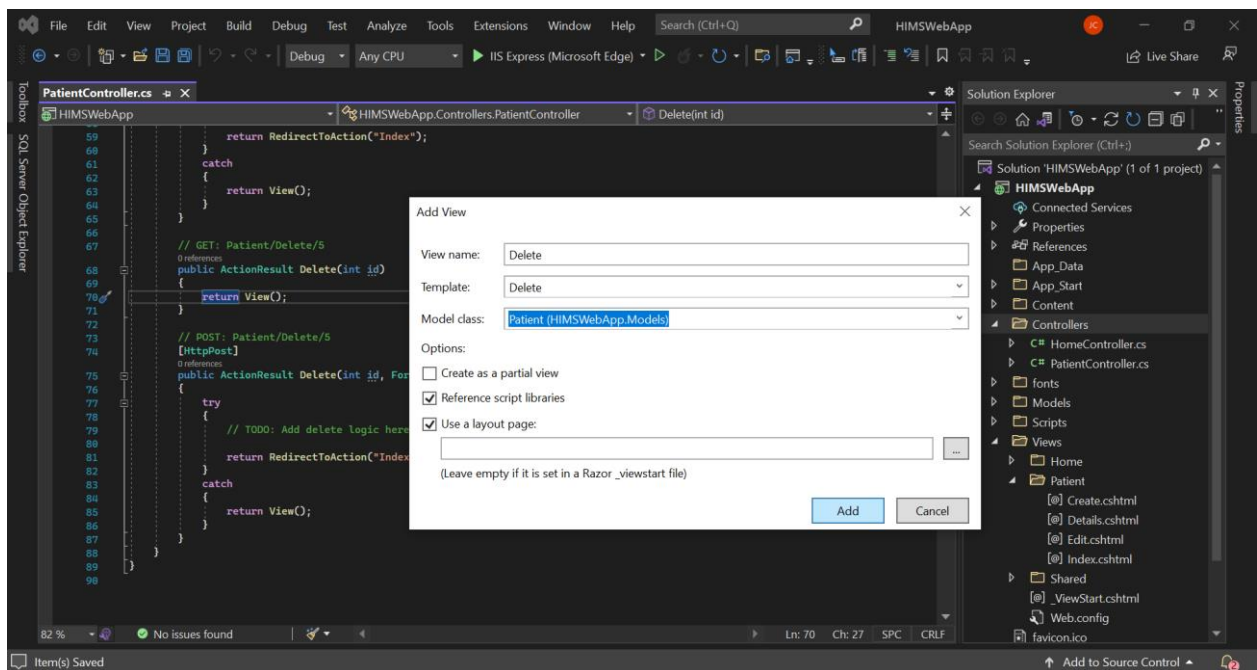
## Creation of MVC View for GET Action - Create



## Creation of MVC View for GET Action - Edit

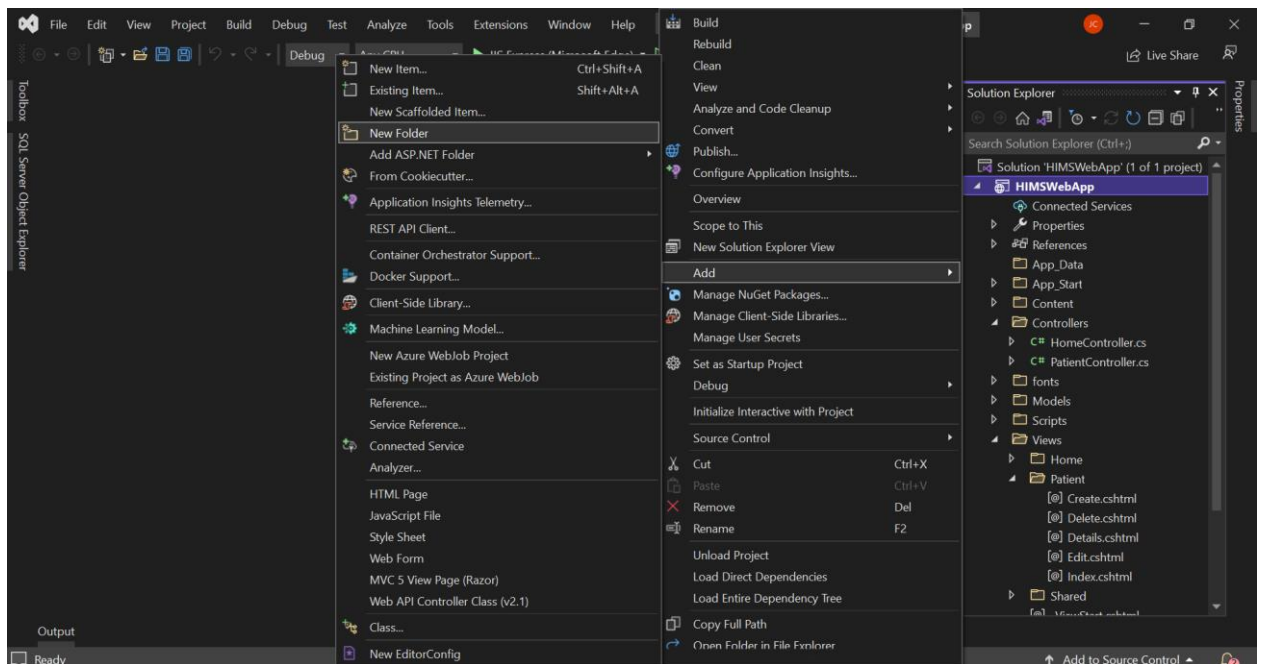


## Creation of MVC View for GET Action - Delete

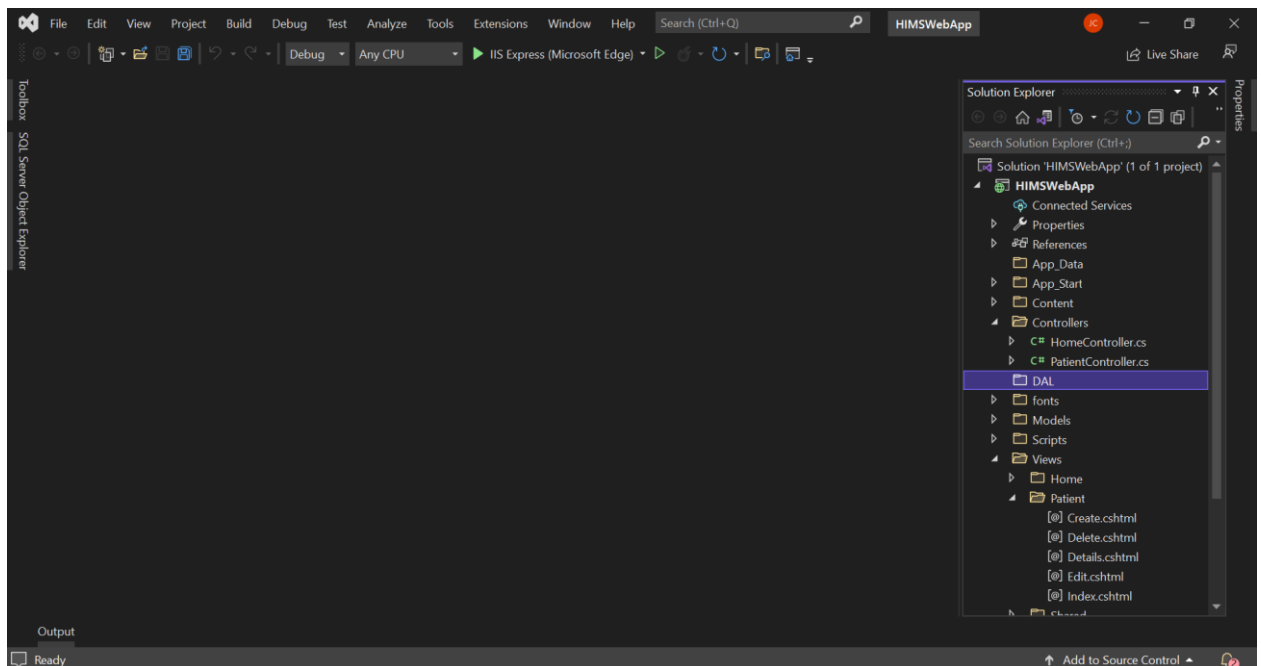


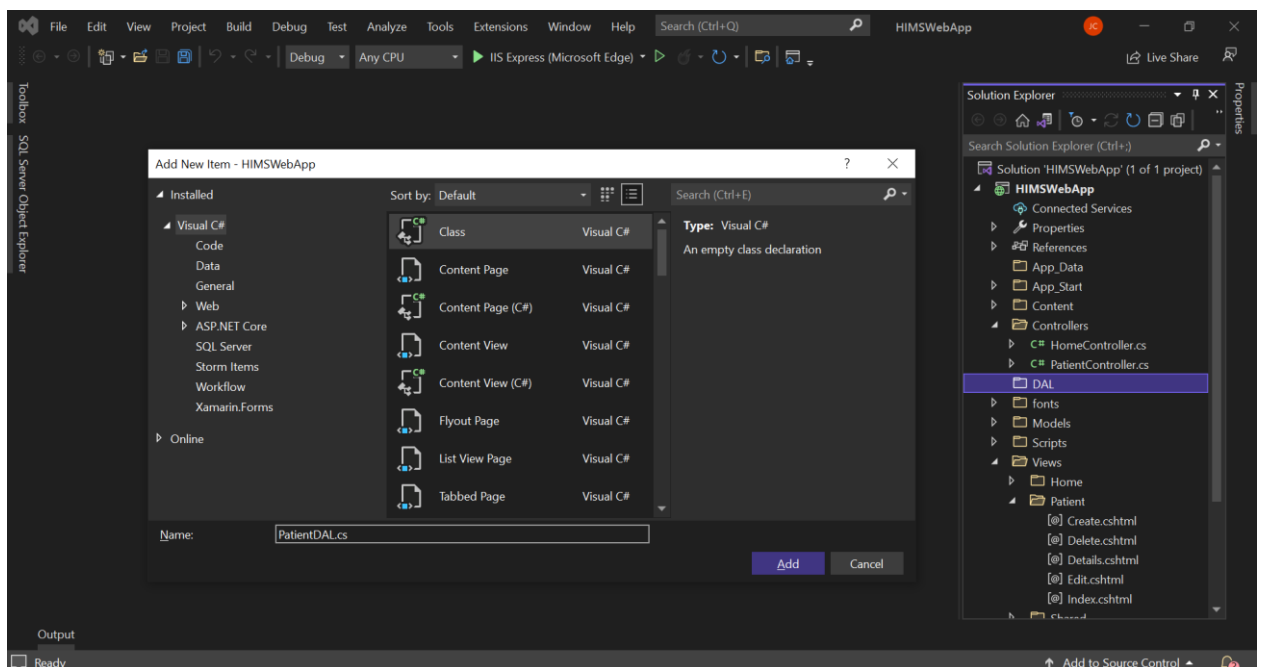
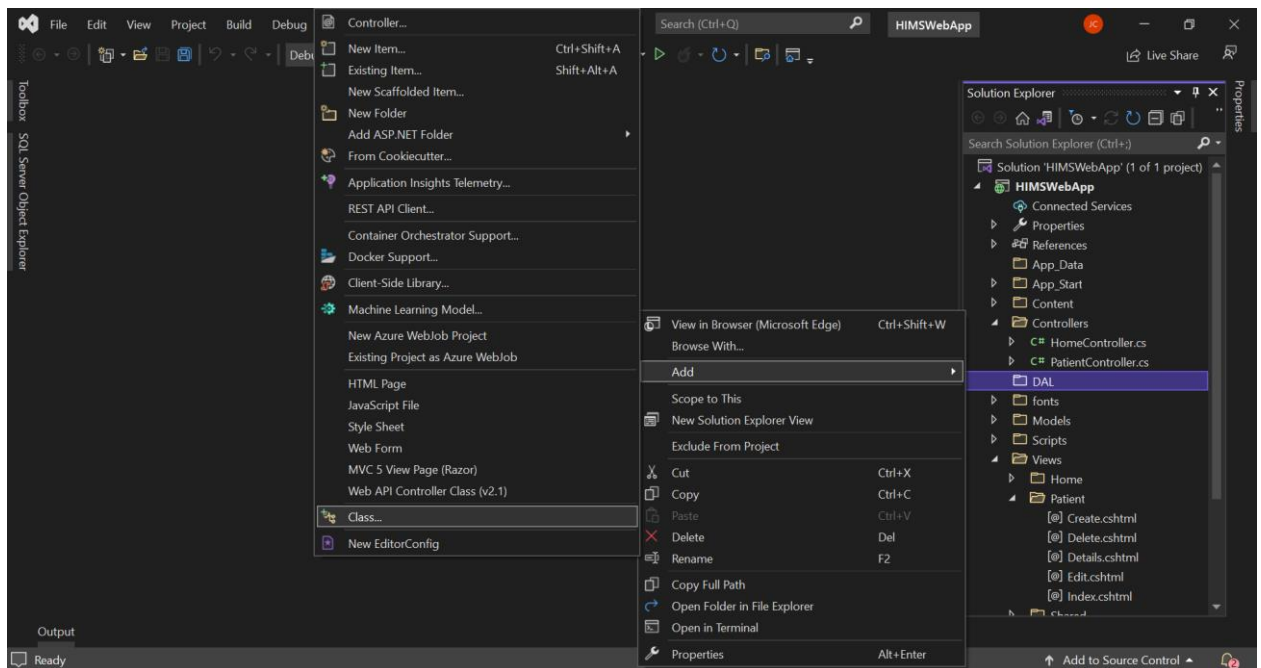
## Creation Data Access Layer (DAL) Class

Right Click Project Name "HIMSWebApp" > Click "Add" > Click "New Folder" > Name the folder as "DAL"



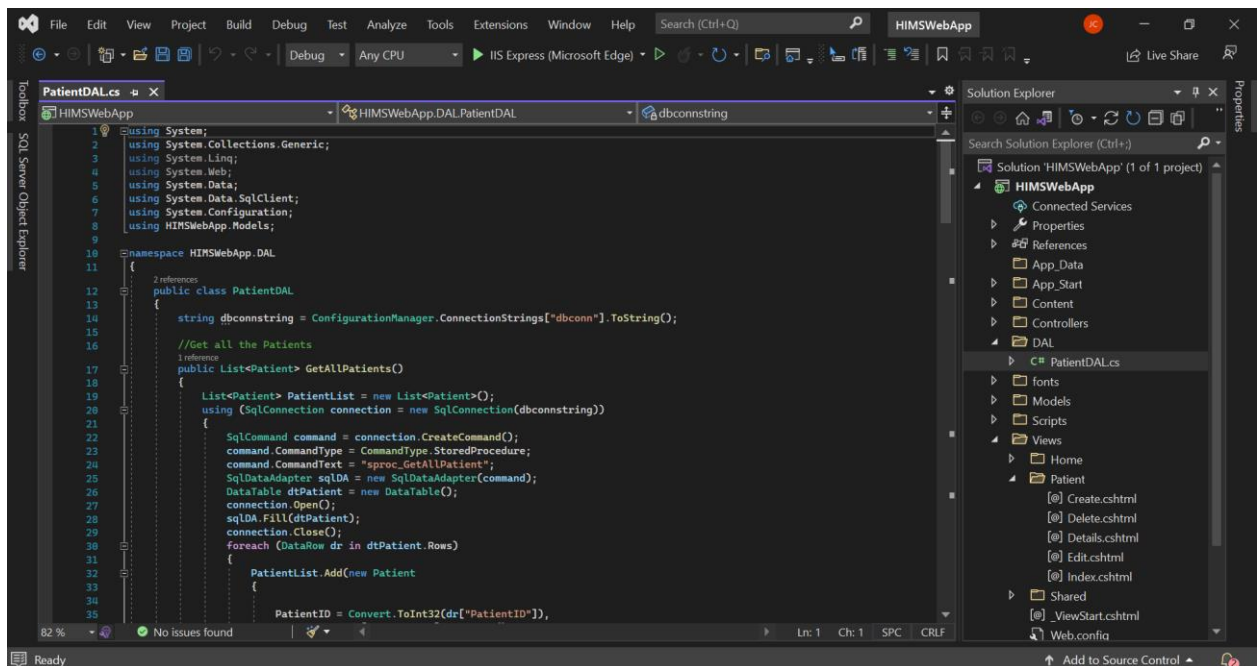
Right Click on Folder Name “DAL” > Click “Add” > Click “Class...” > Name the class as “PatientDAL.cs”





“PatientDAL.cs” to be created as follows





## PatientDAL.cs

Namespace: System.Data.SqlClient

**SqlConnection:** Represents a connection to a SQL Server database.

Property	Purpose
SqlConnection.ConnectionString	Gets or sets the string used to open a SQL Server database.

**SqlCommand:** Represents a Transact-SQL statement or stored procedure to execute against a SQL Server database.

Property	Purpose
SqlCommand.Connection	Gets or sets the SqlConnection used by this instance of the SqlCommand.
SqlCommand.CommandType	Gets or sets a value indicating how the CommandText property is to be interpreted.
SqlCommand.CommandText	Gets or sets the Transact-SQL statement, table name or stored procedure to execute at the data source.
SqlCommand.Parameters	The parameters of the Transact-SQL statement or stored procedure. The default is an empty collection.

**SqlDataReader** Class: Provides a way of reading a forward-only stream of rows from a SQL Server database.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data; //This namespace to be added
using System.Data.SqlClient; //This namespace to be added
```

```

using System.Configuration; //This namespace to be added
using HMSWebApp.Models; //This namespace to be added

namespace HMSWebApp.DAL
{
    public class PatientDAL
    {
        string dbconnstring =
        ConfigurationManager.ConnectionStrings["dbconn"].ToString();

        //Get all the Patients
        public List<Patient> GetAllPatients()
        {
            List<Patient> PatientList = new List<Patient>();
            using (SqlConnection connection = new SqlConnection(dbconnstring))
            {
                SqlCommand command = connection.CreateCommand();
                command.CommandType = CommandType.StoredProcedure;
                command.CommandText = "spc_GetAllPatient";
                SqlDataAdapter sqlDA = new SqlDataAdapter(command);
                DataTable dtPatient = new DataTable();
                connection.Open();
                sqlDA.Fill(dtPatient);
                connection.Close();
                foreach (DataRow dr in dtPatient.Rows)
                {
                    PatientList.Add(new Patient
                    {
                        PatientID = Convert.ToInt32(dr["PatientID"]),
                        Firstname = dr["Firstname"].ToString(),
                        Lastname = dr["LastName"].ToString(),
                        DOB = Convert.ToDateTime(dr["DOB"]),
                        Address = dr["Address"].ToString(),
                        City = dr["City"].ToString(),
                        State = dr["State"].ToString(),
                        Country = dr["Country"].ToString(),
                        Zip = dr["Zip"].ToString(),
                        MobileNumber = dr["MobileNumber"].ToString(),
                        IsActive = Convert.ToBoolean(dr["IsActive"])
                    });
                }
            }

            return PatientList;
        }

        //Insert Patients
        public bool InsertPatients(Patient Patient)
        {

```



```

using (SqlConnection connection = new SqlConnection(dbconnstring))
{
    SqlCommand command = new SqlCommand("spcproc_InsertPatient", connection);
    command.CommandType = CommandType.StoredProcedure;
    command.Parameters.AddWithValue("@Firstname", Patient.Firstname);
    command.Parameters.AddWithValue("@LastName", Patient.Lastname);
    command.Parameters.AddWithValue("@DOB", Patient.DOB);
    command.Parameters.AddWithValue("@Address", Patient.Address);
    command.Parameters.AddWithValue("@City", Patient.City);
    command.Parameters.AddWithValue("@State", Patient.State);
    command.Parameters.AddWithValue("@Country", Patient.Country);
    command.Parameters.AddWithValue("@Zip", Patient.Zip);
    command.Parameters.AddWithValue("@MobileNumber",
Patient.MobileNumber);
    command.Parameters.AddWithValue("@IsActive", Patient.IsActive);

    connection.Open();
    command.ExecuteNonQuery();
    connection.Close();
}
return true;
}

//Get Patient By ID
public List<Patient> GetPatientByID(int PatientID)
{
    List<Patient> PatientList = new List<Patient>();
    using (SqlConnection connection = new SqlConnection(dbconnstring))
    {
        SqlCommand command = connection.CreateCommand();
        command.CommandType = CommandType.StoredProcedure;
        command.CommandText = "spcproc_GetPatientByID";
        command.Parameters.AddWithValue("@PatientID", PatientID);
        SqlDataAdapter sqlDA = new SqlDataAdapter(command);
        DataTable dtPatient = new DataTable();
        connection.Open();
        sqlDA.Fill(dtPatient);
        connection.Close();
        foreach (DataRow dr in dtPatient.Rows)
        {
            PatientList.Add(new Patient
            {
                PatientID = Convert.ToInt32(dr["PatientID"]),
                Firstname = dr["Firstname"].ToString(),
                Lastname = dr["LastName"].ToString(),
                DOB = Convert.ToDateTime(dr["DOB"]),
                Address = dr["Address"].ToString(),
                City = dr["City"].ToString(),
                State = dr["State"].ToString(),
            });
        }
    }
}

```

```

        Country = dr["Country"].ToString(),
        Zip = dr["Zip"].ToString(),
        MobileNumber = dr["MobileNumber"].ToString(),
        IsActive = Convert.ToBoolean(dr["IsActive"])
    });
}

}

return PatientList;
}

//Update Patient By ID
public bool UpdatePatients(Patient Patient)
{
    using (SqlConnection connection = new SqlConnection(dbconnstring))
    {
        SqlCommand command = new SqlCommand("sproc_UpdatePatientByID",
connection);
        command.CommandType = CommandType.StoredProcedure;
        command.Parameters.AddWithValue("@PatientID", Patient.PatientID);
        command.Parameters.AddWithValue("@Firstname", Patient.Firstname);
        command.Parameters.AddWithValue("@LastName", Patient.Lastname);
        command.Parameters.AddWithValue("@DOB", Patient.DOB);
        command.Parameters.AddWithValue("@Address", Patient.Address);
        command.Parameters.AddWithValue("@City", Patient.City);
        command.Parameters.AddWithValue("@State", Patient.State);
        command.Parameters.AddWithValue("@Country", Patient.Country);
        command.Parameters.AddWithValue("@Zip", Patient.Zip);
        command.Parameters.AddWithValue("@MobileNumber",
Patient.MobileNumber);
        command.Parameters.AddWithValue("@IsActive", Patient.IsActive);
        connection.Open();
        command.ExecuteNonQuery();
        connection.Close();
    }
    return true;
}

//Delete Patient By ID
public bool DeletePatients(int StuentID)
{
    using (SqlConnection connection = new SqlConnection(dbconnstring))
    {
        SqlCommand command = new SqlCommand("sproc_DeletePatientByID",
connection);
        command.CommandType = CommandType.StoredProcedure;
        command.Parameters.AddWithValue("@PatientID", StuentID);
        connection.Open();

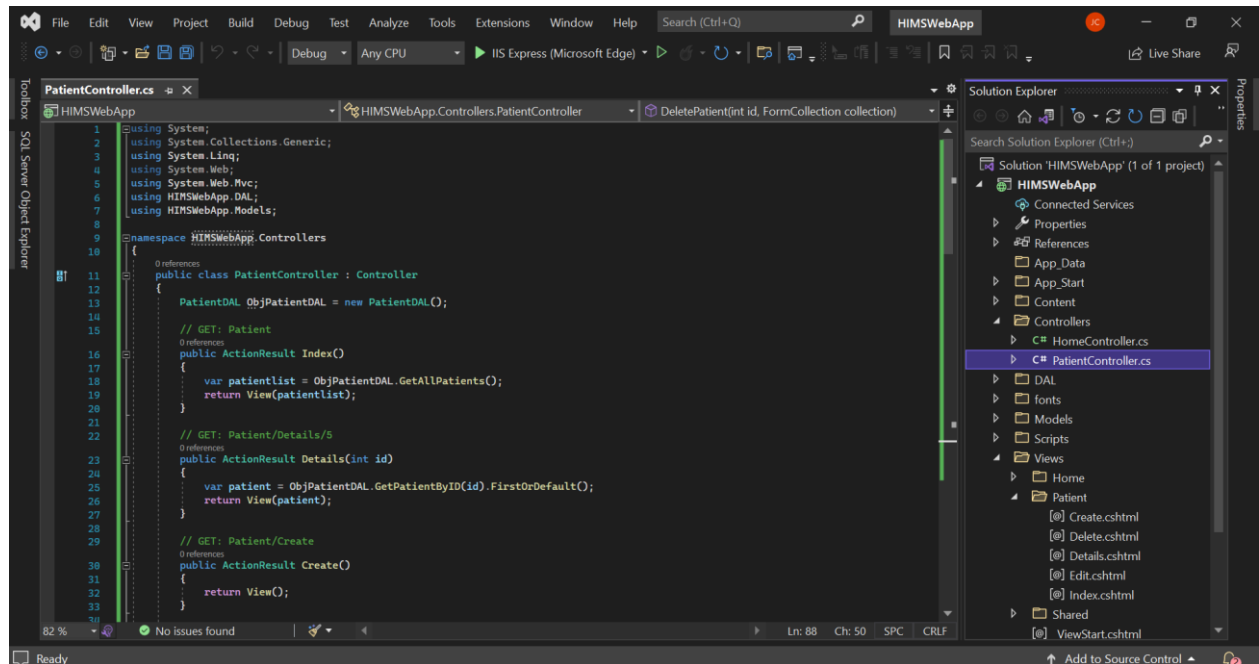
```

```

        command.ExecuteNonQuery();
        connection.Close();
    }
    return true;
}
}
}

```

## Creation of “PatientController.cs”



## PatientController.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using HIMSSWebApp.DAL;
using HIMSSWebApp.Models;

namespace HIMSSWebApp.Controllers
{
    public class PatientController : Controller
    {
        PatientDAL ObjPatientDAL = new PatientDAL();

        // GET: Patient
        public ActionResult Index()
        {
            var patientlist = ObjPatientDAL.GetAllPatients();
            return View(patientlist);
        }
    }
}

```

```

}

// GET: Patient/Details/5
public ActionResult Details(int id)
{
    var patient = ObjPatientDAL.GetPatientByID(id).FirstOrDefault();
    return View(patient);
}

// GET: Patient/Create
public ActionResult Create()
{
    return View();
}

// POST: Patient/Create
[HttpPost]
public ActionResult Create(Patient objPatient)
{
    try
    {
        // TODO: Add insert logic here
        ObjPatientDAL.InsertPatients(objPatient);
        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}

// GET: Patient/Edit/5
public ActionResult Edit(int id)
{
    var patient = ObjPatientDAL.GetPatientByID(id).FirstOrDefault();
    return View(patient);
}

// POST: Patient/Edit/5
[HttpPost, ActionName("Edit")]
public ActionResult Edit(Patient ObjPatient)
{
    try
    {
        // TODO: Add update logic here
        ObjPatientDAL.UpdatePatients(ObjPatient);
        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}

```

```

    }
}

// GET: Patient/Delete/5
public ActionResult Delete(int id)
{
    var patient = ObjPatientDAL.GetPatientByID(id).FirstOrDefault();
    return View(patient);
}

// POST: Patient/Delete/5
[HttpPost, ActionName("Delete")]
public ActionResult DeletePatient(int id, FormCollection collection)
{
    try
    {
        // TODO: Add delete logic here
        ObjPatientDAL.DeletePatients(id);
        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}
}
}
}

```

## CONCLUSION

In conclusion, HealthCare Information Management System (HIMS) using Web-based and Database technologies have the potential to revolutionize healthcare delivery by enabling real-time monitoring of patient data, secure and centralized data storage, and efficient data sharing among healthcare providers.

The results of a project on HealthCare Information Management System (HIMS) using Web-based and Database technologies can be analyzed based on system performance, user satisfaction, and healthcare outcomes. This information can provide valuable insights into the effectiveness of the system and its impact on healthcare delivery.

Overall, smart HealthCare Information Management System (HIMS) using Web-based and Database technologies have the potential to transform healthcare delivery by providing a secure, efficient, and comprehensive approach to healthcare monitoring. With further development and refinement, these systems have the potential to improve healthcare outcomes and reduce healthcare costs, leading to better healthcare delivery for patients and providers alike.

## REFERENCES

Here are some references related to the project "Healthcare Management Information Systems":

**Microsoft SQL Server:** <https://learn.microsoft.com/en-us/sql/?view=sql-server-ver16>

**ASP.NET:** <https://dotnet.microsoft.com/en-us/apps/aspnet>

## APPENDIX

Provides additional information on the project "Smart HealthCare Information Management System (HIMS)" This project aims to develop a system that integrates Web-based and database technologies to improve healthcare monitoring, data security, and transparency.

### **Architecture:**

Smart HealthCare Information Management System (HIMS)" using Web-based and Database has a layered architecture consisting of three layers: the User Interface layer, the Middleware layer, the Data Access Layer.

### **Implementation:**

The HealthCare Information Management System (HIMS) using Web-based and Database uses various hardware and software components. The hardware components include database server that collect data from patients. The software components include a Web application, and database software.

### **Potential Benefits:**

The Smart HealthCare Information Management System (HIMS) using Web-based and Database has various potential benefits, including improved healthcare monitoring, data security, and transparency. By integrating Web-based and Database technologies, the system can collect real-time data from patients and store it in a centralized database. This ensures the integrity and confidentiality of patient data and prevents unauthorized access, modification, or deletion of data. Additionally, the system enables authorized users, such as healthcare providers and patients, to access and interact with the data transparently and securely. This can lead to improved healthcare outcomes, reduced costs, and better patient satisfaction.