# SIX-MONTHS INDUSTRIAL TRAINING

**at**

**WIPRO LIMITED**

**Report**

**Submitted in partial fulfilment of the requirements for the award of degree of**

**Bachelor of Technology**

**In**

**Electronics and communication Engineering**


**Submitted By:**

**Kritika**

**University Roll No.: 1803775**

**Department of Electronics and Communication Engineering,**

**DAV Institute of Engineering and Technology, Jalandhar.**

**Jan-June,2022**

# OFFER LETTER



## APPOINTMENT LETTER

January 21, 2022

Dear Kritika Grover,

Welcome to Wipro Limited (Company/Wipro) and congratulations on your appointment as **Project Engineer.** The terms of your employment with the Company is listed below. Please be informed that the terms may be modified pursuant to changes in the Company policy updated from time to time.

**1. Appointment Details**:

a. The date of appointment is effective from the date of joining, unless otherwise communicated in writing by the Company.

b. You will be on probation for a period of 12 months from the date of your appointment. On completion of the probation period, your appointment shall be confirmed at the discretion of the Company based on your performance and other criteria as applicable to your band and stream. Unless confirmation is communicated in writing, your probation period shall be deemed to have been extended.

c. The retirement age is 58 years.

d. You may be transferred to any other location, department, function, establishment, or branch of the Company or subsidiary, associate or affiliate company ,in such capacity as the Company may from time to time determine. In such a case, you will be governed by the terms and conditions of service applicable to the new assignment including compensation, working hours, holidays, leave, people policies, etc.

e. We provide support to our global customers through various Company locations in India to suit customer requirements by operating 24x7. You would be operating from any of these locations and in any of the shifts, including night shift, as may be decided by the Company, keeping in mind the business needs and deliverables to customers.

f. This offer of appointment is subject to your successful completion of all curricular requirements as laid down by the University/Institution for award of the degree/diploma and the requirements, including aggregate, specified by the Company for your role, and any other criteria specified by the Company in terms of your educational qualifications on/before the date of appointment.

g. The copy of this letter duly signed by you, has to be mandatorily submitted on the date of joining.

**2. Compensation**:

You will be eligible for:

a. Compensation and benefits in accordance with Annexure III - Salary Offer Sheet.

b. Variable Pay - The details of this component are listed in Annexure VI. The Variable Pay program may be changed or modified in part or full thereof from time to time, at the sole discretion of the Company.

c. Other compensation and benefits in accordance with Company policy as modified and intimated to you from time to time.

d. Your salary will be reviewed periodically as per Company policy.

e. Changes in your compensation are at the Company's discretion and will be subject to and on the basis of your effective performance and the performance results of the Company during your period of employment and other

# DAV INSTITUTE OF ENGINEERING AND TECHNOLOGY, JALANDHAR

## DECLARATION

I Kritika, hereby declare that I have undertaken internship at "**WIPRO Limited"** under the mentorship of Mr. TVS Kaushik, in partial fulfillment of requirements for the award of degree of B.Tech. (Electronics and Communication Engineering) at DAV INSTITUTE OF ENGINEERING AND TECHNOLOGY, JALANDHAR.

The work which is being presented in the training report submitted to Department of Electronics and Communication Engineering at DAV Institute of Engineering and Technology, Jalandhar is an authentic record of training work.

Kritika

Signature of Examiner:

# ABSTRACT

The quest for knowledge can never end. The deeper you dig the greater the unexplored seems to be. No man can honestly say that he has learned all that this world has to offer. They can't achieve anything worthwhile in any field only on basis of theoretical knowledge from book, programming knowledge obtain through working at zero level and gaining experience in my view.

To attain tangible positive and concert result, the classroom knowledge needs to be effective worked to the realities of the situation outside the classroom.

During my B-Tech program, as a part of it, I joined **"Wipro Limited"** as a Software Engineer Intern.

This report covers the small introduction of **Wipro Limited** and complete description of my work as an intern there.

# ACKNOWLEDGEMENT

I am highly grateful to the Dr. Manoj Kumar, Principal, DAV Institute of Engineering & Technology, Jalandhar, for providing this opportunity to carry out the industrial training at Wipro Limited.

The constant guidance and encouragement received from Mrs. Neeru Malhotra, HoD Department of Electronics and Communication & Engineering, DAVIET Jalandhar, has been of great help in carrying out the project work and is acknowledged with reverential thanks.

I would like to express my gratitude to Mr. Vishav Kapoor, Manager, Department of Training & Placement, DAVIET Jalandhar, for his stimulating guidance, continuous encouragement, and supervision throughout the course of present work.

I would like to express a deep sense of gratitude and thank profusely to my mentor at WIPRO Limited, Mr. TVS Kaushik. Without his wise counsel and able guidance, it would have been impossible to complete the report in this manner.

I express gratitude to other faculty members of Electronics and Communication department of DAVIET for their intellectual support throughout the course of this work.

Kritika

# TABLE OF CONTENTS

# LIST OF FIGURES

# 1.Company's profile:


Fig.1. WIPRO Limited

Wipro Limited (formerly, Western Indian Palm Refined Oils Limited) is an Indian multinational corporation that provides information technology, consulting, and business process services. Delaporte has served as CEO and managing director of Wipro from July 2020. It is headquartered in Bangalore, Karnataka, India. Wipro's capabilities range across cloud computing, cyber security, digital transformation, artificial intelligence, robotics, data analytics, and other technology consulting services to customers in 67 countries.

Wipro Limited is a leading global information technology, consulting, and business process services company. It harnesses the power of cognitive computing, hyper-automation, robotics, cloud, analytics, and emerging technologies to help their clients adapt to the digital world and make them successful. A company recognized globally for its comprehensive portfolio of services, strong commitment to sustainability and good corporate citizenship, having over 220,000 dedicated employees serving clients across six continents.

Vision: The vision of Wipro includes being a leader in areas of business, customer, and people.

• The company aims to be among the best IT company in India and the world.

• Its goals also include being the first choice among customers and employees.

Mission: To be RF system solution provider, through its innovative Research & Design works for a new world of Broadband Wireless Communications.

# 2.Introduction to the technologies to be used throughout the Internship:

## 2.1: Introduction to basic technologies:


Fig. 2. Dot net

**.NET:**

.NET (pronounced as "dot net"; previously named .NET Core) is a free and open-source, managed computer software-framework for Windows, Linux, and macOS operating systems. It is a cross-platform successor to .NET Framework. The project is primarily developed by Microsoft employees by way of the .NET Foundation, and released under the MIT License.

On November 12, 2014, Microsoft announced .NET Core, in an effort to include cross-platform support for .NET, including Linux and macOS, source for the .NET Core CLR implementation, source for the "entire [...] library stack" for .NET Core, and the adoption of a conventional open-source development model under the stewardship of the .NET Foundation. NET fully supports C# and F# and supports Visual Basic .NET .

.NET supports four cross-platform scenarios: ASP.NET Core web apps; command-line/console apps; libraries; and Universal Windows Platform apps. Prior to .NET Core 3.0, it did not implement Windows Forms or Windows Presentation Foundation (WPF), which render the standard GUI for desktop software on Windows.

The two main components of .NET are CoreCLR and CoreFX, which are comparable to the Common Language Runtime (CLR) and the Framework Class Library (FCL) of the .NET Framework's Common Language Infrastructure (CLI) implementation.As a CLI implementation of the foundational Standard Libraries, CoreFX shares a subset of .NET Framework APIs, however, it also comes with its own APIs that are not part of the .NET Framework. The .NET command-line interface offers an execution entry point for operating systems and provides developer services like compilation and package management.

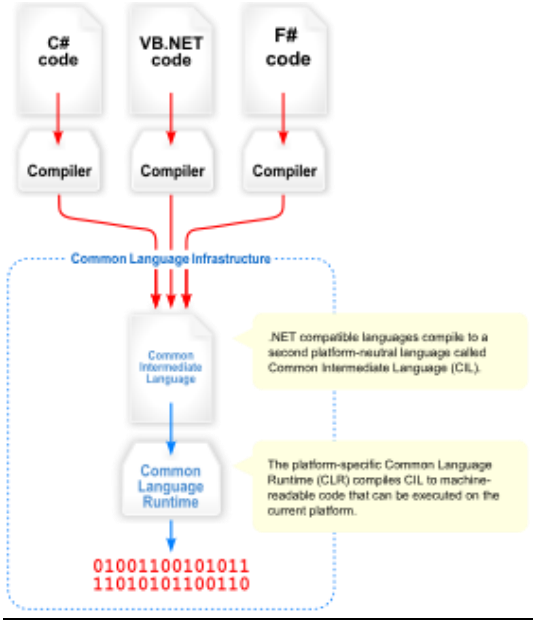



Fig.3. NET uses CLI

**C#:**


Fig.4. C#

C# (pronounced "See Sharp") is a modern, object-oriented, and type-safe programming language. C# enables developers to build many types of secure and

robust applications that run in .NET. C# has its roots in the C family of languages and will be immediately familiar to C, C++, Java, and JavaScript programmers. C# is an object-oriented, component-oriented programming language. C# provides language constructs to directly support these concepts, making C# a natural language in which to create and use software components.

Several C# features help create robust and durable applications. Garbage collection automatically reclaims memory occupied by unreachable unused objects. Nullable types guard against variables that don't refer to allocated objects. Exception handling provides a structured and extensible approach to error detection and recovery. Lambda expressions support functional programming techniques. Language Integrated Query (LINQ) syntax creates a common pattern for working with data from any source. Language support for asynchronous operations provides syntax for building distributed systems. C# has a unified type system. All C# types, including primitive types such as int and double, inherit from a single root object type. All types share a set of common operations. Values of any type can be stored, transported, and operated upon in a consistent manner. Furthermore, C# supports both user-defined reference types and value types. C# allows dynamic allocation of objects and in-line storage of lightweight structures. C# supports generic methods and types, which provide increased type safety and performance. C# provides iterators, which enable implementers of collection classes to define custom behaviours for client code.

**C# and DOTNET:**

C# programs run on .NET, a virtual execution system called the common language runtime (CLR) and a set of class libraries. The CLR is the implementation by Microsoft of the common language infrastructure (CLI), an international standard. The CLI is the basis for creating execution and development environments in which languages and libraries work together seamlessly.

Source code written in C# is compiled into an intermediate language (IL) that conforms to the CLI specification. The IL code and resources, such as bitmaps and strings, are stored in an assembly, typically with an extension of .dll.

When the C# program is executed, the assembly is loaded into the CLR. The CLR performs Just-In-Time (JIT) compilation to convert the IL code to native machine

instructions. Code that's executed by the CLR is sometimes referred to as "managed code." "Unmanaged code," is compiled into native machine language that targets a specific platform. Language interoperability is a key feature of .NET. IL code produced by the C# compiler conforms to the Common Type Specification (CTS). IL code generated from C# can interact with code that was generated from the .NET versions of F#, Visual Basic, C++.

**SQL and RDBMS:**

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

What it can do -

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

**RDBMS** stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access. The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries, and it consists of columns and rows.

Every table is broken up into smaller entities called fields A field is a column in a table that is designed to maintain specific information about every record in the table.

A record, also called a row, is each individual entry that exists in a table. A record is a horizontal entity in a table. A column is a vertical entity in a table that contains all information associated with a specific field in a table.

**Microsoft Visual Studio:**

Fig.5. Visual Studio

It is known to be the best comprehensive IDE (Integrated Development Environment) for .NET and C++ developers on Windows. Fully packed with an array of tools and features to elevate and enhance every stage of software development. The first 64-bit IDE makes it easier to work with even bigger projects and more complex workloads.

IntelliCode of VS is a powerful set of automatic code completion tools that understand your code context: variable names, functions, and the type of code you're writing. This makes IntelliCode able to complete up to a whole line at once, helping you code more accurately and confidently. CodeLens of VS helps you easily find important insights, like what changes have been made, what those changes did, and whether you've run unit testing on your method.

**Microsoft SQL Server 2014 Management Studio:**


Fig.6. SSMS

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL Database. SSMS provides tools to configure, monitor, and administer instances of SQL Server and databases.

SQL Server Management Studio has the following components:

- Object Explorer
- Security
- Server Objects
- Query and Text Editor
- Template Explorer
- Solution Explorer
- Visual Database Tools

## 2.2: Introduction to C# .NET fundamentals-

- C# is case sensitive.
- All statements and expression must end with a semicolon (;).
- The program execution starts at the Main method.
- Unlike Java, program file name could be different from the class name.

A C# program consists of the following parts −

```
using System;

namespace HelloWorldApplication {
   class HelloWorld {
      static void Main(string[] args) {
         /* my first program in C# */
         Console.WriteLine("Hello World");
         Console.ReadKey();
      }
   }
}
```

Fig.7. C# basic program

- The first line of the program using System; - the using keyword is used to include the System namespace in the program. A program generally has multiple using statements.
- The next line has the namespace declaration. A namespace is a collection of classes. The *HelloWorldApplication* namespace contains the class *HelloWorld*.
- The next line has a class declaration, the class *HelloWorld* contains the data and method definitions that your program uses. Classes generally contain multiple methods. Methods define the behaviour of the class.
- The next line defines the **Main** method, which is the **entry point** for all C# programs. The **Main** method states what the class does when executed.
- The next line /*...*/ is ignored by the compiler and it is put to add **comments** in the program.
- *WriteLine* is a method of the *Console* class defined in the *System* namespace. This statement causes the message "Hello, World!" to be displayed on the screen.

**Various fundamentals of C#:**

**Data types:** C# is a strongly typed language. It means we must declare the type of a variable that indicates the kind of values it is going to store, such as integer, float, decimal, text, etc.

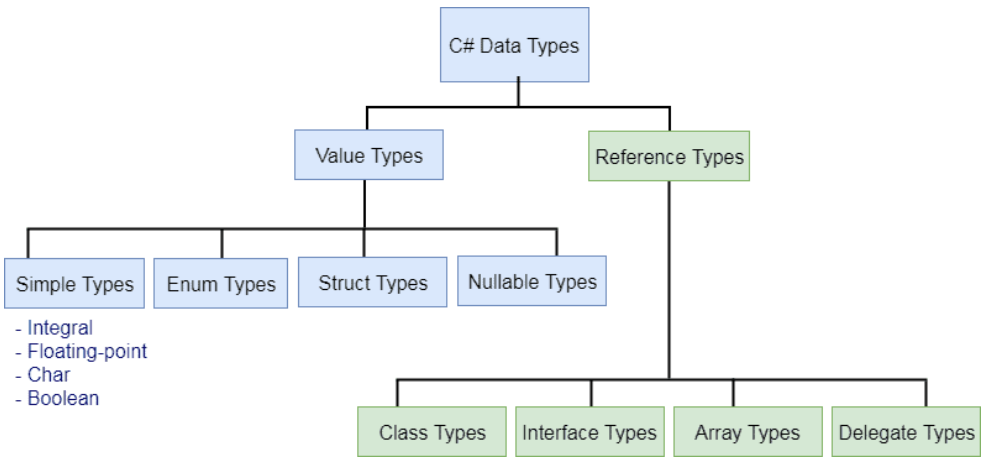



Fig.8. Datatypes in C#

**Operators:** Operators in C# are some special symbols that perform some action on operands. C# includes the following categories of operators:

- Arithmetic operators: The arithmetic operators perform arithmetic operations on all the numeric type operands such as sbyte, byte, short, ushort, int, uint, long, ulong, float, double, and decimal.
- Assignment operators: The assignment operator = assigns its right had value to its left-hand variable, property, or indexer. It can also be used with other arithmetic, Boolean logical, and bitwise operators.
- Comparison operators: Comparison operators compare two numeric operands and returns true or false.
- Equality operators: The equality operator checks whether the two operands are equal or not.
- Boolean logical operators: The Boolean logical operators perform a logical operation on bool operands.

Evaluation of the operands in an expression starts from left to right. If multiple operators are used in an expression, then the operators with higher priority are evaluated before the operators with lower priority.

**Control statements:** Control statements give you additional means to control the processing within the applications you develop. This includes statements like-

If-then-else: The if statement has three forms: - single selection, if-then-else selection, and multicase selection.

Switch: the switch statement is like an if-else if-else form of an if statement.

While and Do-while loop: The while loop allows the user to repeat a section of code until a guard condition is met.

The do-while construct checks the condition at the end of the loop. Therefore, the do-while loop executes at least once even though the condition to be checked is false from the beginning.

For loop: The for loop is useful when you know how many times the loop needs to execute.

Foreach: The foreach statement allows the iteration of processing over the elements in arrays and collections. Within the foreach loop parentheses, the expression consists of two parts separated by the keyword in. To the right of in is the collection, and to the left is the variable with the type identifier matching whatever type the collection returns.

Goto: You can use the goto statement to jump to a specific segment of code. You can also use goto for jumping to switch cases and default labels inside switch blocks. You should avoid the overuse of goto because code becomes difficult to read and maintain if you have many goto jumps within your code.

Break: The break statement, used within for, while, and do-while blocks, causes processing to exit the innermost loop immediately. When a break statement is used, the code jumps to the next line following the loop block

Continue: The continue statement is used to jump to the end of the loop immediately and process the next iteration of the loop.

Return: The return statement is used to prematurely return from a method. The return statement can return empty or with a value on the stack, depending upon the return value definition in the method. Void methods do not require a return value. For other functions, you need to return an appropriate value of the type you declared in the method signature.

**Arrays:** An array stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type stored at contiguous memory locations.

It is declared as:  datatype [] arrayName;

## 2.3: Introduction to object-oriented programming (OOP)-

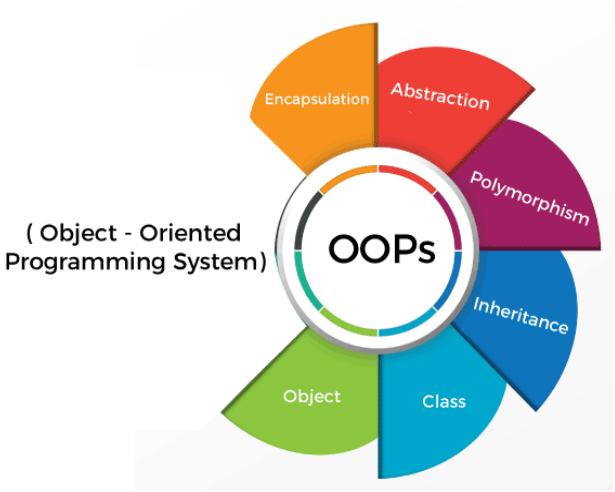



Fig.9. OOPs

Object-oriented programming (OOP) is the core ingredient of the .NET framework. The fundamental idea behind OOP is to combine into a single unit both data and the methods that operate on that data; such units are called an object. All OOP languages provide mechanisms that help you implement the object-oriented model. They are encapsulation, inheritance, polymorphism, and reusability.

**Encapsulation**-Encapsulation binds together code and the data it manipulates and keeps them both safe from outside interference and misuse. Encapsulation is a protective container that prevents code and data from being accessed by other code defined outside the container.

**Inheritance-**Inheritance is the process by which one object acquires the properties of another object. A type derives from a base type, taking all the base type members fields and functions. Inheritance is most useful when you need to add functionality to an existing type. For example, all .NET classes inherit from the System.Object class, so a class can include new functionality as well as use the existing object's class functions and properties as well.

**Polymorphism**-Polymorphism is a feature that allows one interface to be used for a general class of action. The specific action is determined by the exact nature of circumstances.

**Reusability**-Once a class has been written, created, and debugged, it can be distributed to other programmers for use in their own program. This is called reusability, or in .NET terminology this concept is called a component or a DLL. In OOP, however, inheritance provides an important extension to the idea of reusability. A programmer can use an existing class and without modifying it, add additional features to it.

**Classes** are special kinds of templates from which you can create **objects**. Each object contains data and methods to manipulate and access that data. The class defines the data and the functionality that each object of that class can contain. A class declaration consists of a class header and body. The class header includes attributes, modifiers, and the class keyword. The class body encapsulates the members of the class, that are the data members and member functions.

Attributes provide additional context to a class, like adjectives, for example the Serializable attribute. Accessibility is the visibility of the class. The default accessibility of a class is internal. Private is the default accessibility of class members.

**Methods and properties in C#:**

A method is a group of statements that together perform a task. Every C# program has at least one class with a method named Main.

To use a method, you need to −

- Define the method
- Call the method

<Access Specifier> <Return Type> <Method Name> (Parameter List) {

Method Body

}

Following are the various elements of a method −

- **Access Specifier** − This determines the visibility of a variable or a method from another class.

- **Return type** − A method may return a value. The return type is the data type of the value the method returns. If the method is not returning any values, then the return type is **void**.
- **Method name** − Method name is a unique identifier, and it is case sensitive. It cannot be same as any other identifier declared in the class.
- **Parameter list** − Enclosed between parentheses, the parameters are used to pass and receive data from a method. The parameter list refers to the type, order, and number of the parameters of a method. Parameters are optional; that is, a method may contain no parameters.
- **Method body** − This contains the set of instructions needed to complete the required activity.

**Properties** are named members of classes, structures, and interfaces. Member variables or methods in a class or structures are called **Fields**. Properties are an extension of fields and are accessed using the same syntax. They use **accessors** through which the values of the private fields can be read, written, or manipulated. A property is like a combination of a variable and a method, and it has two methods: a get and a set method.

## 2.4: Inheritance, polymorphism, exception handling and file handling:

**Inheritance:** In C#, inheritance allows us to create a new class from an existing class. It is a key feature of Object-Oriented Programming (OOP). The class from which a new class is created is known as the base class (parent or superclass). And the new class is called derived class (child or subclass) The derived class inherits the fields and methods of the base class. This helps with the code reusability in C#.

Example: - class Animal {

```
  // fields and methods
}
// Dog inherits from Animal
class Dog: Animal {
  // fields and methods of Animal
```

```
 // fields and methods of Dog
}
```

Types of inheritance-

There are the following types of inheritance: -

- Single Inheritance: - In single inheritance, a single derived class inherits from a single base class.

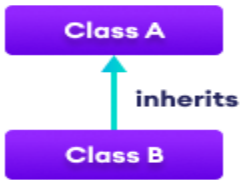



Fig.10.Single Inheritance

- Multilevel Inheritance: - In multilevel inheritance, a derived class inherits from a base and then the same derived class acts as a base class for another class.

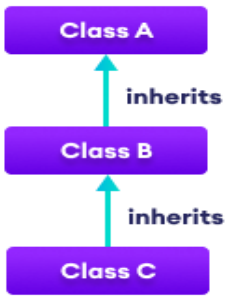



Fig.11.Multilevel inheritance

- Hierarchical Inheritance: - In hierarchical inheritance, multiple derived classes inherit from a single base class.

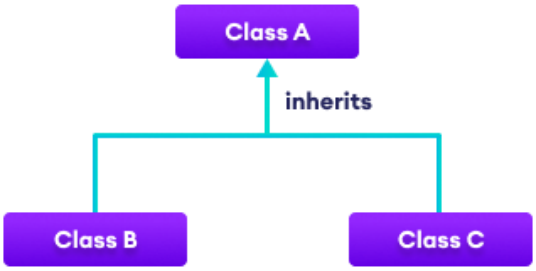



Fig.12.Hierarchical inheritance

- Multiple Inheritance: -In multiple inheritance, a single derived class inherits from multiple base classes. C# doesn't support multiple inheritance. However, we can achieve multiple inheritance through **interfaces. Interface** in C# is a blueprint of a class. It is like **abstract class** because all the methods which are declared inside the interface **are abstract methods.** It cannot have method body and cannot be instantiated. Abstract class is a restricted class that cannot be used to create objects (to access it, it must be inherited from another class). Abstract method can only be used in an abstract class, and it does not have a body. The body is provided by the derived class (inherited from).

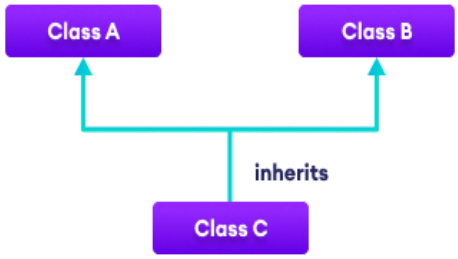



Fig 13. Multiple inheritance

- Hybrid Inheritance: -Hybrid inheritance is a combination of two or more types of inheritance. The combination of multilevel and hierarchical inheritance is an example of Hybrid inheritance.

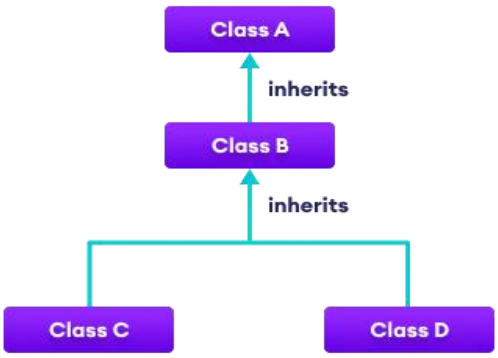



Fig.14. Hybrid inheritance

**Polymorphism:** Polymorphism provides the ability to a class to have multiple implementations with the same name. It is one of the core principles of OOP.

Types of Polymorphism: -

- Static / Compile Time Polymorphism.
- Dynamic / Runtime Polymorphism.

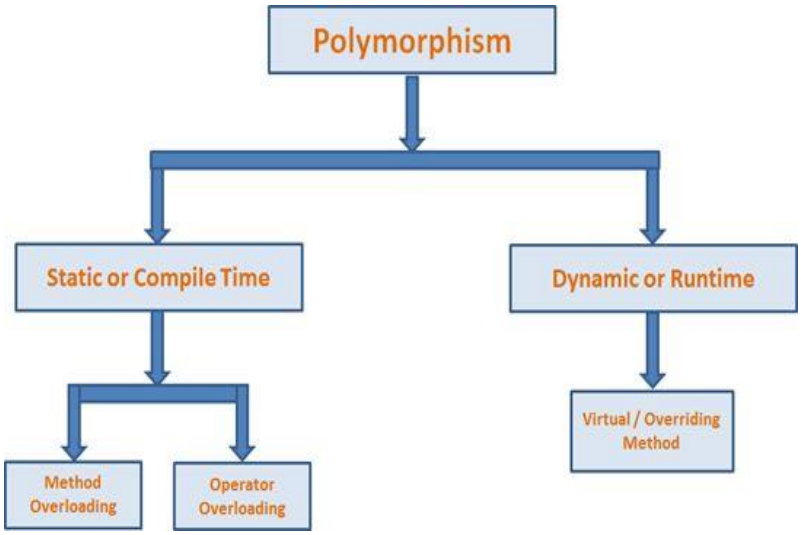



Fig 15. Polymorphism

Static polymorphism (or Early Binding) is a type of polymorphism that resolves at compile time. Method overloading is an example of static polymorphism. In method overloading, there are methods with the same name but different parameters. In other words, there are methods with the same name, but they have different data types and a

different number of arguments. Moreover, the method to call is determined at compile time.

Dynamic Polymorphism (or Late Binding) is a type of polymorphism that resolves at run time. Method overriding is an example of dynamic polymorphism. In method overriding, there are two classes: one is the parent class while the other is the child class. However, both classes have the same method. Moreover, the method in the child class overrides the method of the parent class.

**Exception handling in C#:** An exception is a problem that arises during the execution of a program. A C# exception is a response to an exceptional circumstance that arises while a program is running, such as an attempt to divide by zero. Exceptions provide a way to transfer control from one part of a program to another. C# exception handling is built upon four keywords: **try**, **catch**, **finally**, and **throw**.

- **try** − A try block identifies a block of code for which exceptions is activated. It is followed by one or more catch blocks.
- **catch** − A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The catch keyword indicates the catching of an exception.
- **finally** − The finally block is used to execute a given set of statements, whether an exception is thrown or not thrown.
- **Throw**-A program throws an exception when a problem shows up.

Syntax: -

```
try {
// statements causing exception
} catch( ExceptionName e1 ) {
// error handling code
} finally {
// statements to be executed
}
```

**File handling in C#:**

The term File Handling refers to the various operations like creating the file, reading from the file, writing to the file, appending the file, etc. There are two basic operation which is mostly used in file handling is reading and writing of the file. The file becomes stream when we open the file for writing and reading. A stream is a sequence of bytes which is used for communication. In C#, System.IO namespace contains classes which handle input and output streams and provide information about file and directory structure. Diagram to represent file-handling class hierarchy –

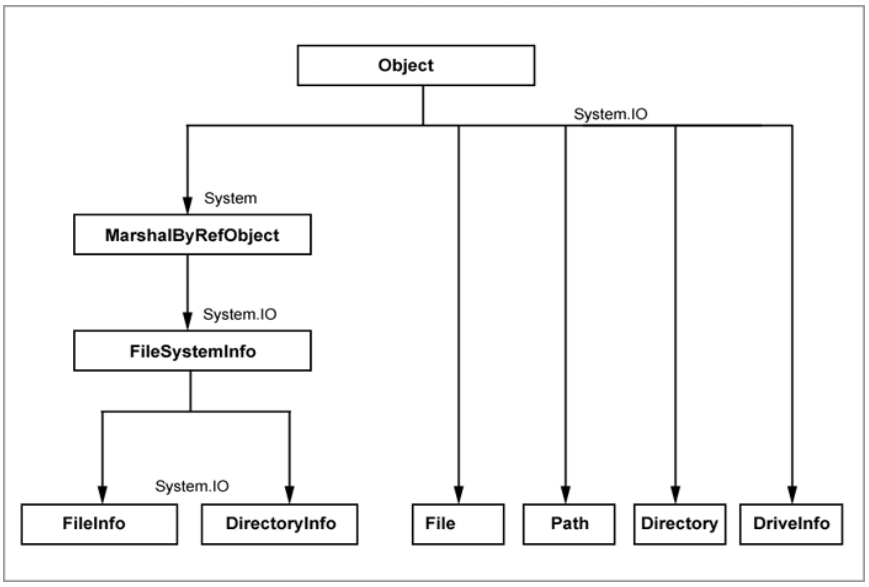



Fig.16. System.IO

Some commonly used classes in the System.IO namespace: -

FileStream: It is used to read from and write to any location within a file

BinaryReader: It is used to read primitive data types from a binary stream

BinaryWriter: It is used to write primitive data types in binary format

StreamReader: It is used to read characters from a byte Stream

StreamWriter: It is used to write characters to a stream.

StringReader: It is used to read from a string buffer

StringWriter: It is used to write into a string buffer

DirectoryInfo: It is used to perform operations on directories

FileInfo: It is used to perform operations on files

## 2.5: Collections:

**Collections:** C# includes specialized classes that store series of values or objects are called collections. There are two types of collections available in C#: non-generic collections and generic collections.

The System.Collections namespace contains the non-generic collection types and System.Collections.Generic namespace includes generic collection types. In most cases, it is recommended to use the generic collections because they perform faster than non-generic collections and minimize exceptions by giving compile-time errors.

Generic Collections-

| Generic Collections | Description |
|---|---|
| List<T> | Generic List<T> contains elements of specified type. It grows automatically as you add elements in it. |
| Dictionary<TKey,TValue> | Dictionary<TKey,TValue> contains key-value pairs. |
| SortedList<TKey,TValue> | SortedList stores key and value pairs. It automatically adds the elements in ascending order of key by default. |
| Queue<T> | Queue<T> stores the values in FIFO style (First In First Out). It keeps the order in which the values were added. It provides an Enqueue() method to add values and a Dequeue() method to retrieve values from the collection. |
| Stack<T> | Stack<T> stores the values as LIFO (Last In First Out). It provides a Push() method to add a value and Pop() & Peek() methods to retrieve values. |
| Hashset<T> | Hashset<T> contains non-duplicate elements. It eliminates duplicate elements. |

Fig.17.Generics

Non-generic Collections-

| Non-generic Collections | Usage |
|---|---|
| ArrayList | ArrayList stores objects of any type like an array. However, there is no need to specify the size of the ArrayList like with an array as it grows automatically. |
| SortedList | SortedList stores key and value pairs. It automatically arranges elements in ascending order of key by default. C# includes both, generic and non-generic SortedList collection. |
| Stack | Stack stores the values in LIFO style (Last In First Out). It provides a Push() method to add a value and Pop() & Peek() methods to retrieve values. C# includes both, generic and non-generic Stack. |
| Queue | Queue stores the values in FIFO style (First In First Out). It keeps the order in which the values were added. It provides an Enqueue() method to add values and a Dequeue() method to retrieve values from the collection. C# includes generic and non-generic Queue. |
| Hashtable | Hashtable stores key and value pairs. It retrieves the values by comparing the hash value of the keys. |
| BitArray | BitArray manages a compact array of bit values, which are represented as Booleans, where true indicates that the bit is on (1) and false indicates the bit is off (0). |

Fig. 18. Non-generics

## 2.6: Overview of Unit testing using NUNIT Framework:

**UNIT: -**

In this IT world a unit refers to simply the smallest piece of code which takes an input, does certain operation, and gives an output. And testing this small piece of code is called Unit Testing.

A lot of unit test frameworks are available for .Net nowadays, if we check in Visual Studio, we have MS Test from Microsoft integrated in Visual Studio.

Some 3rd party frameworks are:

- NUnit
- MbUnit

Out of all these Nunit is the most-used testing Framework.

**What Is NUnit?**


Fig.19. NUNIT

NUnit is a unit-testing framework for all .Net languages. NUnit is Open-Source software and NUnit 3.0 is released under the MIT license. This framework is very easy to work with and has user friendly attributes for working.

## 2.7: Overview of Advanced C#:

**Delegates:** The delegate is a reference type data type that defines the method signature. You can define variables of delegate, just like other data type, that can refer to any method with the same signature as the delegate.

There are three steps involved while working with delegates:

- Declare a delegate
- Set a target method
- Invoke a delegate

A delegate can be declared using the delegate keyword followed by a function signature, as shown below.

[access modifier] delegate [return type] [delegate name] ([parameters])

**Reflection in C#: -** Reflection provide metadata information on types, modules, assemblies etc. at runtime.

Uses of Reflection: - Some of the situations when reflections are useful in C# are given as follows:

1. Reflections are quite useful for creating new types at runtime.
2. It is easy to use reflection with the program metadata attributes.
3. Reflection is needed to examine and instantiate types in an assembly.

4. For late binding to methods and properties, reflections are quite useful.

System.Reflection Namespace: - The System.Reflection namespace is required to use reflections as it has classes that allow the user to get the information required about an application. It also allows the user to dynamically add objects, types, values etc. to the application.

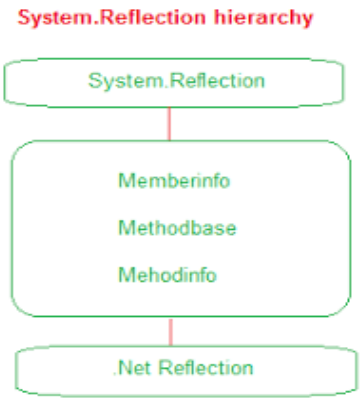



Fig.20. Reflections

## 2.8: Introduction to MS SQL server & RDBMS fundamentals:

**RDBMS: -**

RDBMS is the collection of programs and capabilities that enables the user to interact with a relational database. A relational database management system (RDBMS) is a type of DBMS with a row-based table structure. Most commercial RDBMSes use SQL.
Features of RDBMS-

- An RDBMS is easily accessible using SQL commands.
- An RDBMS provides full data independence.
- The basic unit of data storage in a relational database is called a table.
- A table consists of tuples/rows/records and each record has one or more columns used to store values.
- In RDBMS, we can use conditional operations such as joins and restrictions.
- An RDBMS enables data sharing between users.
- Also at the same time, you can ensure consistency of data across multiple tables by using integrity constraints.

- An RDBMS minimizes the redundancy of data.
- The structured query language (SQL) is used to communicate with RDBMS.

Advantages of RDBMS-

- Support for a very large database.
- Automatic optimization of searching (when possible).
- RDBMS has a simple view of the database that conforms to much of the data used in businesses.
- RDBMS uses Structured Query Language.
- Easy extendibility, as new data may be added without modifying existing records this is also known as scalability.
- RDBMS has data security which is critical when data sharing is based on privacy.
- RDBMS defines how the data is organized and how the relations among them are associated.
- It defines the entities and relationships among them. It contains a descriptive detail of the database.

Disadvantages of RDBMS-

- No support for complex objects such as documents, video, images.
- Often poor support for storage of complex objects.
- Still no efficient and effective integrated support.

**Introduction to SQL:**

- SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDMS).
- It is a standard language for Relational Database System. It enables a user to create, read, update, and delete relational databases and tables.
- All the RDBMS like MySQL, Informix, Oracle, MS Access, and SQL Server use SQL as their standard database language.
- SQL allows users to query the database in several ways, using English-like statements.

- Structure query language is not case sensitive. Generally, keywords of SQL are written in uppercase.
- Statements of SQL are dependent on text lines.

Types of SQL commands: -

i. Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL is auto committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

ii. Data Manipulation Language

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto committed that means it can't permanently save all the changes in the database. They can be rollback.

Here are some commands that come under DML:

- INSERT
- UPDATE
- DELETE

iii. Data Control Language

DCL commands are used to grant and take back authority from any database user.

Here are some commands that come under DCL:

- Grant
- Revoke

iv. Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only. These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

- COMMIT
- ROLLBACK
- SAVEPOINT

v. Data Query Language

DQL is used to fetch the data from the database.

It uses only one command:

- SELECT

Datatypes in SQL: -

SQL Data Types define the type of value that can be stored in a table column.

SQL data types can be broadly divided into following categories.

- Numeric data types such as int, tinyint, bigint, float, real, etc.
- Date and Time data types such as Date, Time, Datetime, etc.
- Character and String data types such as char, varchar, text, etc.
- Unicode character string data types, for example nchar, nvarchar, ntext, etc.
- Binary data types such as binary, varbinary, etc.
- Miscellaneous data types – clob, blob, xml, cursor, table, etc.

Built-in functions: - SQL Server has many built-in functions.

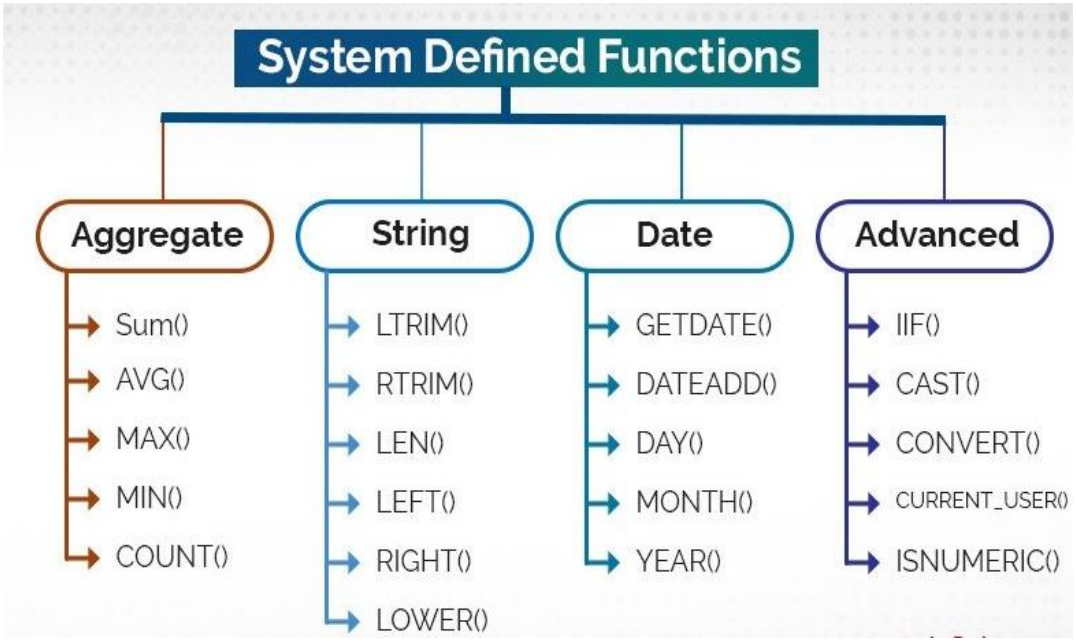



Fig.21.SQL Built-in functions

Example: -

```
SELECT COUNT(CustomerID),Country
FROM Customers
GROUP BY Country;
```

**2.9: ASP.NET Core:**


Fig.22.ASP.NET

ASP.NET Core is the new web framework from Microsoft. It has been redesigned from the ground up to be fast, flexible, modern, and work across different platforms.

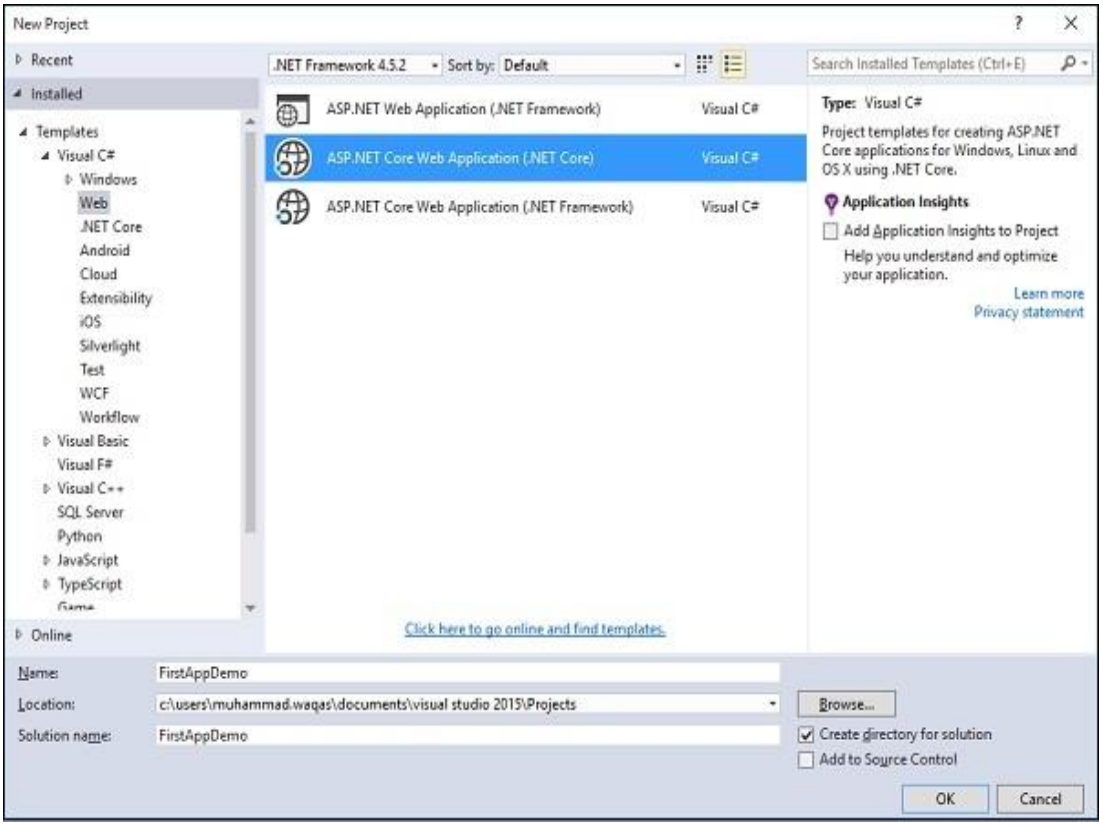
ASP.NET Core is an open source and cloud-optimized web framework for developing modern web applications that can be developed and run-on Windows, Linux, and the Mac. It includes the MVC framework, which now combines the features of MVC and Web API into a single web programming framework.

- ASP.NET Core apps can run on .NET Core or on the full .NET Framework.
- It was architected to provide an optimized development framework for apps that are deployed to the cloud or run on-premises.
- It consists of modular components with minimal overhead, so you retain flexibility while constructing your solutions.
- You can develop and run your ASP.NET Core apps cross-platform on Windows, Mac, and Linux.

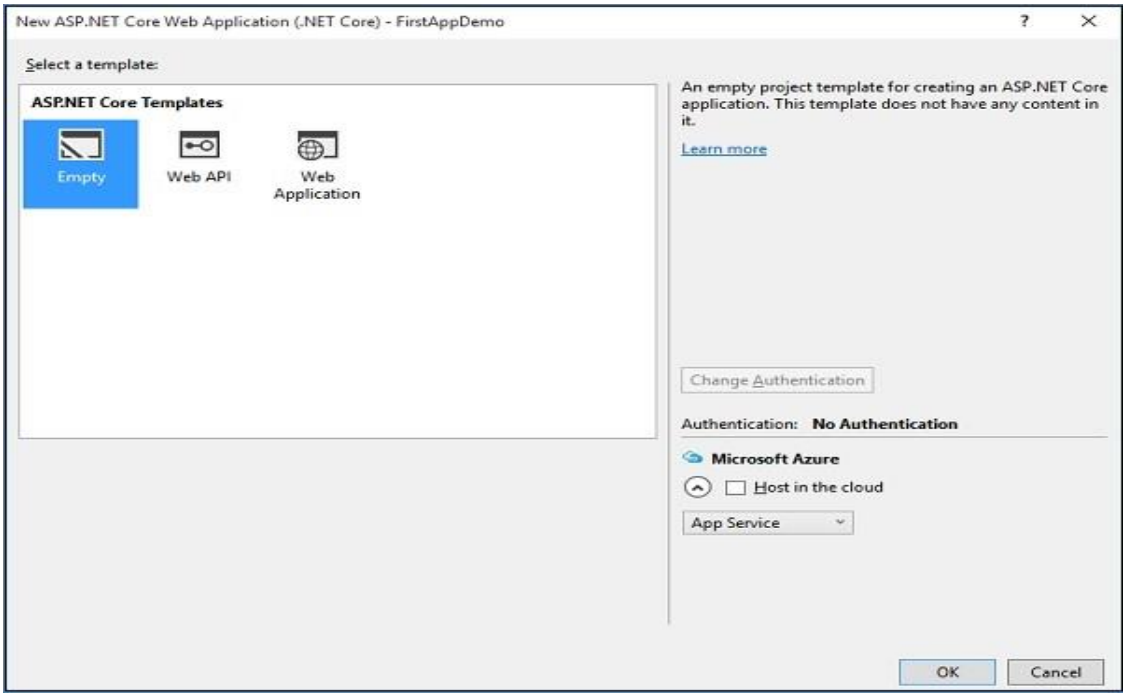
ASP.NET Core comes with the following advantages −

- ASP.NET Core has several architectural changes that result in a much leaner and modular framework.
- ASP.NET Core is no longer based on System.Web.dll. It is based on a set of granular and well factored NuGet packages.
- This allows you to optimize your app to include just the NuGet packages you need.
- The benefits of a smaller app surface area include tighter security, reduced servicing, improved performance, and decreased costs

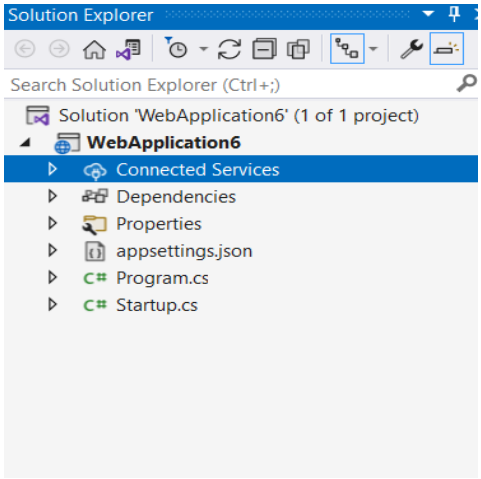
One can choose project of type ASP.NET in Visual Studio: -




After that one can choose the desired template: -




Asp.net web application fundamentals: -




- Connected services: The Visual Studio Connected Services feature lets app developers connect their applications to service providers that run in the cloud, or on premises. The Connected Services feature can automate the multiple steps it takes to connect a Visual Studio project to a service.

- Dependencies: The Dependencies in the ASP.NET Core 2.1 project contain all the installed server-side NuGet packages.

- Properties: The Properties node includes launchSettings.json, file which includes Visual Studio profiles of debug settings.

- wwwroot: By default, the **wwwroot** folder in the ASP.NET Core project is treated as a web root folder. Static files can be stored in any folder under the web root and accessed with a relative path to that root. In the standard ASP.NET application, static files can be served from the root folder of an application or any other folder under it. This has been changed in ASP.NET Core. Now, only those files that are in the web root - wwwroot folder can be served over an http request. All other files are blocked and cannot be served by default.
- Program.cs: ASP.NET Core web application is a console project which starts executing from the entry point. In Program class we can create a host for the web application.

- Startup.cs: ASP.NET Core application must include Startup class. As the name suggests, it is executed first when the application starts. The name

"Startup" is by ASP.NET Core convention. However, we can give any name to the Startup class, just specify it as the generic parameter in the UseStartup<T> () method.

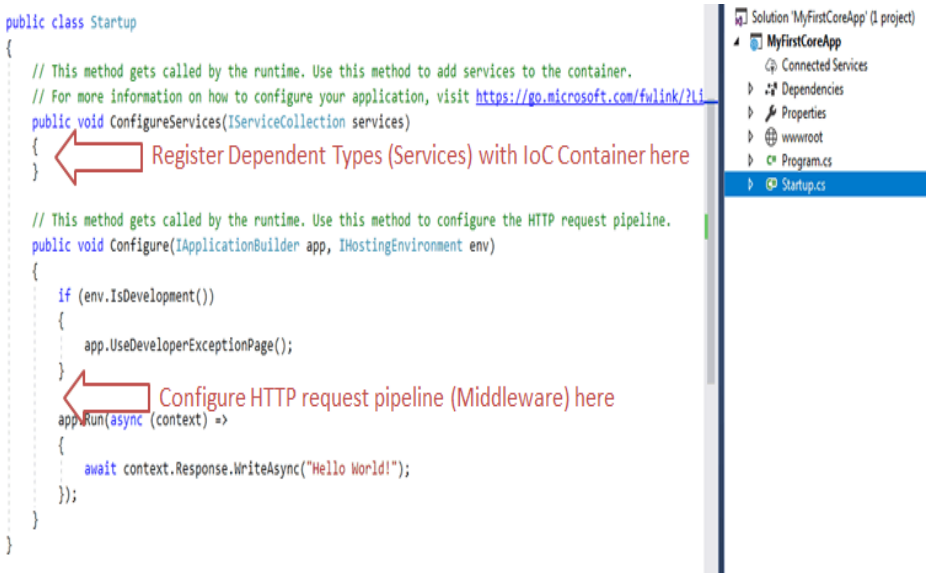



Fig.23.Startup.cs

ASP.NET Core – Middleware: -

ASP.NET Core introduced a new concept called **Middleware.** A middleware is nothing but a component (class) which is executed on every request in ASP.NET Core application. In the classic ASP.NET, HttpHandlers and HttpModules were part of request pipeline. Middleware is like HttpHandlers and HttpModules where both needs to be configured and executed in each request.

Typically, there will be multiple middleware in ASP.NET Core web application. It can be either framework provided middleware, added via NuGet or your own custom middleware. We can set the order of middleware execution in the request pipeline. Each middleware adds or modifies http request and optionally passes control to the next middleware component.

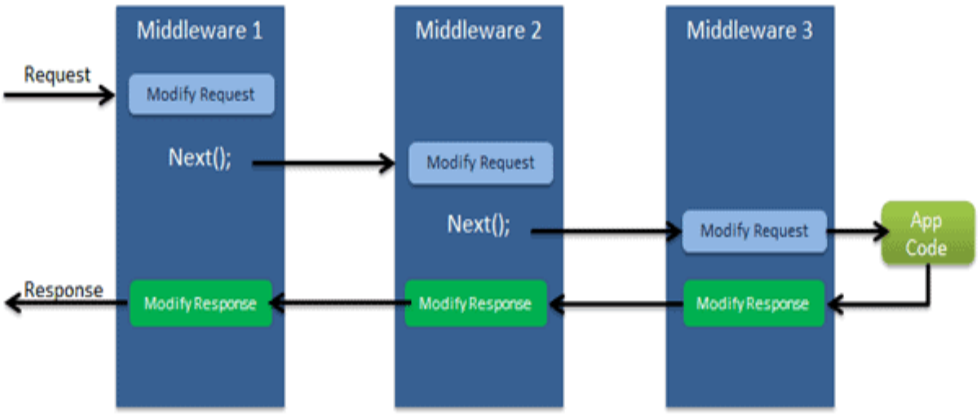



Fig.24.ASP.NET Core Middleware

Static files asp.net core: One of the most important features almost all web applications should have the ability to serve the static files directly from the file system. The static files such as HTML, Images, CSS, and JavaScript are the important assets of an application and ASP.NET Core can serve these files directly to the clients. By default, the ASP.NET Core cannot serve these static files. Some configuration is required to enable the ASP.NET Core to serve these static files directly.

In ASP.NET Core Application, the default directory or location for the static files is **wwwroot** (webroot) folder and moreover, this folder or directory should be present in the project root folder.

ASP.NET Core Request Processing Pipeline Execution: -

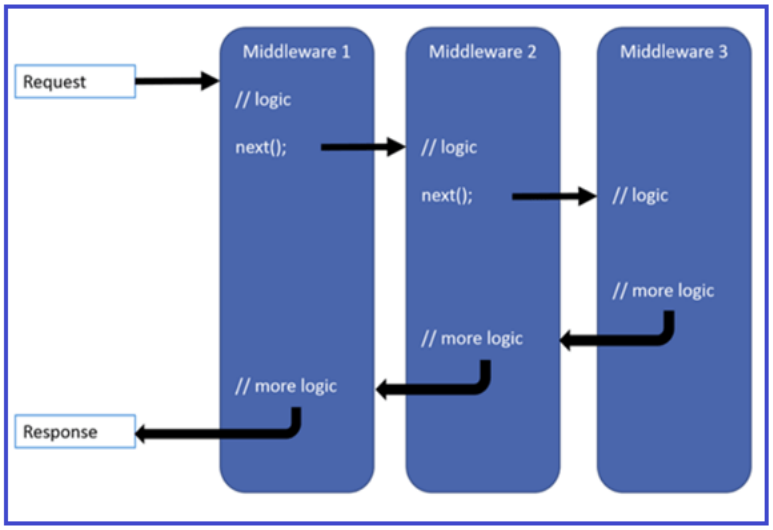



Fig.25.Middleware

When the incoming HTTP request comes, first it receives by the first middleware component i.e., Middleware1 which logs “Middleware1: Incoming Request” in the response stream. So as a result, first, we see this message first on the browser.
Once the first middleware logs the information, then it calls the next() method which will invoke the second middleware in the request processing pipeline i.e. Middleware2. The second middleware logs the information “Middleware2: Incoming Request” as a result we see this log information after the first log. Then the second middleware calls the next() which will invoke the third middleware in the request pipeline which is Middleware3.The third middleware handles the request and then produces the response. So, the third information that we see in the browser is “Middleware3: Incoming Request handled, and response generated”. This middleware component is registered using the Run() extension method, so it is a terminal component. So, from this point, the request pipeline starts reversing.

**Razor pages in asp.net core**: Razor pages are simple and introduce a page-focused framework that is used to create cross-platform, data-driven, server-side web pages with clean separation of concerns. Razor Pages is a newer, simplified web application programming model. It removes much of the ceremony of ASP.NET MVC by adopting a file-based routing approach. Razor Pages have an associated C# objected called the page model, which holds each page's behaviour.

Advantages of Razor Pages: -

1. It supports cross-platform; hence it can be deployed on Windows, Unix, and Mac operating systems.
2. It is easy to learn.
3. Lightweight and flexible framework so it can fit with any application you want to build.
4. Can work with C# programming language with Razor markup.
5. More organized with code behind page like asp.net web forms.

**Tag helpers in ASP.NET: -** Microsoft introduced a new feature in the MVC Razor engine with the release of ASP.NET Core which is known as Tag Helpers. With the help of Tag Helpers, developers can design their presentation layer using HTML tag

while they still can write business logic in the C# the code at server-side which will run in web server.

So, with the help of Tag Helpers which one is the Microsoft's new features in ASP.NET CORE, developers can replace the Razor cryptic syntax with @ symbol with a more natural looking HTML-like syntax.

<a asp-controller="Controller1" asp-action="CheckData" class="my-css-classname" my-attr="my-attribute">Click</a>

Advantages of Tag Helpers over HTML Helpers objects: -

- Tag Helpers use server-side binding without any server-side code.
- This helper object is very much use full when HTML developers do the UI designing who does not have any idea or concept about Razor syntax.
- It provides us an experience of working on basic HTML environments.
- It Supports rich IntelliSense environment support to create a markup between HTML and Razor.

**Asp.net core MVC: -**

MVC stands for Model, View, and Controller. MVC separates an application into three components - Model, View, and Controller.

**Model**: Model represents the shape of the data. A class in C# is used to describe a model. Model objects store data retrieved from the database.

**Model represents the data**.

**View**: View in MVC is a user interface. View display model data to the user and enables them to modify them. View in ASP.NET MVC is HTML, CSS, and some special syntax (Razor syntax) that makes it easy to communicate with the model and the controller.

**View is the User Interface.**

**Controller**: The controller handles the user request. Typically, the user uses the view and raises an HTTP request, which will be handled by the controller. The controller processes the request and returns the appropriate view as a response.

**Controller is the request handler.**

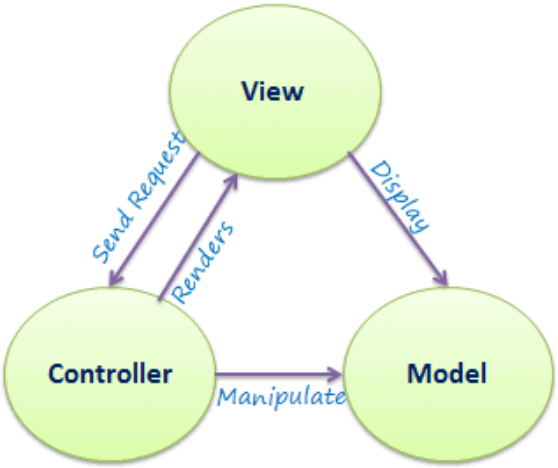
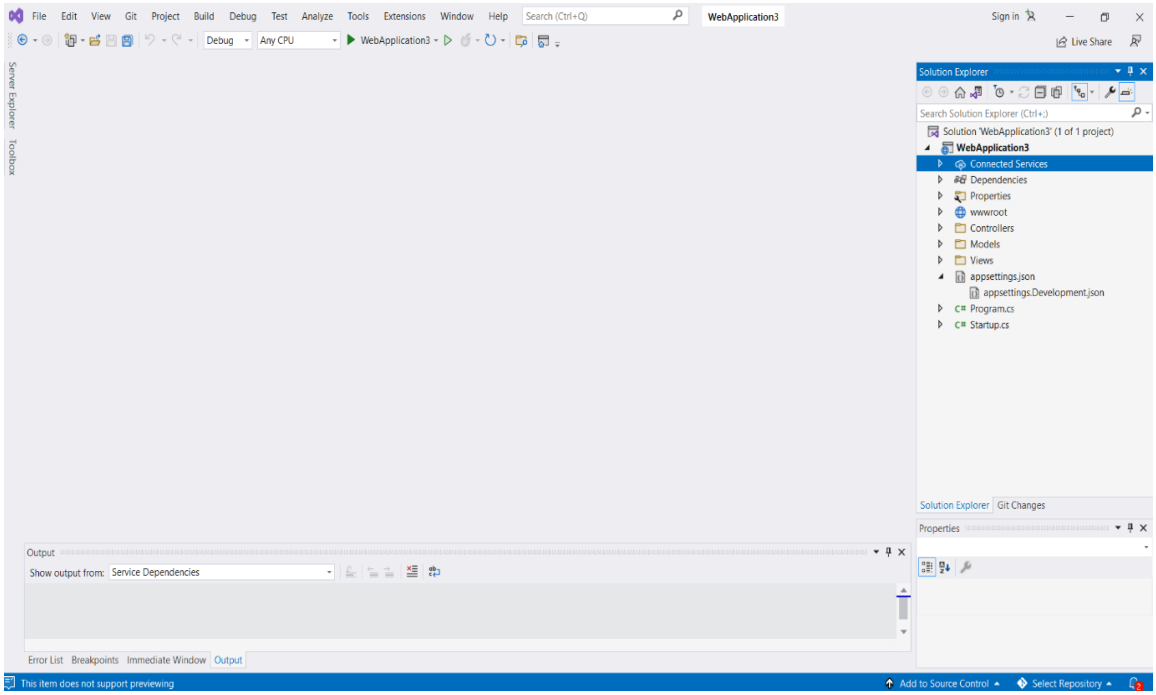



Fig.26.MVC

**Model binding** is a well-designed bridge between the HTTP request and the C# action methods. It makes it easy for developers to work with data on forms (views), because POST and GET is automatically transferred into a data model you specify. ASP.NET MVC uses default binders to complete this behind the scenes.

How the ASP.NET MVC Application looks-



## 2.10: Overview of web API core:

API stands for Application Programming Interface. It is an intermediate software agent that allows two or more applications to interact with each other.

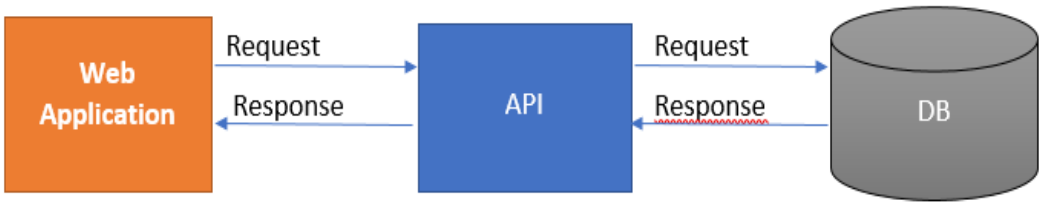



Fig.27.Web API

We can say that a web API is an application programming interface for a web application or web server. It uses HTTP protocol to communicate between clients and websites to have data access.

Asp.net Core web API is a cross-platform web API. The user wants to access the application from different devices like mobile, browser, Google devices, etc. In this case, Web API can be useful. Different devices request to Web API and Web API will respond in JSON format. Most of the devices can understand JSON output.

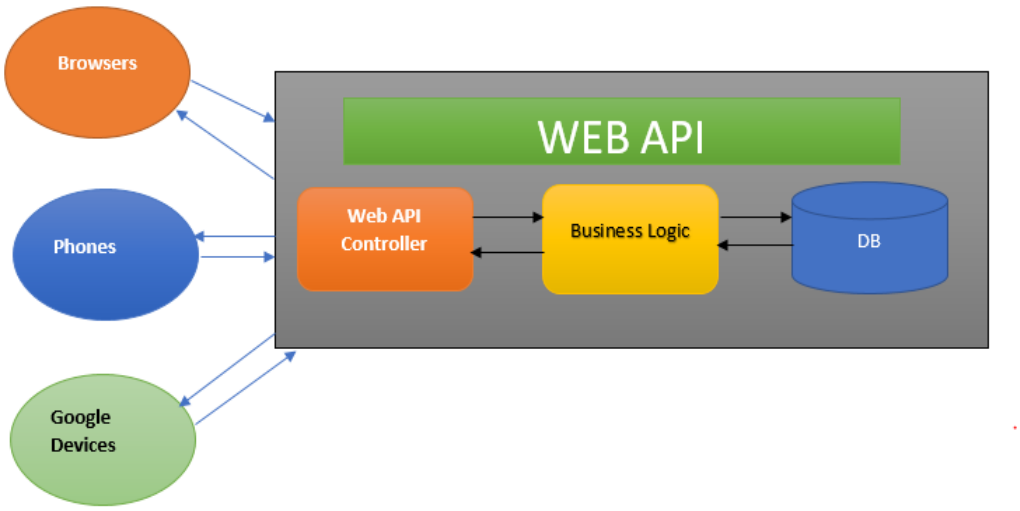
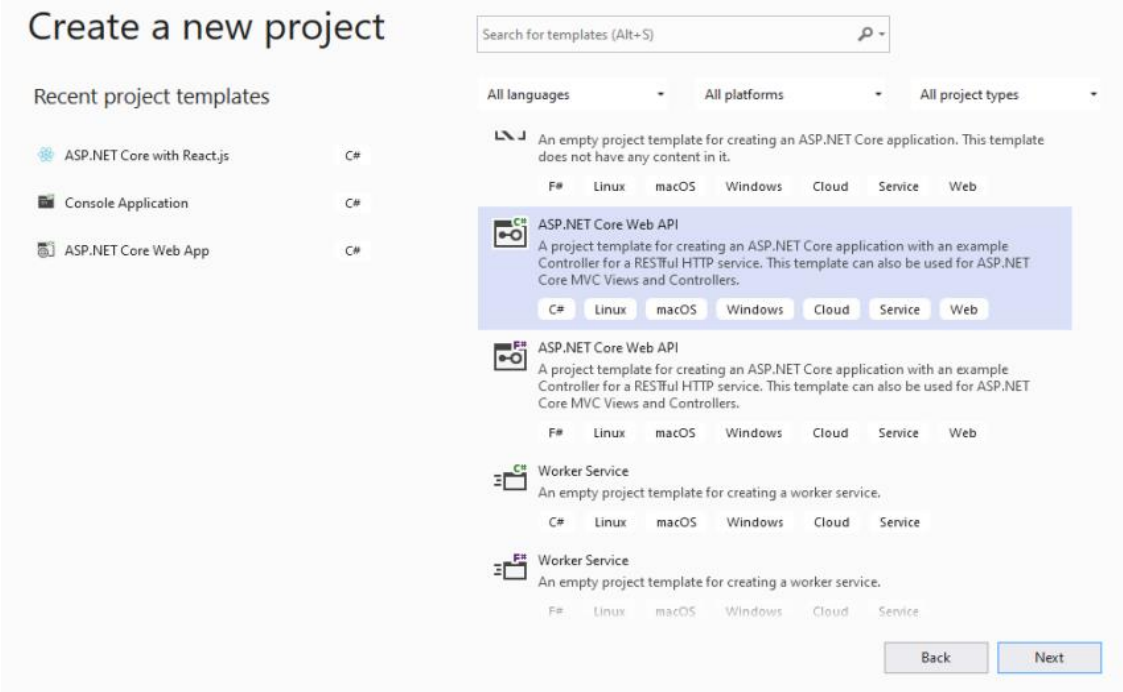



Fig.28.Web API

This diagram explains the architecture of Web API.

- A client called API/controller – In the above diagram Browsers, Phones, and Google Devices are called Web API Controllers.
- API/Controller interact with business layer and get Data from DB.
- The output will be returned in JSON format.

Select the “Asp.Net Core Web API” template-




Web API is mostly used for CRED (create, read, edit, delete) operations. It follows HTTP verbs for these operations.

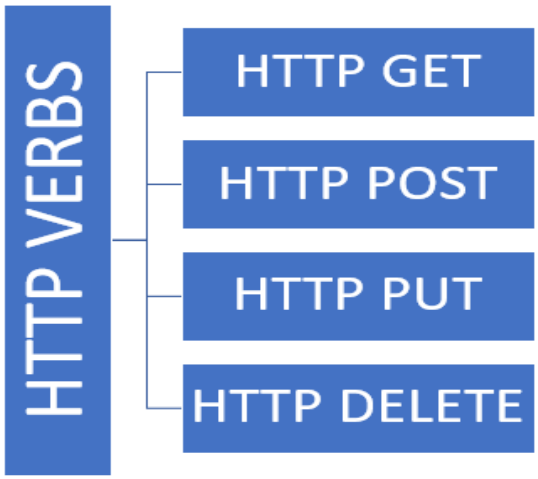



Fig.29.Web API

## 2.11.ADO.NET model and Entity Framework Core:

**ADO.NET** provides a bridge between the front-end controls and the back-end database. The ADO.NET objects encapsulate all the data access operations, and the controls interact with these objects to display data, thus hiding the details of movement of data.Ado.net is both connection oriented as well as disconnection oriented. Depending upon the functionality of an application, we can make it connection-oriented or disconnection oriented. We can even use both the modes together in a single application.

The following figure shows the ADO.NET objects briefly: -

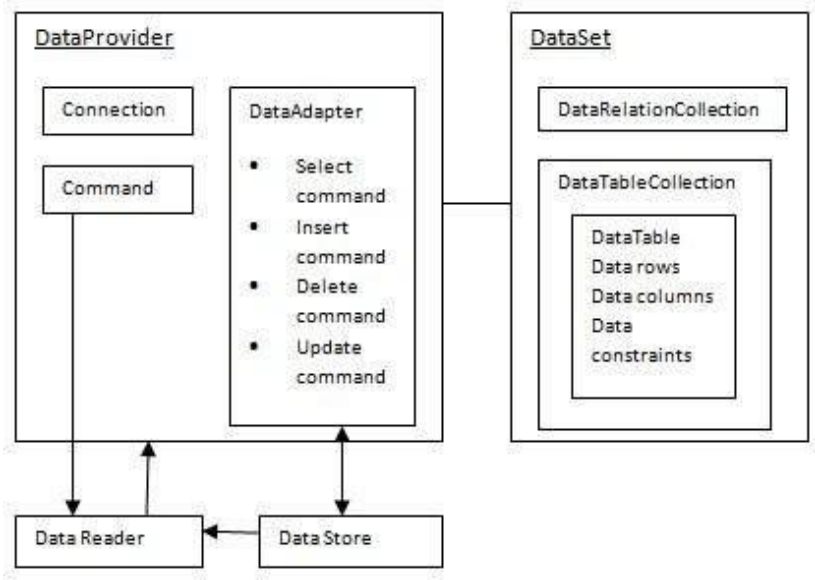



Fig.30.ADO.NET

The DataSet Class: -The dataset represents a subset of the database. It does not have a continuous connection to the database. To update the database a reconnection is required. The DataSet contains DataTable objects and DataRelation objects. The DataRelation objects represent the relationship between two tables.

The DataTable Class: -The DataTable class represents the tables in the database.

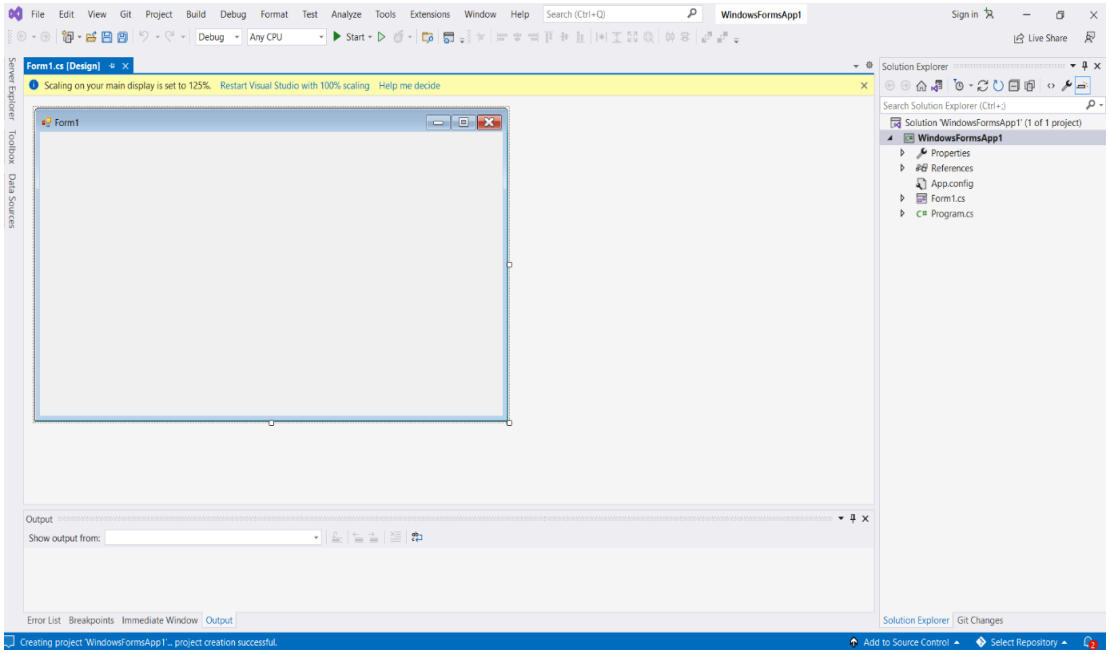
The DataRow Class: -The DataRow object represents a row in a table.

The DataAdapter Object: -The DataAdapter object acts as a mediator between the DataSet object and the database. This helps the Dataset to contain data from multiple databases or other data source.

The DataReader Object: - The DataReader object is an alternative to the DataSet and DataAdapter combination. This object provides a connection-oriented access to the data records in the database. These objects are suitable for read-only access, such as populating a list and then breaking the connection.

DbCommand and DbConnection Objects: - The DbConnection object represents a connection to the data source. The connection could be shared among different command objects.The DbCommand object represents the command, or a stored procedure sent to the database from retrieving or manipulating data.

We choose windows web form template to implement ADO.NET: –




**Entity framework in ADO.NET: -** Entity Framework is the development of data-oriented applications using ADO.NET. Entity Framework solves problems in entity models, relationships, and business logic. Also, it works with data engines.

This means Entity Framework is an Object-Relational Mapping (ORM) framework. An ORM structure provides database access and data operation process. Two approaches followed with EF are–

- Code First approach- In the Code-First approach, you focus on the domain of your application and start creating classes for your domain entity rather than

design your database first and then create the classes which match your database design.

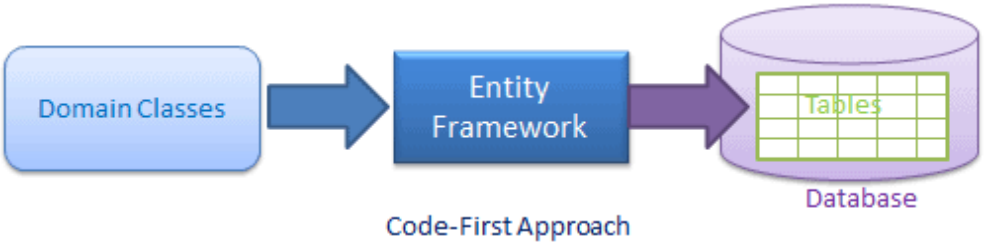



Fig.31.Code-first

EF API will create the database based on your domain classes and configuration. This means you need to start coding first in C# and then EF will create the database from your code.

- Database first approach- Database First Approach creates the Entity Framework from an existing database. It creates model codes from the database. The database in the project and those classes become the link between the database and controller.

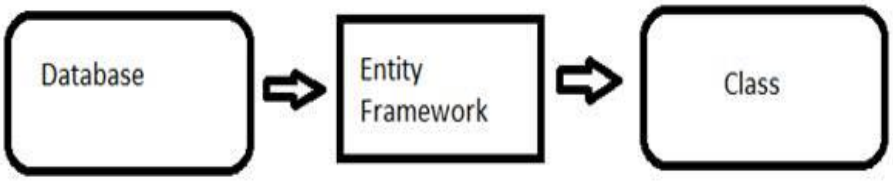



Fig.32.Database-first

## 2.12: Overview of Microservices:

The term microservices portrays a software development style that has grown from contemporary trends to set up practices that are meant to increase the speed and efficiency of developing and managing software solutions at scale. Microservices is more about applying a certain number of principles and architectural patterns as

architecture. Each microservice lives independently, but on the other hand, also all rely on each other. All microservices in a project get deployed in production at their own pace, on-premises on the cloud, independently, living side by side.

Microservices Architecture: -

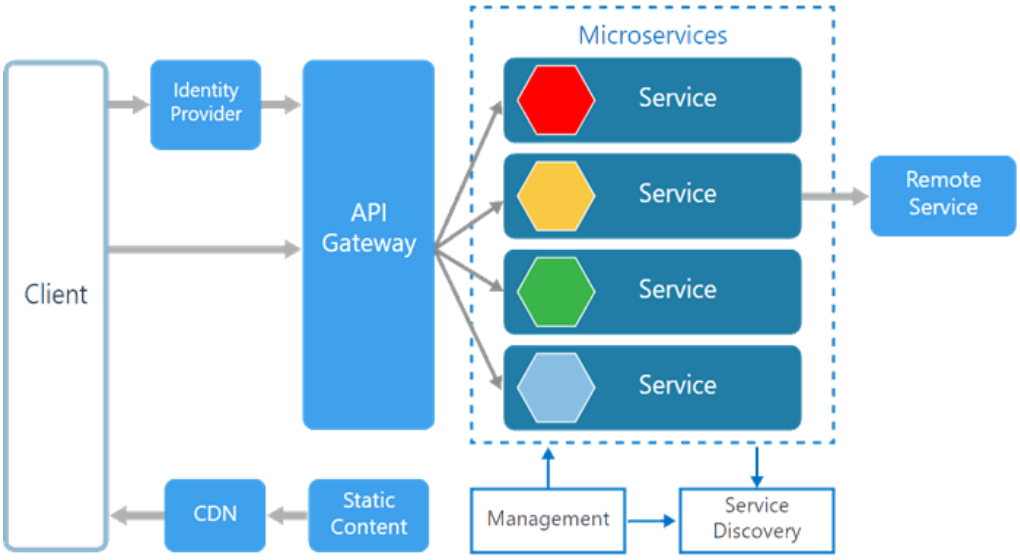



Fig.33.Microservices

There are various components in a microservices architecture apart from microservices themselves.

- Management. Maintains the nodes for the service.
- Identity Provider: Manages the identity information and provides authentication services within a distributed network.
- Service Discovery: Keeps track of services and service addresses and endpoints.
- API Gateway: Serves as client's entry point.
- CDN: A content delivery network to serve static resources for e.g., pages and web content in a distributed network
- Static Content: The static resources like pages and web content.

Microservices are deployed independently with their own database per service so the underlying microservices look as shown in the following picture.

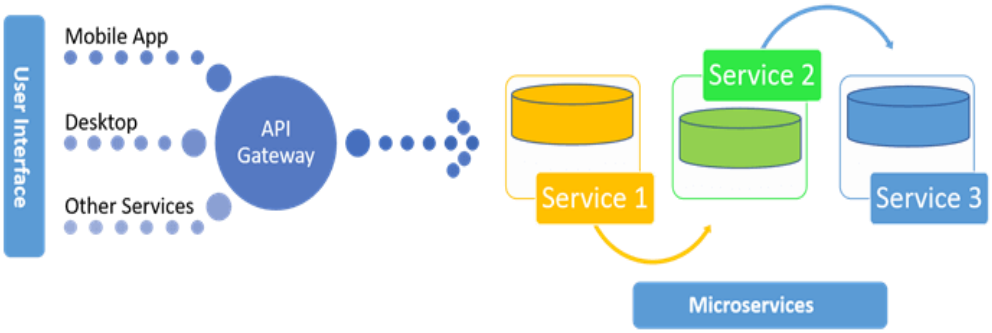



Fig.34.Microservices

# 3.CAPSTONE PROJECT- ONLINE ADS

## 3.1. Problem specification:

This problem is assigned to me during my internship at WIPRO Limited. To design an "Online advertising platform" using Visual Studio, ASP.NET, Microsoft SQL Server Management Studio, with the necessary functionalities as given.

## 3.2. Problem definition:

Create an online advertising platform. A user can post his ads on this platform. The system will be on windows platform using C#, ASP.NET, SQL, Entity Framework Core. We can also deploy the Application on cloud.

There are two types of users for the web application, Admin, and a User. Both are having different functionalities. Data organization is done using SQL Server by creating the necessary normalized tables.

## 3.3. Limitation of existing system:

- Data redundancy: It means that same data fields appear in many different files and often in different formats. In Manual system it poses quite a big problem because the data must be maintained in large Volumes, but in our system, this

problem can be overcome by providing the condition that if the data entered is duplicate, it will not be entered otherwise updating will take place.

- Difficulty in accessing the data: In manual system, searching information is time consuming but in our system, any information Can be accessed by providing the primary key.
- Unsatisfactory security measures: In manual system, no security measures were provided but, in this system, Password security Has been provided. The person can access the system by providing the correct password Otherwise he is denied the access.

### 3.4. Objective of the project and SDLC Model Used:

- There are two types of users. Admin administrates the website. User can post his/her ads online using the advertising platform.
- Admin and user both are having different functionalities.
- Login and registration functionalities help having smooth experience for the user.
- Data is stored in the required normalized tables with the help of SQL Server.
- The overall method is very easy and based on few steps. No huge amount of knowledge is needed to complete the task.

SDLC (Software Development Life Cycle) is essentially a process that consists of a series of planned activities to develop, alter, or manage the software or pieces of software. SDLC is a structure to develop a software product that defines the process of planning, implementation, testing, documentation, deployment, and ongoing maintenance and support. The methodology within the SDLC process can vary across industry and organization, but the steps or phases remain same.

The Waterfall model is probably the simplest software development model. As its name suggests, this model has been inspired by a natural phenomenon – waterfall. Like any waterfall, this SDLC model maintains a linear approach throughout. Developers who follow this web development model believe in completing and documenting the scheduled tasks in each phase as they continue to move downward.

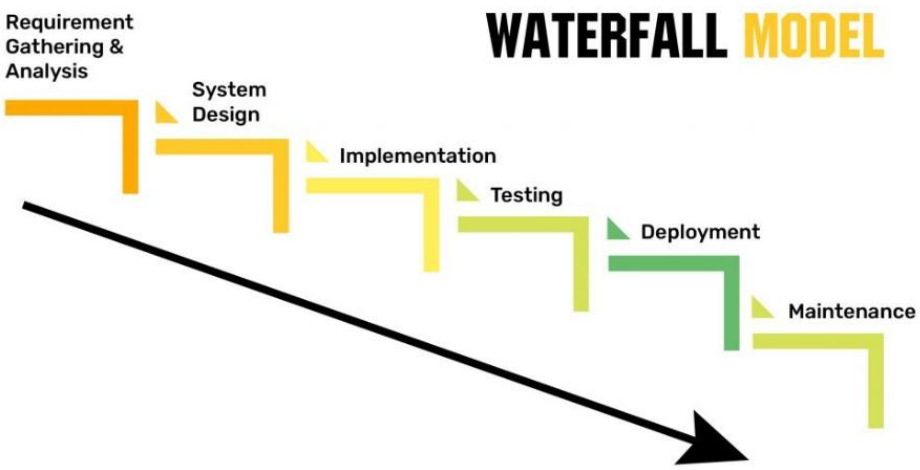



Fig.35.Waterfall model

### 3.5. Feasibility study:

An initial investigation in a proposal that determines whether an alternative system is feasible. A proposal summarizing the thinking of the analyst is presented to the user for review. When approved, the proposal initiates feasibility study that describes and evaluates candidate systems and provides for the selection of best system that meets system performance requirements. To do a feasibility study, we need to consider the economic, technical factors in system development. First a project team is formed. The team develops system flowcharts that identify the characteristics of candidate systems, evaluate the performance of each system, weigh system performance and cost data and select the best candidate system for the job. The study culminates in a final report to the management.

Introduction:

- Describe and identify characteristics of candidate systems.
- Determine and evaluate performance and cost effectiveness of each candidate system.
- Weigh system performance and cost data.
- Select the best candidate system.

Summary:

1. A feasibility study is conducted to select the best system that meets performance requirements. This entails an identification description, an evaluation of candidate systems, and the selection of the best system for the job.

2. A statement of constraints, the identification of specific system objectives and a description of outputs define a system's required performance. The analyst is then ready to evaluate the feasibility of candidate systems to produce these outputs.

3. Three key considerations are involved in feasibility analysis: economic, technical, and behavioural.

4. There are few steps in feasibility study:

a. Statement of constraints: - Constraints are factors that limit the solution of a problem. Some constraints are identified during the initial investigation
b. Identification of specific system objectives: - Once the constraints are spelled out, the analyst proceeds to identify the system's specific performance objectives. They are derived from the general objectives specified in the project directive at the end of the initial investigation. The steps are to state the system's benefits and then translate them into measurable objectives.
c. Description of outputs: - A final step in system performance definition is describing the output required by the user. An actual sketch of the format and contents of the reports as well as a specification of the media used, their frequency, size and numbers of copies required are prepared at this point.

Types of feasible study: -

i. Legal Feasibility: - Determines whether the proposed system conflicts with legal requirements, e.g., a data processing system must comply with the local Data Protection Acts.

ii. Operational Feasibility: -Operational feasibility is a measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives about development schedule, delivery date, corporate culture, and existing business processes.

iii. Economic Feasibility: -The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

iv. Technical Feasibility: -The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

### 3.6. Hardware and Software Requirements:

- 7+ version of Microsoft Windows
- 1 GHZ or faster, 32 bit or 64-bit processor.
- 1 GB RAM for 32 bit or 2 GB RAM for 64 bits.
- .NET Framework 4.5 or higher
- Microsoft Visual Studio
- SQL Server Management Studio 2014
- Browser capabilities- Here, Chrome (17+)

### 3.7. Development Environment:

As explained earlier **Visual Studio** is the main IDE (integrated development environment) we are using. Similarly brief introduction has been given of the following technologies.

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual studio works well with .NET as the required .NET framework can be downloaded.

**.NET** comes in different flavours, more formally known as implementations. .NET Framework is the original implementation of .NET and runs only on Windows. Each implementation includes a runtime and a class library. It may also include application frameworks and development tools. .Net supports three languages – C#, F# and Visual basic.

**ASP.NET MVC:** We must use asp.net MVC template to create the project. As explained earlier MVC stands for models, views, and controllers. MVC makes the web development easy.


Fig.36.MVC

What are the benefits of MVC?
*Separation of concerns* – the MVC framework provides separation of the Business Logic, UI, Data or Model. This is another way of saying that it provides separation of user interface and program logic.
*More control* – with the MVC framework, we have more control over CSS, HTML, and JavaScript relative to the older frameworks.
*Testability* – the MVC framework provides good support towards testing during development of any web application.

*Lightweight* – since the MVC framework does not use view state it significantly reduces the bandwidth.

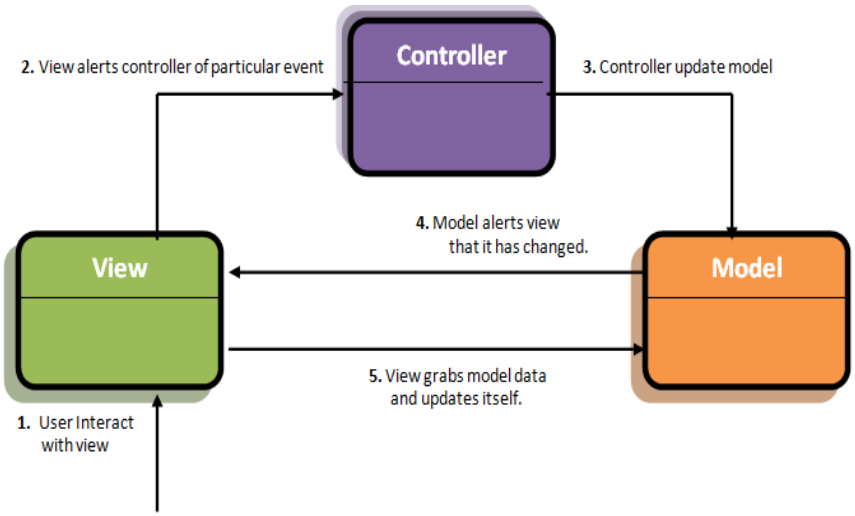



Fig.37 MVC

What are the differences between ASP.NET and ASP.NET MVC?

*Development model* – ASP.NET uses the event-driven model whereas MVC uses the Model View Controller model.

*Html helpers and server controls* – ASP.NET uses server controls whereas MVC uses HTML helpers.

*Route based and file-based URLs* – ASP.NET has file-based URLs. This means that the file names that have been mentioned in the URL must exist physically. The MVC framework follows a router-based URL. This means that the URLs are divided into actions and controllers. It is the controller which is more important than the actual physical file.

*Syntax* – ASP.NET follows the Web Form Syntax whereas MVC follows customizable syntax.

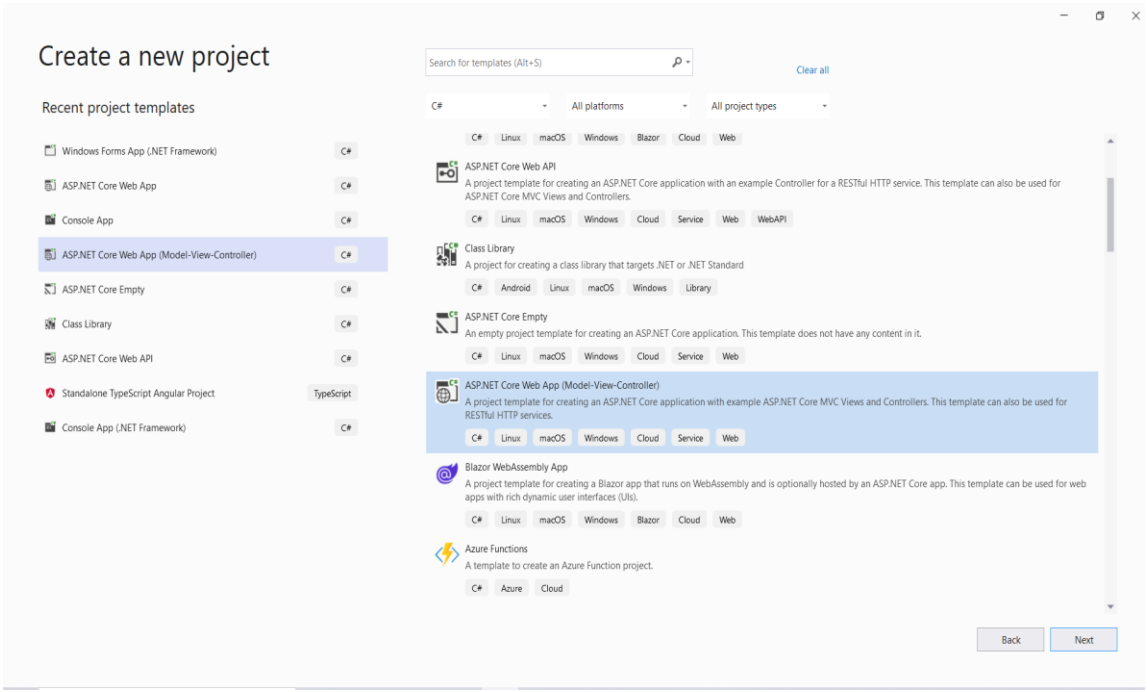
*Logic* – in ASP.NET, the views are tightly coupled. In MVC the logic and views are kept separately.

*Layouts and Master Pages* – for look and feel, ASP.NET uses Master Pages while MVC uses layout.

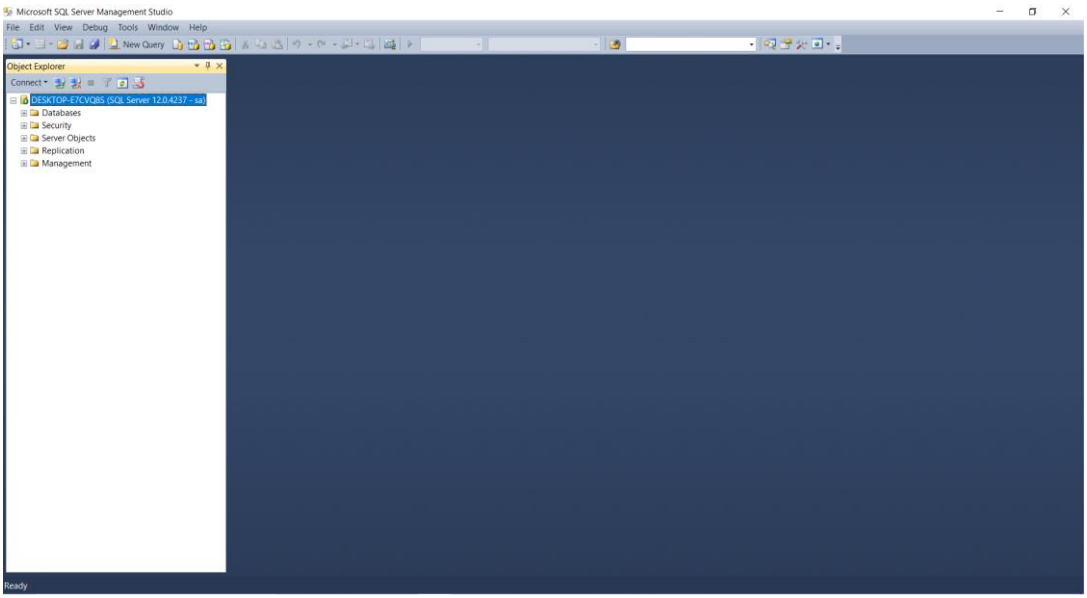
*Code reusability* – ASP.NET has user controls whereas MVC has Partial Views.

*Open source*– ASP.NET is not open source while MVC is free.

To create a ASP.NET MVC web application choose the MVC template for the project-



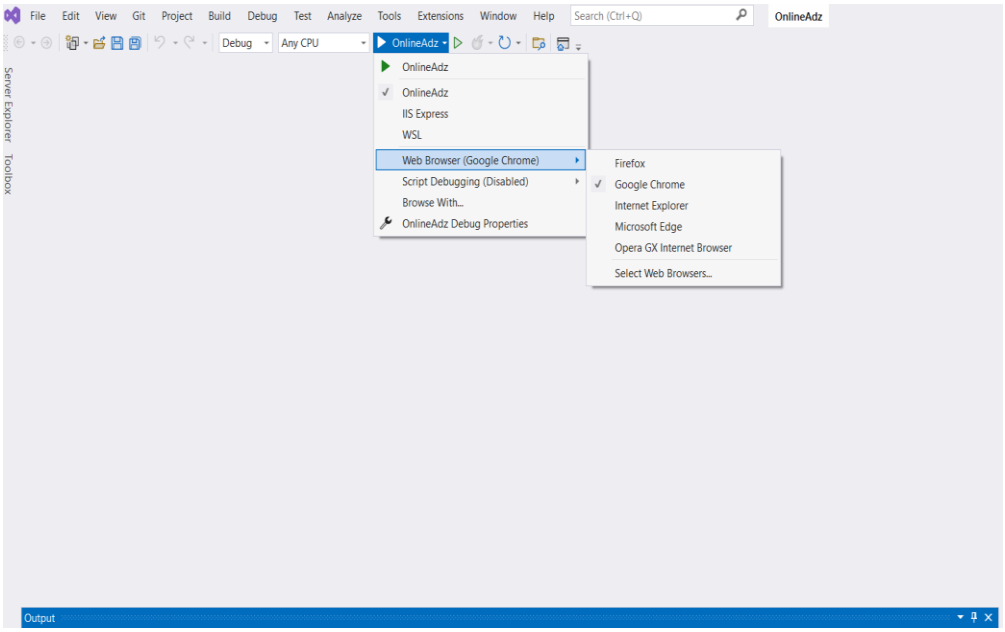

Next, necessary tables are created using **MS SQL SERVER 2014 Management Studio (SSMS)**-



**Web browser-**

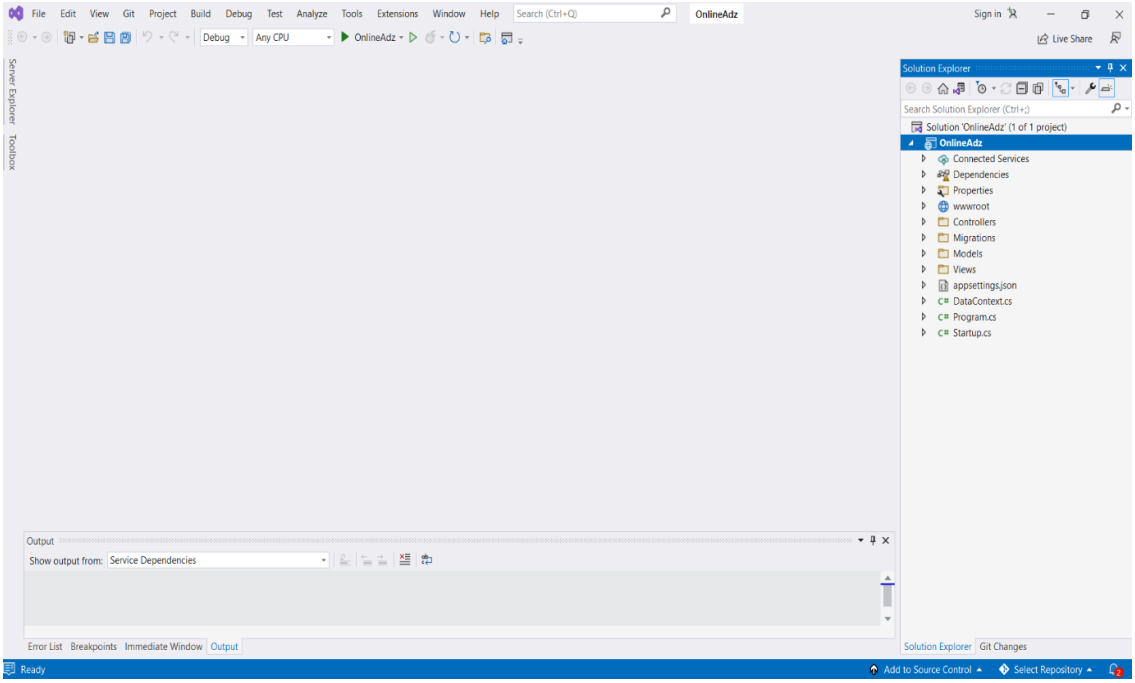
I have used Google Chrome is being used to host the web application.

After we run the application, it automatically takes us to the web browser.

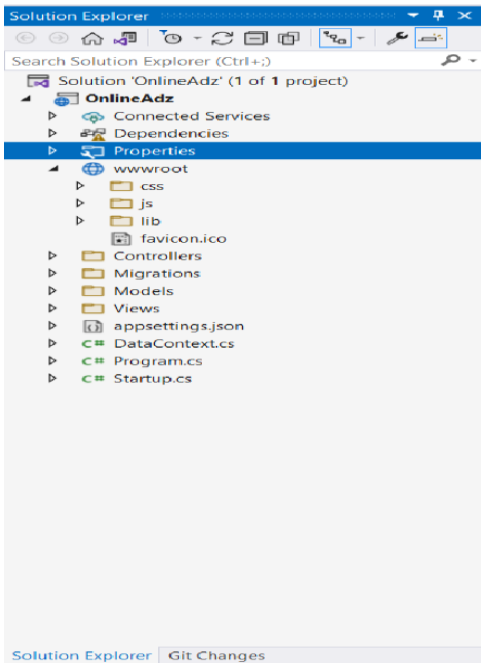



## 3.8. System Design and Implementation:

- Basic structure of the project -



- wwwroot- It consists of CSS, JS, and lib folder. MVC provides the necessary web design essentials such as CSS, JS, and Bootstrap.




- Controllers-This folder consists of-

  *Account Controller*- It consists of backend code for login and registration and account management functionalities.
  *Category Ad*- This controller is used by the "admin" so he can manage categories of ads for the users to choose from. Only admin can create, delete, add, or update categories of ads.
  *Home*- It is for the home page of the website.
  Platform controller- This is for the description of the type of ads a user can post with the preferable online/social platform, ad is going to be posted.
  *User Ad controller*- This is for the "user" to check his/her ad. Only that user can delete, edit, update it.
  *User Interest Controller*- A user can give his ad details to the admin to post his/her ad on the platform. This data is received to the admin via this controller when breakpoint is added in the code.

- Migrations- This folder contains the information about tables formed in the SQL database after we connected that to our visual studio using code first approach of EF Core. The tables formed, saves the data about the person who has registered. The tables our automatically formed in SSMS after we connect VS to SSMS.
- Models – This folder consists of-

  *AdApplicationContext*- this class inherits from dbcontext class. **DbContext** is an important class in Entity Framework API. It is a bridge between your domain or entity classes and the database.

  AdApplicationContext is formed when I connected my database with the visual studio. It is in relation to user Ad and category Ad controller. EF Core Database first approach is used here. AdApplication is the name of database in SSMS. It consists of tables related to category of ads and ad information of users.

  Model classes related to the tables in database are formed automatically (*Category Ad and UserAd*)

  *LoginViewModel*- Class is created for variables used in login.

  *RegisterViewModel*- Class is created for variables used in registration.

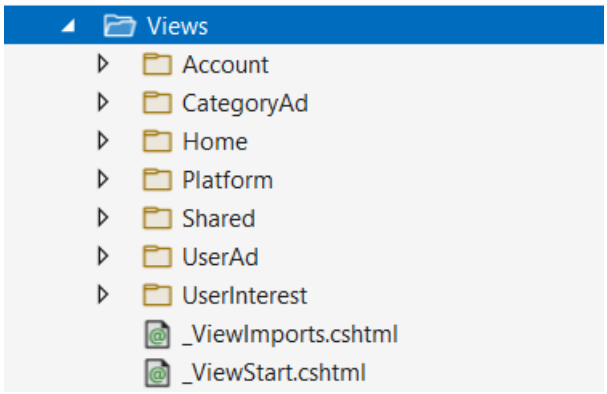
  *popularityContext*- Also inherits from dbContext class as EF Core database first approach is used. popularity is the name of database and platform is the table in it. So, a model class called Platform is formed automatically.

  *UserInterest*- model class about the credentials to be sent to admin by user.

- Views – Views tell us about how the web page is going to look(front-end).

  For every Controller we are having corresponding folder in the views folder of the application. Every folder in the view folder consists of the views of type. cshtml in it. cshtml stands for C# HTML. These views allow for Razor syntax, which is a combination of HTML mixed with C#.

  Every method in the specific controller class is having a corresponding view like the index method in the controller class will have index view.




*Account* - It consists of views-

- Login.cshtml
- Register.cshtml

*CategoryAd*- It consists of views-

- Create.cshtml
- Delete.cshtml
- Details.cshtml
- Edit.cshtml
- Index.chtml

*Home*- It consists of views-

- AboutUs.cshtml
- ContactUs.cshtml
- Index.cshtml
- Privacy.cshtml

*Platform*- It consists of views-

- _viewstart
- Index.cshtml

*Shared*- The Shared Folder in MVC is used to contain all the views which are needed to be shared by different controllers e.g., error files, layout files, etc. I have used three layouts (_Layout, _Layout1, _Layout2.cshtml), one for Home page, one for admin and one for user.

*UserAd*- It consists of views-

- _viewstart
- Index.cshtml
- Edit.cshtml
- Create.cshtml
- Details.cshtml
- Delete.cshtml

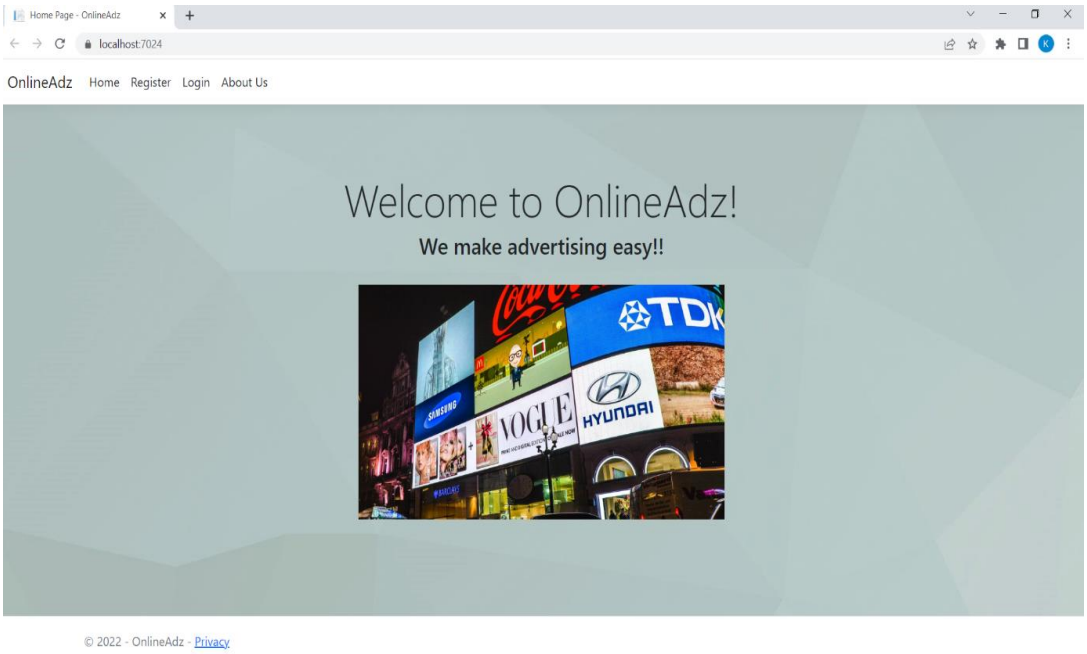
*UserInterest*- It consists of views-

- viewstart
- index.cshtml

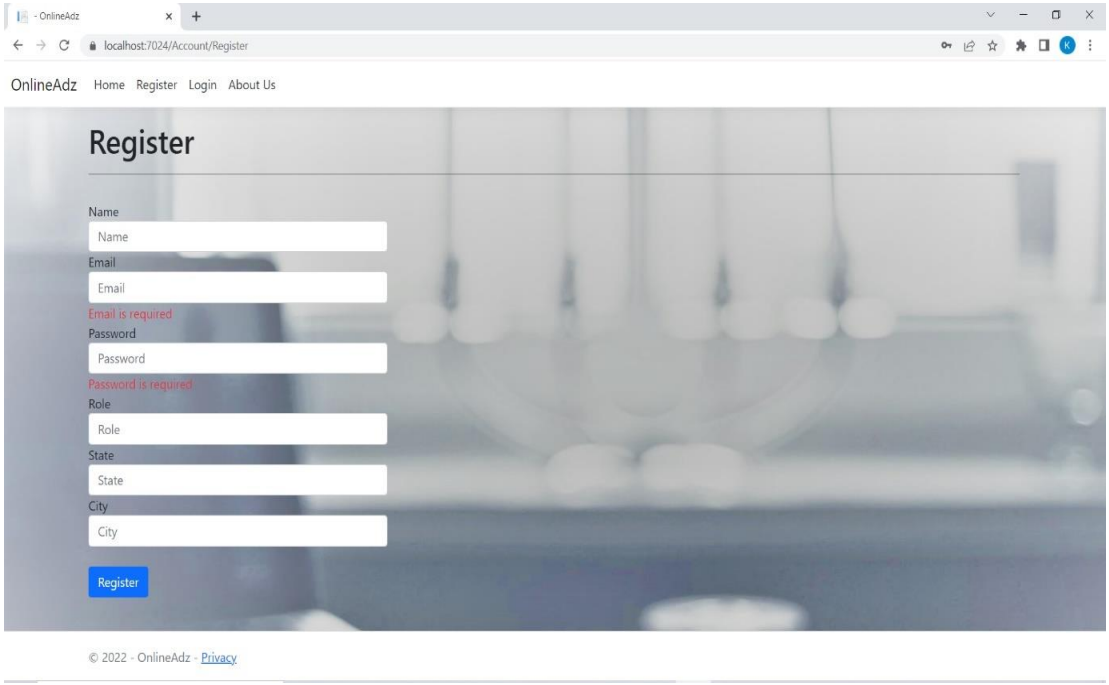
*Viewstart.cshtml*- Specifies the layout of the project.

- appsettings.json- The appsettings.json file is an application configuration file used to store configuration settings such as database connections strings, any application scope global variables, etc.
- **Web Application**-

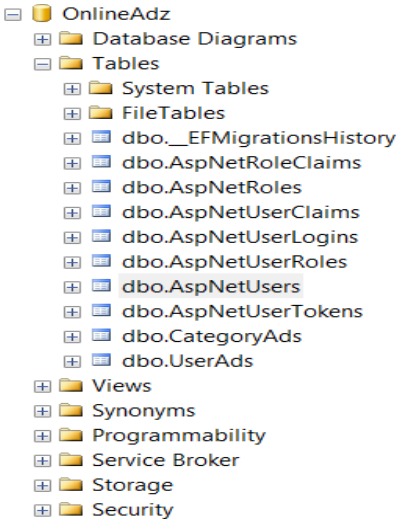
  After we run the application, we are taken to the web browser where we get the Home Page.



Registration Page-




After we connected database to VS tables our automatically created-




After one register, table gets automatically updated-

SQLQuery4.sql - DE...OnlineAdz (sa (55)) | SQLQuery3.sql - DE...OnlineAds (sa (54)) | SQLQuery2.sql - DES...plication (sa (53)) | SQLQuery1.sql - DES...plication (sa (52))

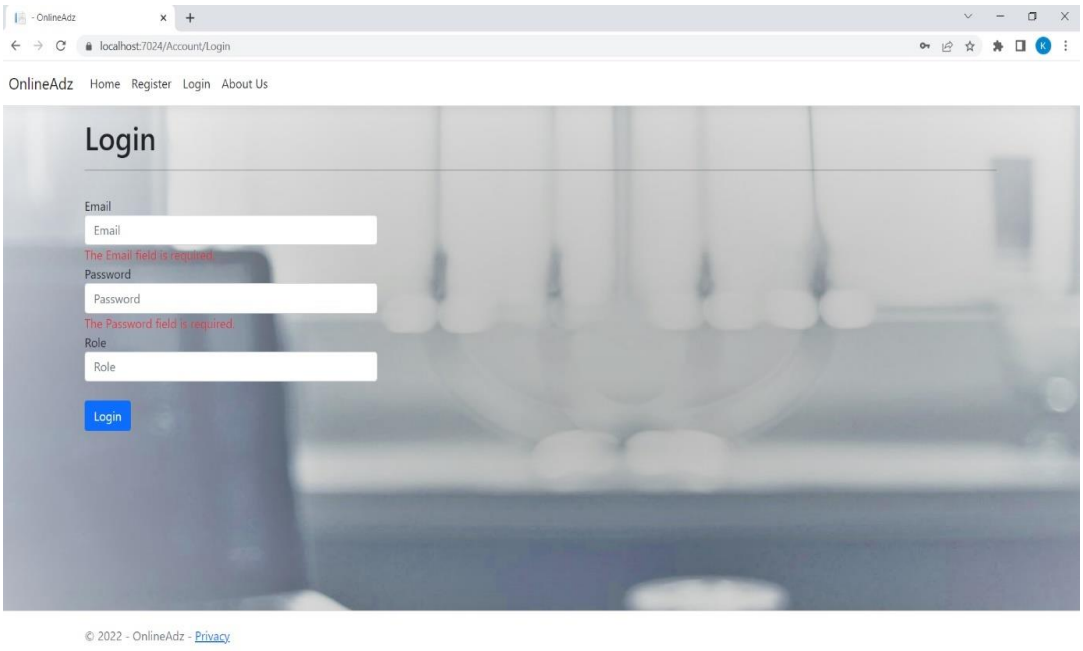
```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [Id]
      ,[UserName]
      ,[NormalizedUserName]
      ,[Email]
      ,[NormalizedEmail]
      ,[EmailConfirmed]
      ,[PasswordHash]
      ,[SecurityStamp]
      ,[ConcurrencyStamp]
      ,[PhoneNumber]
      ,[PhoneNumberConfirmed]
      ,[TwoFactorEnabled]
      ,[LockoutEnd]
      ,[LockoutEnabled]
      ,[AccessFailedCount]
  FROM [OnlineAdz].[dbo].[AspNetUsers]
```

100 %

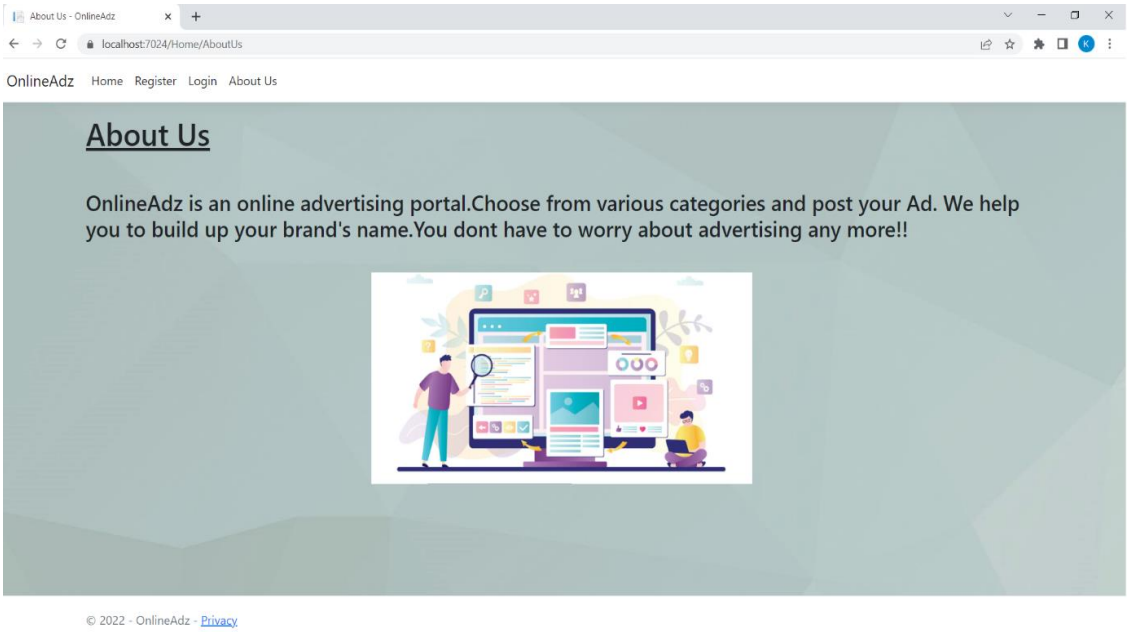
Results | Messages

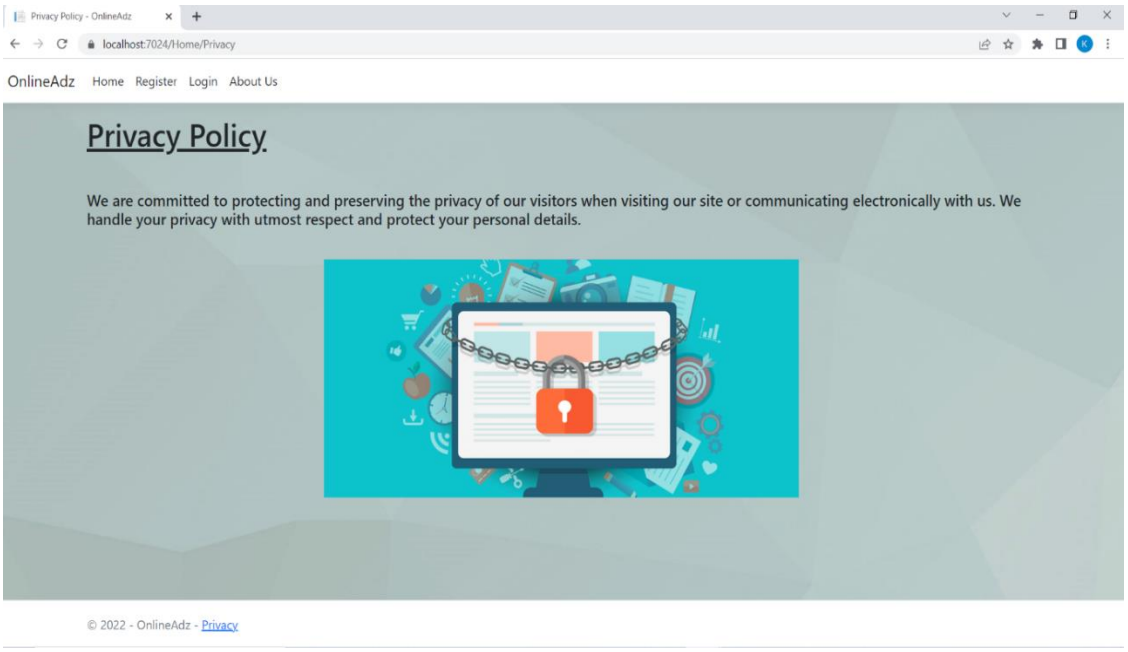
| | Id | UserNa... | NormalizedUserNa... | Email | NormalizedEmail | EmailConfirm... | PasswordHash | S |
|---|---|---|---|---|---|---|---|---|
| 1 | 407587ee-6dd5-435a-a3a0-0bca3fa512c1 | Adam | ADAM | adam@gmail.com | ADAM@GMAIL.COM | 0 | AQAAAAEAACcQAAAAEGoO8mb0nwM2pcGlPA9MmssXSLBqZhW7... | 4 |
| 2 | 44303770-2d41-4954-adde-153a08ce666b | Paul | PAUL | paul@gmail.com | PAUL@GMAIL.COM | 0 | AQAAAAEAACcQAAAAECFBcFvX9TQsG6Q9WmRBXXRjK550f6Cx... | C |
| 3 | 74d6def1-8cfa-4448-838c-13985b7b302d | Celine | CELINE | celine@gmail.com | CELINE@GMAIL.COM | 0 | AQAAAAEAACcQAAAAEO8wDMrq5V42sOkpXMn4adArvuLH2kH5dP... | C |
| 4 | cd78e97e-87ee-4753-b9fb-5e04e11439ae | Brandon | BRANDON | brandon@gmail.com | BRANDON@GMAIL.COM | 0 | AQAAAAEAACcQAAAAEO2Pru+SV4LNAVmGrtyGKom4u6dVgzjP9m... | 5 |
| 5 | ed2a83bf-e66c-4567-a8f0-483eefeec6e7 | Derek | DEREK | derek@gmail.com | DEREK@GMAIL.COM | 0 | AQAAAAEAACcQAAAAEDUO2i5ncdAJ5ToKB1tkjycvHeOQ9SOErSQ... | F |

If one is already registered one can go to the login page-

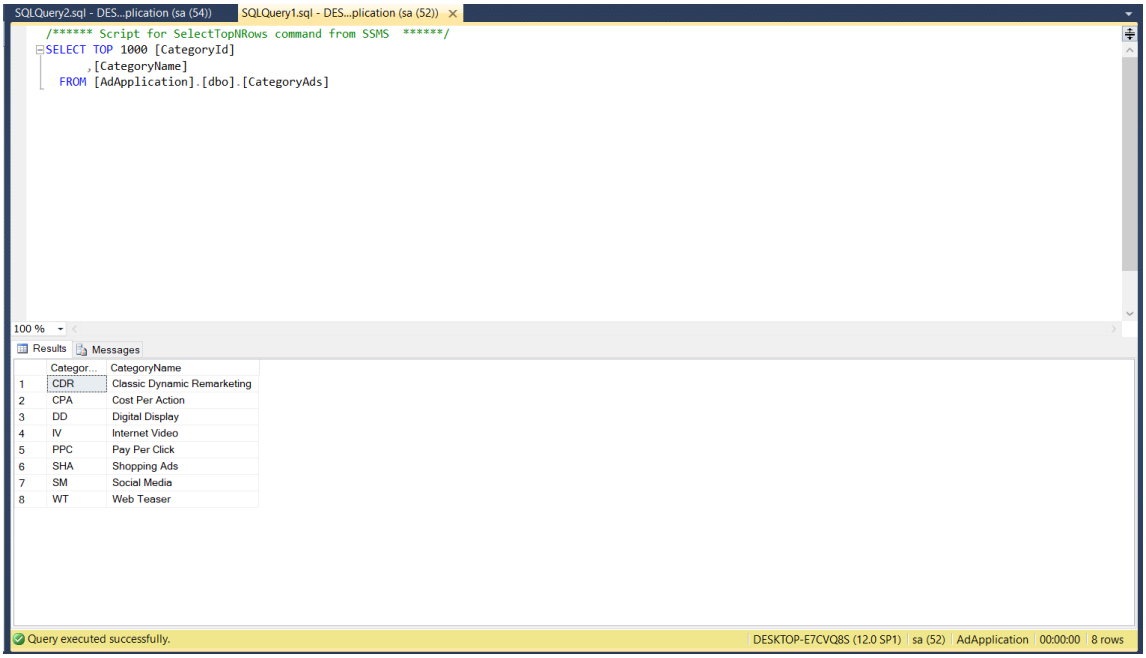



At the Home page one can click on the About Us option and similarly at the bottom of the page one can click on Privacy option -

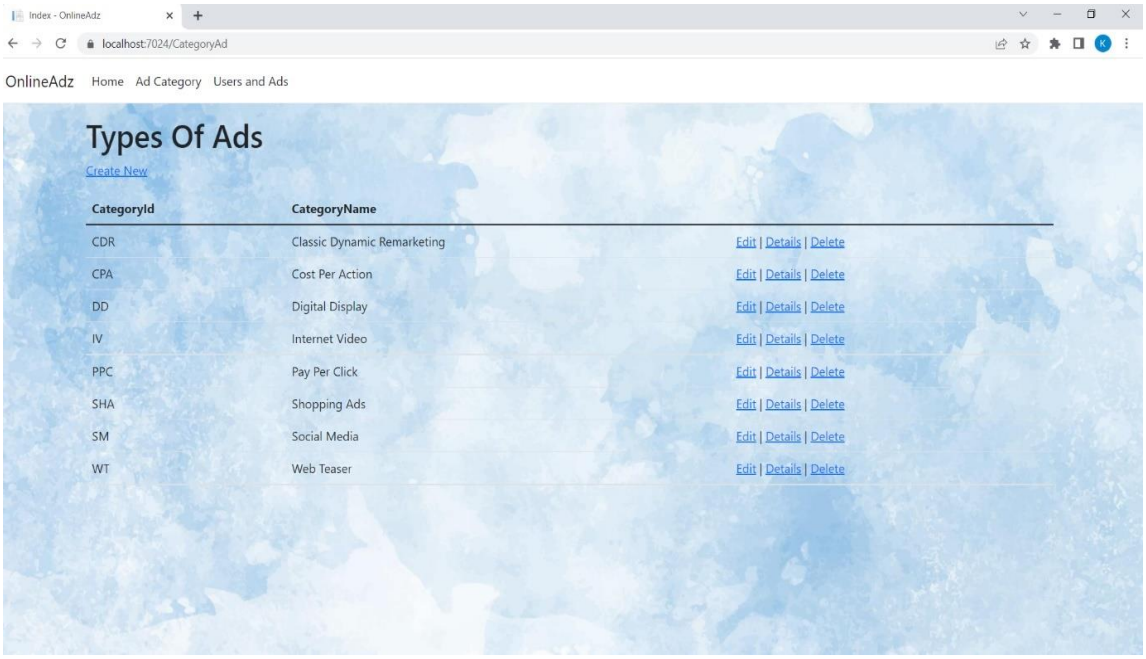



I have created table about ad categories-




“Admin” can change or add Category of Ads and also can look the Layout for the “User”-




I have created the table of registered users -

SQLQuery2.sql - DES...plication (sa (53)) | SQLQuery1.sql - DES...plication (sa (52))

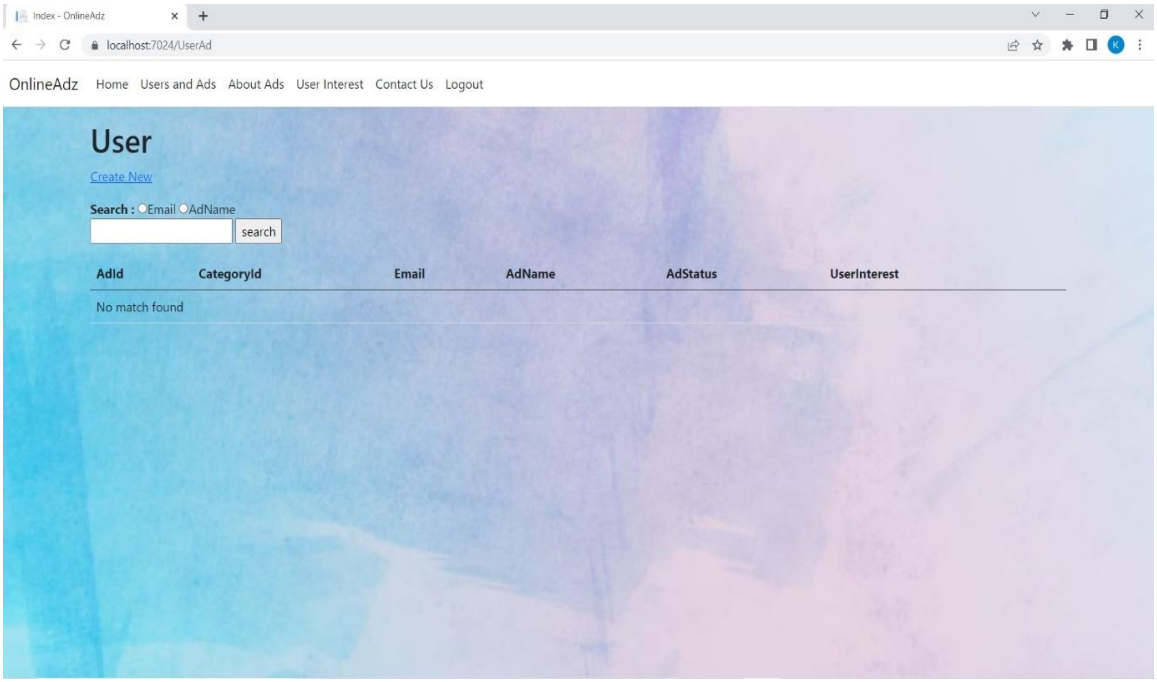
```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [AdId]
      ,[CategoryId]
      ,[Email]
      ,[AdName]
      ,[AdStatus]
      ,[UserInterest]
  FROM [AdApplication].[dbo].[UserAds]
```

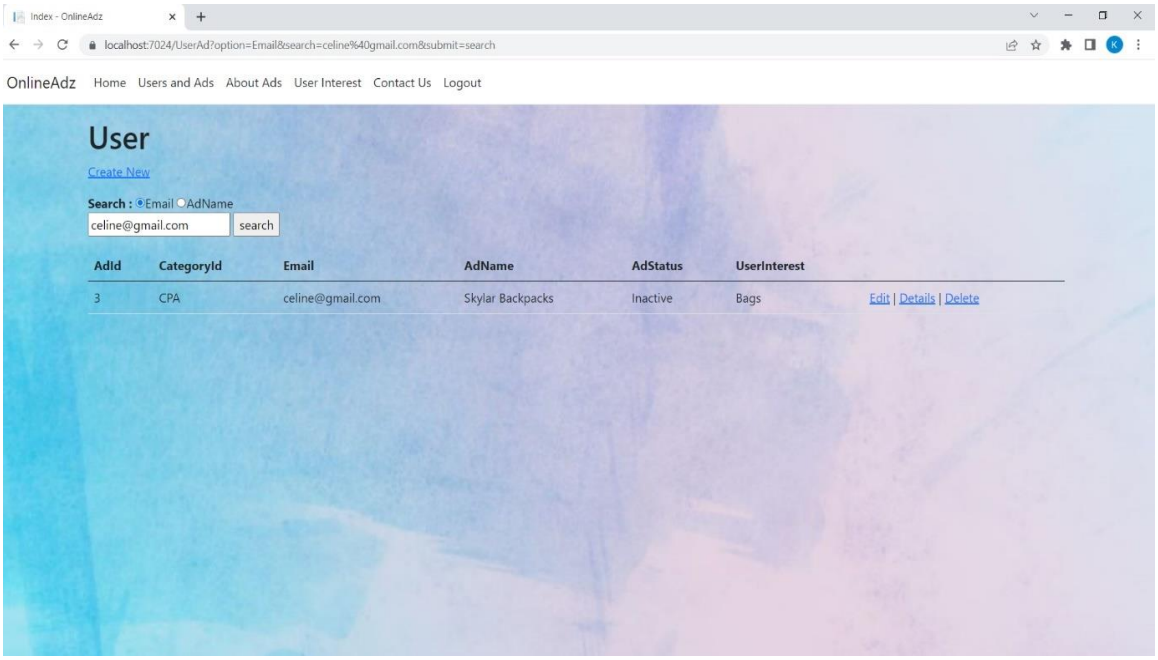
100 %

Results | Messages

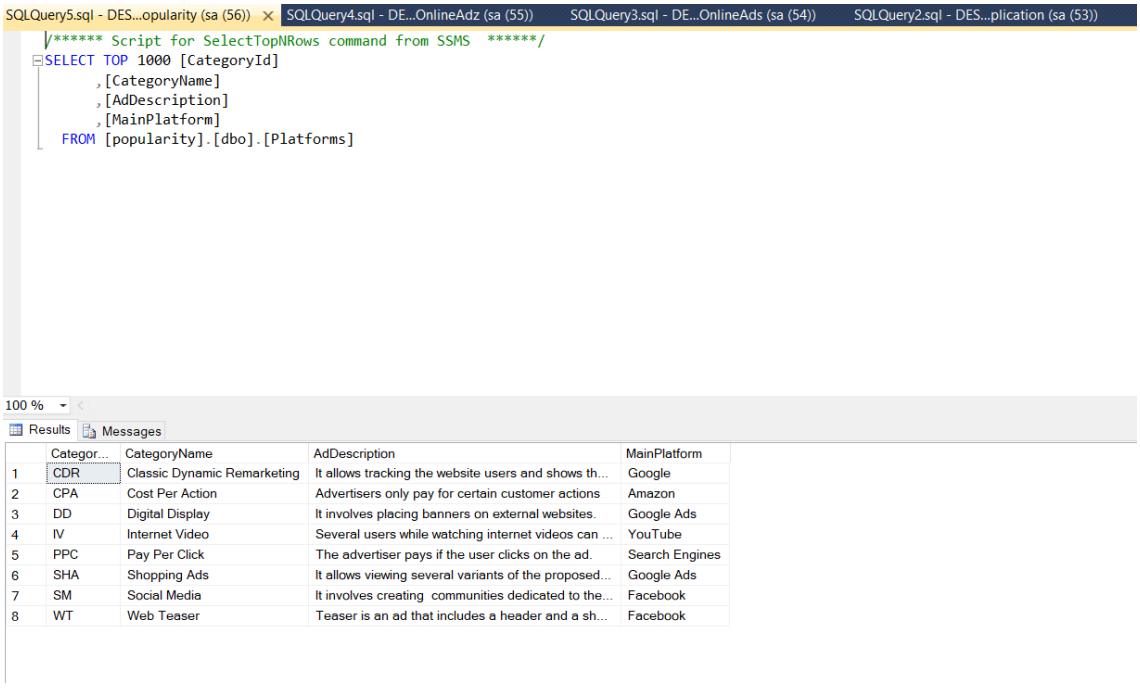
| | AdId | Categor... | Email | AdName | AdStat... | UserInterest |
|---|---|---|---|---|---|---|
| 1 | 1 | PPC | adam@gmail.com | AutoScout:Buy And Sell cars | Active | Cars |
| 2 | 2 | SHA | paul@gmail.com | Titan Watches | Active | Accessories |
| 3 | 3 | CPA | celine@gmail.com | Skylar Backpacks | Inactive | Bags |
| 4 | 5 | IV | derek@gmail.com | Grammarly | Active | English Learning |

User page is having different layout as compared to Admin. User can search for his email id or ad name to get his information-

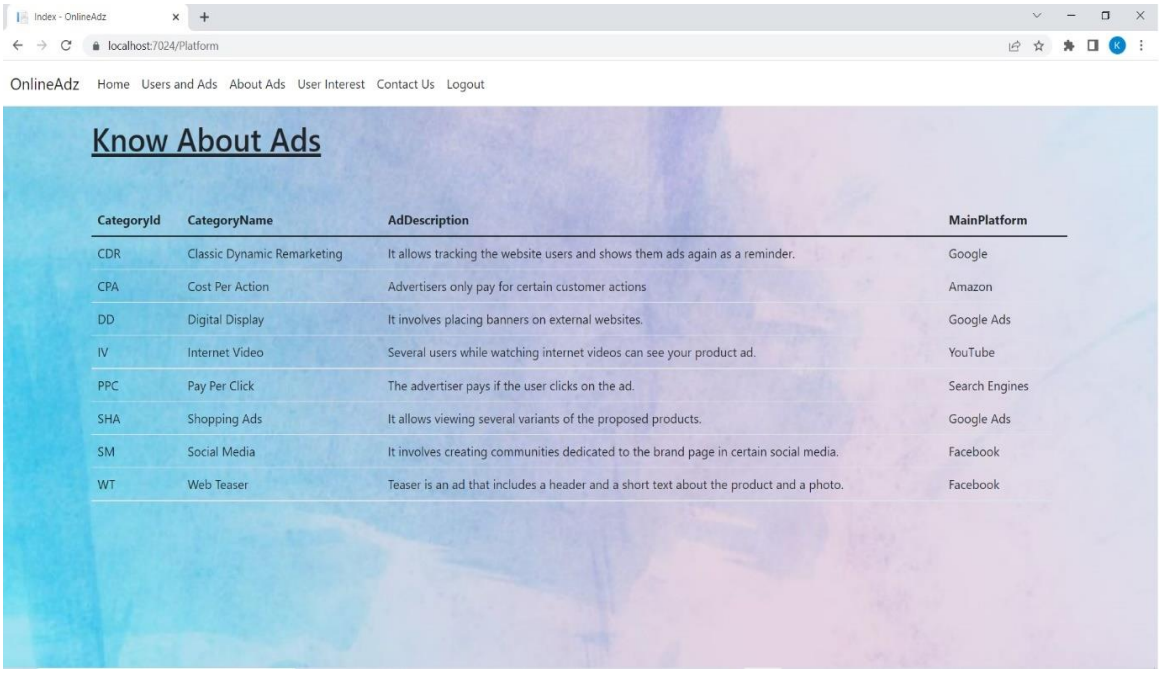




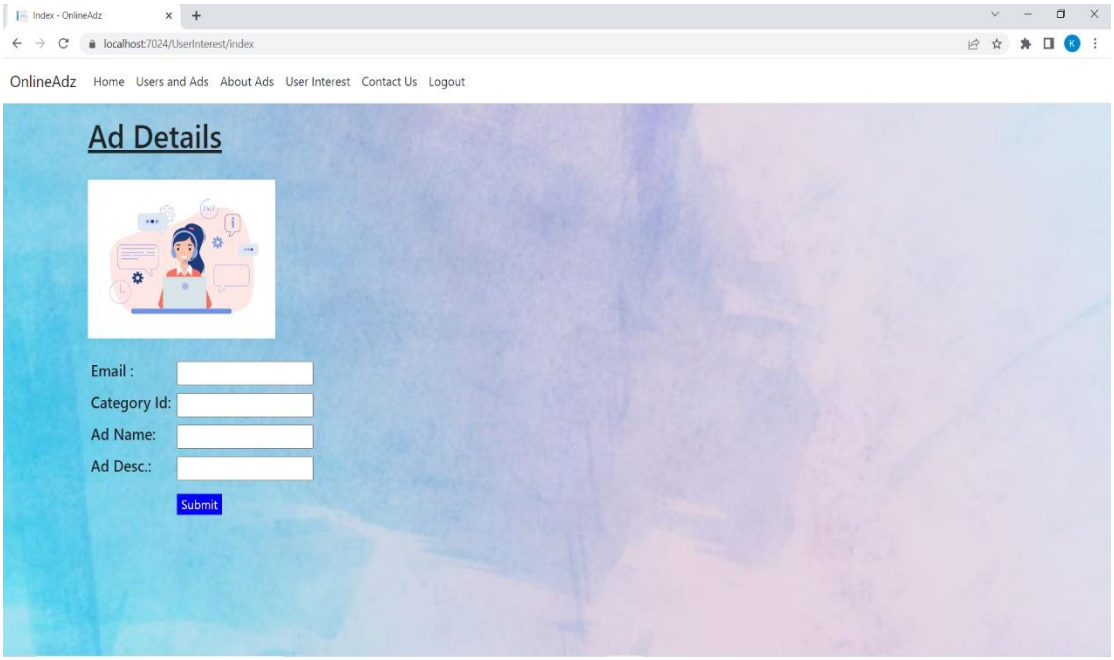

For Users there is chart made for them to choose what Ad is suitable and on which online platform will it be posted by the Admin. I have created the table related to it.

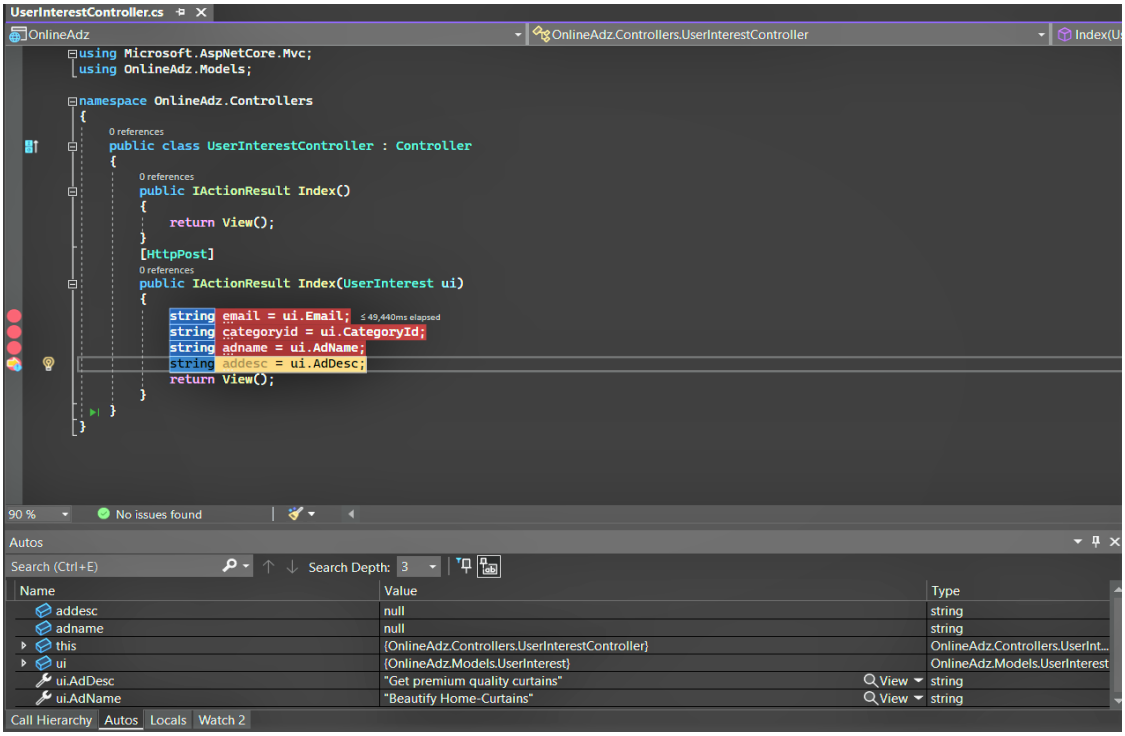



| | Categor... | CategoryName | AdDescription | MainPlatform |
|---|---|---|---|---|
| 1 | CDR | Classic Dynamic Remarketing | It allows tracking the website users and shows th... | Google |
| 2 | CPA | Cost Per Action | Advertisers only pay for certain customer actions | Amazon |
| 3 | DD | Digital Display | It involves placing banners on external websites. | Google Ads |
| 4 | IV | Internet Video | Several users while watching internet videos can ... | YouTube |
| 5 | PPC | Pay Per Click | The advertiser pays if the user clicks on the ad. | Search Engines |
| 6 | SHA | Shopping Ads | It allows viewing several variants of the proposed... | Google Ads |
| 7 | SM | Social Media | It involves creating communities dedicated to the... | Facebook |
| 8 | WT | Web Teaser | Teaser is an ad that includes a header and a sh... | Facebook |

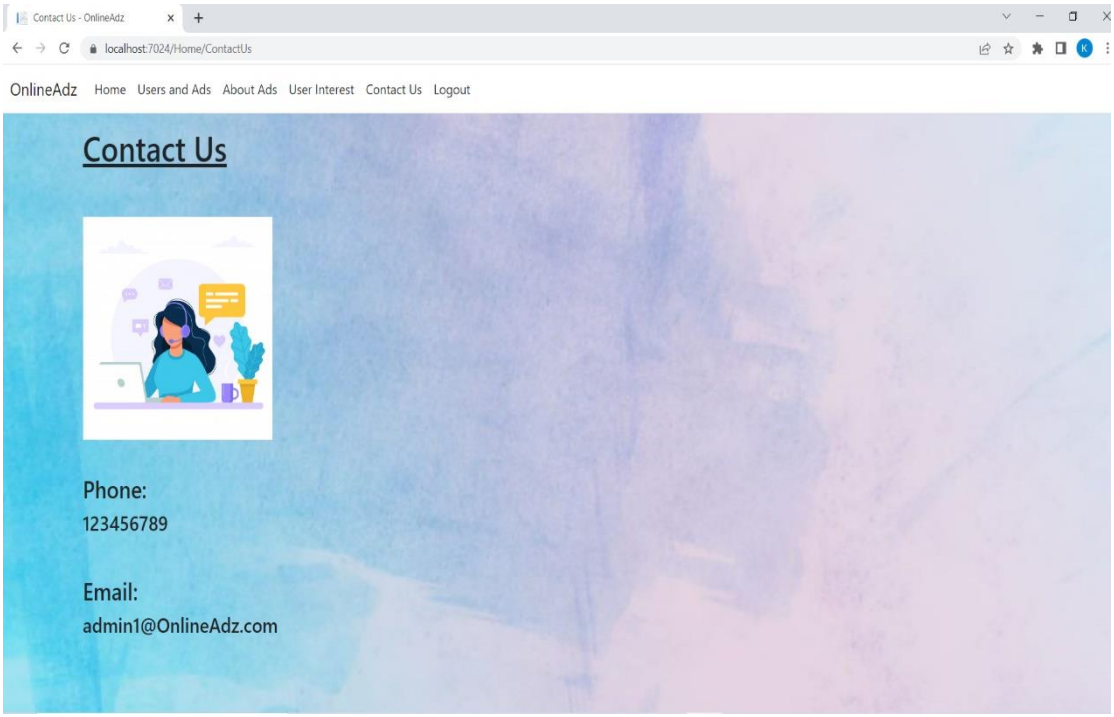



If a user wants Admin to post details of his Ad, he can fill the form of Ad Details and the Admin will be able to see it in the UserInterest Controller page after adding the breakpoint, in the “autos” terminal-






A user can contact admin by following the information mentioned on the contact us page-



# 4.CONCLUSION

Working as an intern at Wipro limited was a great experience. The IT industry is one of the leading industries these days. The internship program helped me learn about the growth of this industry and gain further knowledge about it. Working here also helped broaden my technical knowledge in the programming field as before the internship I had little knowledge about Dotnet.

I learned various aspects of Dotnet, C# and SQL and gained some practical knowledge about these technologies as various assignments and mini projects were given on daily basis. The project "**Online Adz**" will help the users to market their product online. The aim is to post ads and make advertising easy. As social media platforms are very popular these days so advertising online and, on this platform, will be very convenient.

Overall, the internship program helped me develop skills such as handling real-world programs, working under the rules and regulations of an organization, working under a deadline, and working under pressure. It helped me gain technical knowledge as well as professional experience which will be very helpful through my professional career.

# 5.REFERENCES


- docs.microsoft.com
- www.c-sharpcorner.com
- www.javatpoint.com
- www.tutorialspoint.com
- www.tutorialsteacher.com
- www.wikipedia.com