**DISTRIBUTED DATABASE**

**AIM: Implementation of Distributed Database for Bookstore**

**INTRODUCTION:**

A distributed database is basically a database that is not limited to one system, it is spread over different sites i.e., on multiple computers or over a network of computers. A distributed database system is located on various sited that don’t share physical components. This may be required when a particular database needs to be accessed by various users globally. It needs to be managed such that for the users it looks like one single database.

Types:

Homogeneous Database:

In a homogeneous database, all different sites store database identically. The operating system, database management system and the data structures used – all are same at all sites. Hence, they’re easy to manage.

Heterogeneous Database:

In a heterogeneous distributed database, different sites can use different schema and software that can lead to problems in query processing and transactions. Also, a particular site might be completely unaware of the other sites. Different computers may use a different operating system, different database application. They may even use different data models for the database. Hence, translations are required for different sites to communicate.

Here we have used three databases in a distributed fashion. The three databases contain again three tables each. The three databases are as follows:

1. Site 1
2. Site 2
3. Site 3

The three tables in each of the databases are as follows:

1. Books
2. Book Store
3. Stock

**SOFTWARE AND TOOLS USED:**

1. **Visual Studio 2017 with .NET framework 4.6.1**  
   - used to create Windows forms
2. **Oracle Database Desktop class 12c**  
   - used to create an Oracle database
3. **ODP.NET Managed Driver**  
   - used to create a database connection to the windows forms

**SYSTEM REQUIREMENTS:**

1. **Visual Studio 2017 with .NET framework 4.6.1**

**Supported Operating Systems:** Visual Studio 2017 will install and run on the following operating systems:

* Windows 10 version 1507 or higher: Home, Professional, Education, and Enterprise (LTSC and S are not supported)
* Windows Server 2016: Standard and Datacentre
* Windows 8.1 (with Update 2919355): Core, Professional, and Enterprise
* Windows Server 2012 R2 (with Update 2919355): Essentials, Standard, Datacentre
* Windows 7 SP1 (with latest Windows Updates): Home Premium, Professional, Enterprise, Ultimate

**Hardware:**

* 1.8 GHz or faster processor. Dual-core or better recommended
* 2 GB of RAM; 4 GB of RAM recommended (2.5 GB minimum if running on a virtual machine)
* Hard disk space: up to 130 GB of available space, depending on features installed; typical installations require 20-50 GB of free space.
* Hard disk speed: to improve performance, install Windows and Visual Studio on a solid-state drive (SSD).
* Video card that supports a minimum display resolution of 720p (1280 by 720); Visual Studio will work best at a resolution of WXGA (1366 by 768) or higher.

**Additional Requirements:**

* .NET Framework 4.5.2 or above is required to **install** Visual Studio. Visual Studio requires .NET Framework 4.7.2 to run, but this will be installed during setup.

1. **Oracle Database Desktop class 12c**

**Operating system general requirements:** Oracle Database for Windows x64 is supported on the following operating system versions:

* Windows 7 x64 - Professional, Enterprise, and Ultimate editions
* Windows 8 x64 and Windows 8.1 x64 - Pro and Enterprise editions
* Windows 8.1 x64 - Pro and Enterprise editions
* Windows 10 x64 - Pro, Enterprise, and Education editions
* Windows Server 2012 x64 - Standard, Datacentre, Essentials, and Foundation editions
* Windows Server 2012 R2 x64 - Standard, Datacentre, Essentials, and Foundation editions
* Windows Server 2016 x64 - Standard, Datacentre, and Essentials editions

**System Architecture: Processor:** AMD64 and Intel EM64T

**Physical memory (RAM):** 2 GB minimum

**Virtual memory (swap):**

* If physical memory is between 2 GB and 16 GB, then set virtual memory to 1 time the size of the RAM
* If physical memory is more than 16 GB, then set virtual memory to 16 GB

**Disk space:**

* Typical Install Type total: **10 GB**
* Advanced Install Types total: **10 GB**

**Video adapter:** 256 colours

**Screen Resolution:** 1024 X 768 minimum

**PROCEDURE:**

Backend:

1. Open SQL Plus for Oracle and login with credentials
2. Create database 1:

**create database site1;**

**use site1;**

**create table books(**

**ISBN int primary key,**

**author varchar(10),**

**topic varchar(100),**

**total\_stock int,**

**price int**

**);**

**create table book\_store(**

**store\_no int primary key,**

**city varchar(50),**

**state varchar(50),**

**zip int,**

**inventory int**

**);**

**create table stock(**

**store\_no int,**

**ISBN int,**

**quantity int,**

**constraint fk\_1 foreign key (store\_no) references book\_store(store\_no),**

**constraint fk\_2 foreign key (ISBN) references books(ISBN)**

**);**

1. Create database 2: **create database site2;**

**use site2;**

**create table books(**

**ISBN int primary key,**

**author varchar(10),**

**topic varchar(100),**

**total\_stock int,**

**price int**

**);**

**create table book\_store(**

**store\_no int primary key,**

**city varchar(50),**

**state varchar(50),**

**zip int,**

**inventory int**

**);**

**create table stock(**

**store\_no int,**

**ISBN int,**

**quantity int,**

**constraint fk\_1 foreign key (store\_no) references book\_store(store\_no),**

**constraint fk\_2 foreign key (ISBN) references books(ISBN)**

**);**

1. Create database 3:

**create database site3;**

**use site3;**

**create table books(**

**ISBN int primary key,**

**author varchar(10),**

**topic varchar(100),**

**total\_stock int,**

**price int**

**);**

**create table book\_store(**

**store\_no int primary key,**

**city varchar(50),**

**state varchar(50),**

**zip int,**

**inventory int**

**);**

**create table stock(**

**store\_no int,**

**ISBN int,**

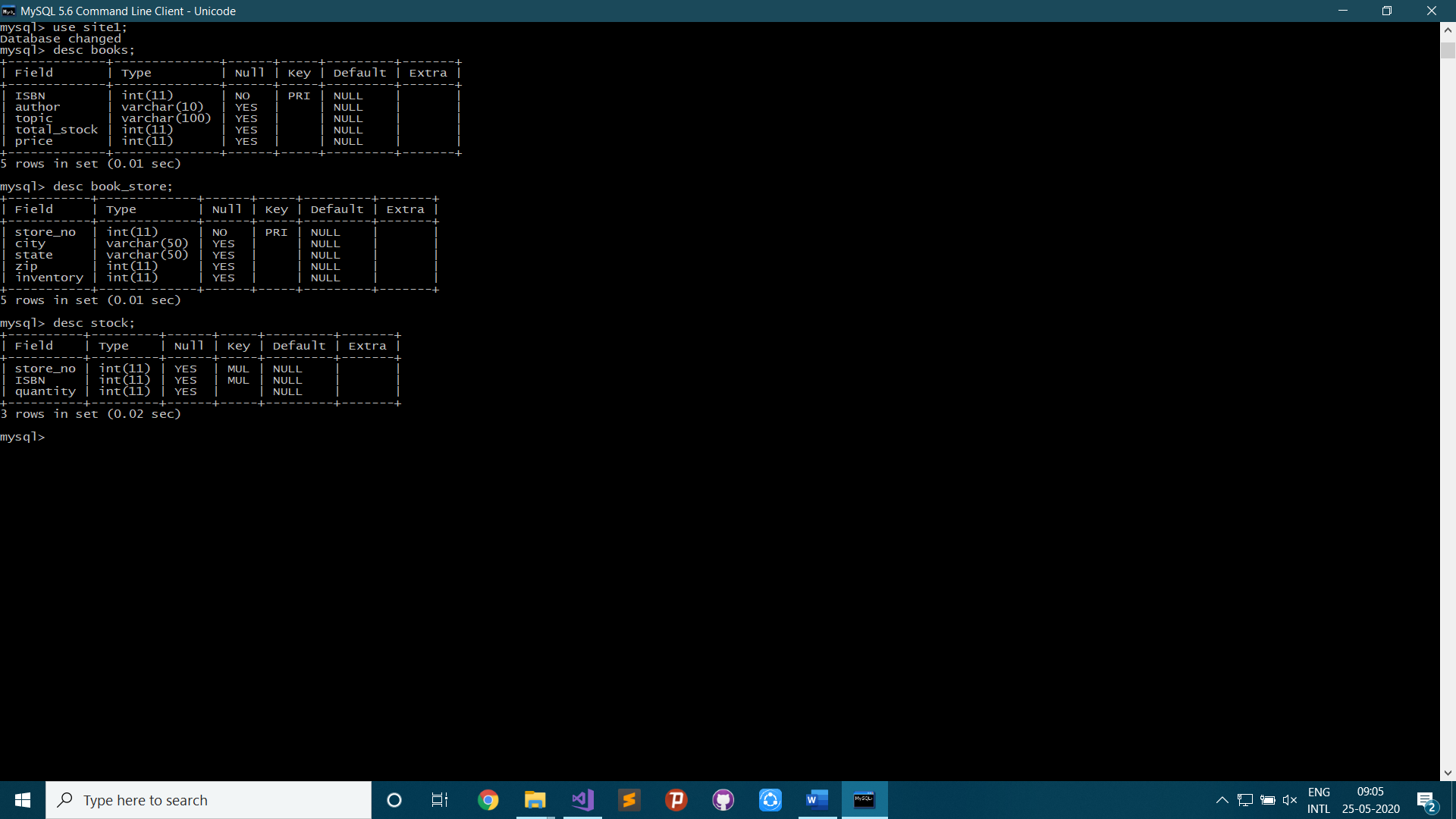
**quantity int,**

**constraint fk\_1 foreign key (store\_no) references book\_store(store\_no),**

**constraint fk\_2 foreign key (ISBN) references books(ISBN)**

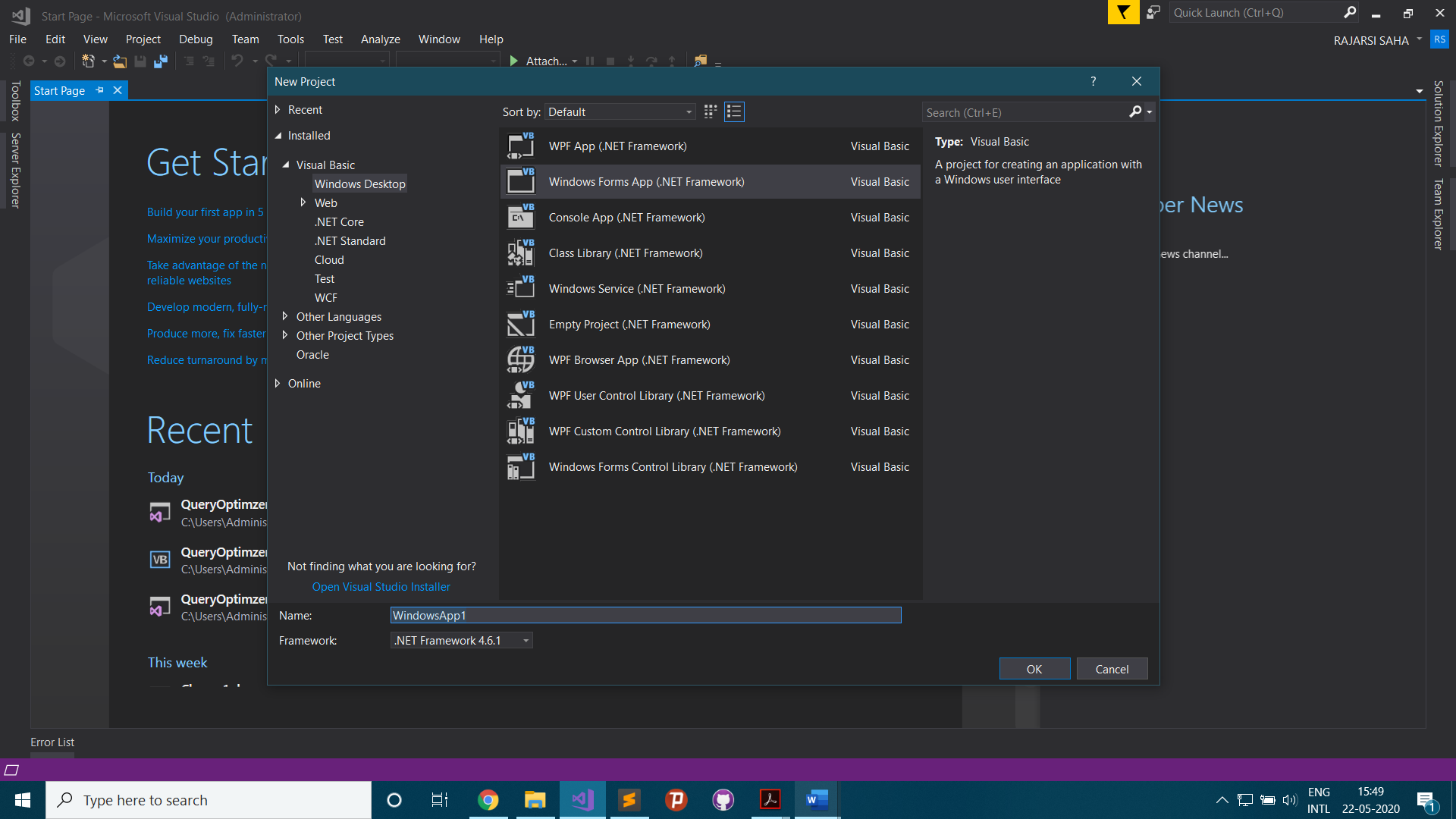
**);**

1. Schema: A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

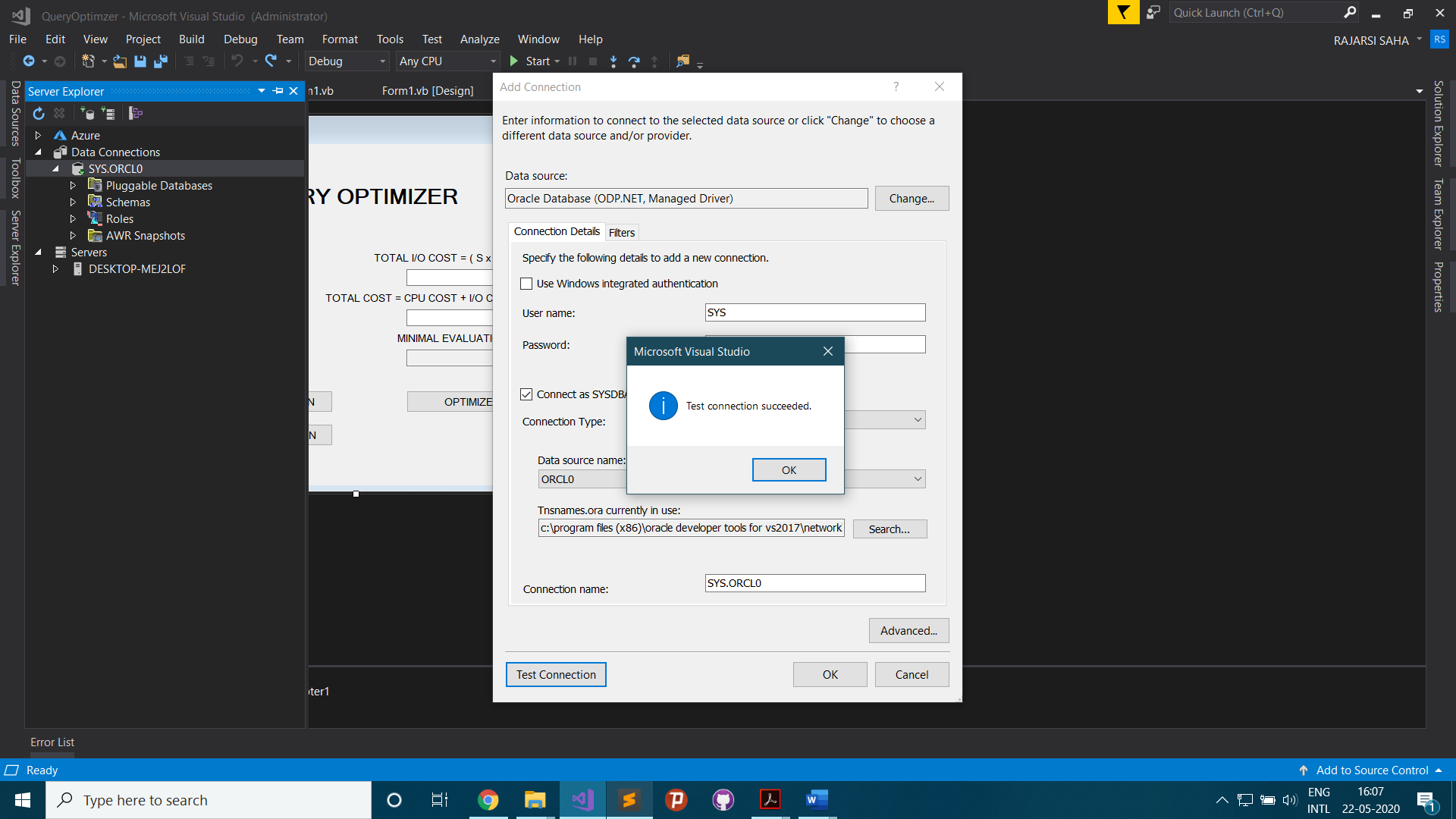


Frontend:

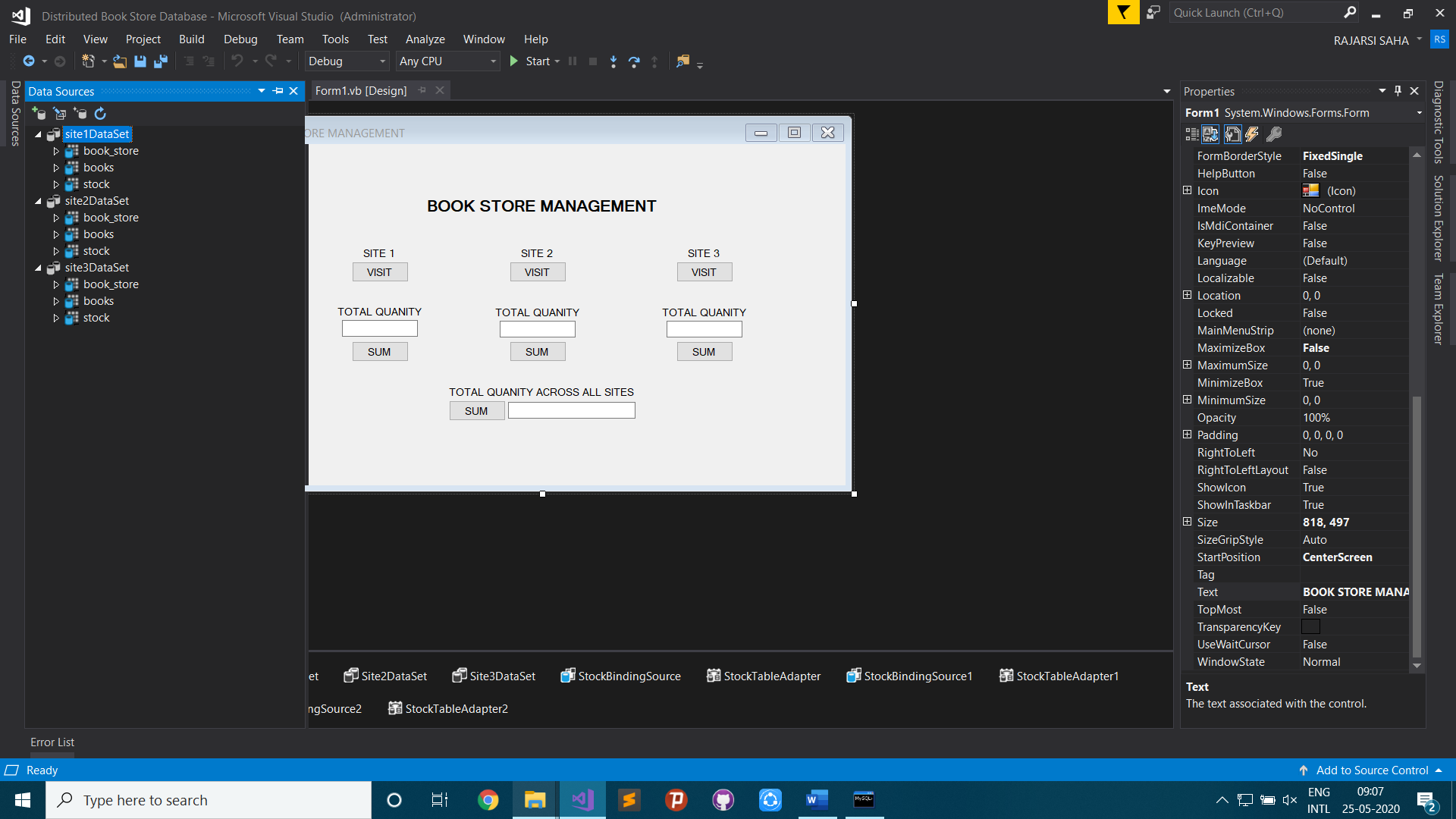
1. Create new Windows Forms App (.NET Framework) in VB Project



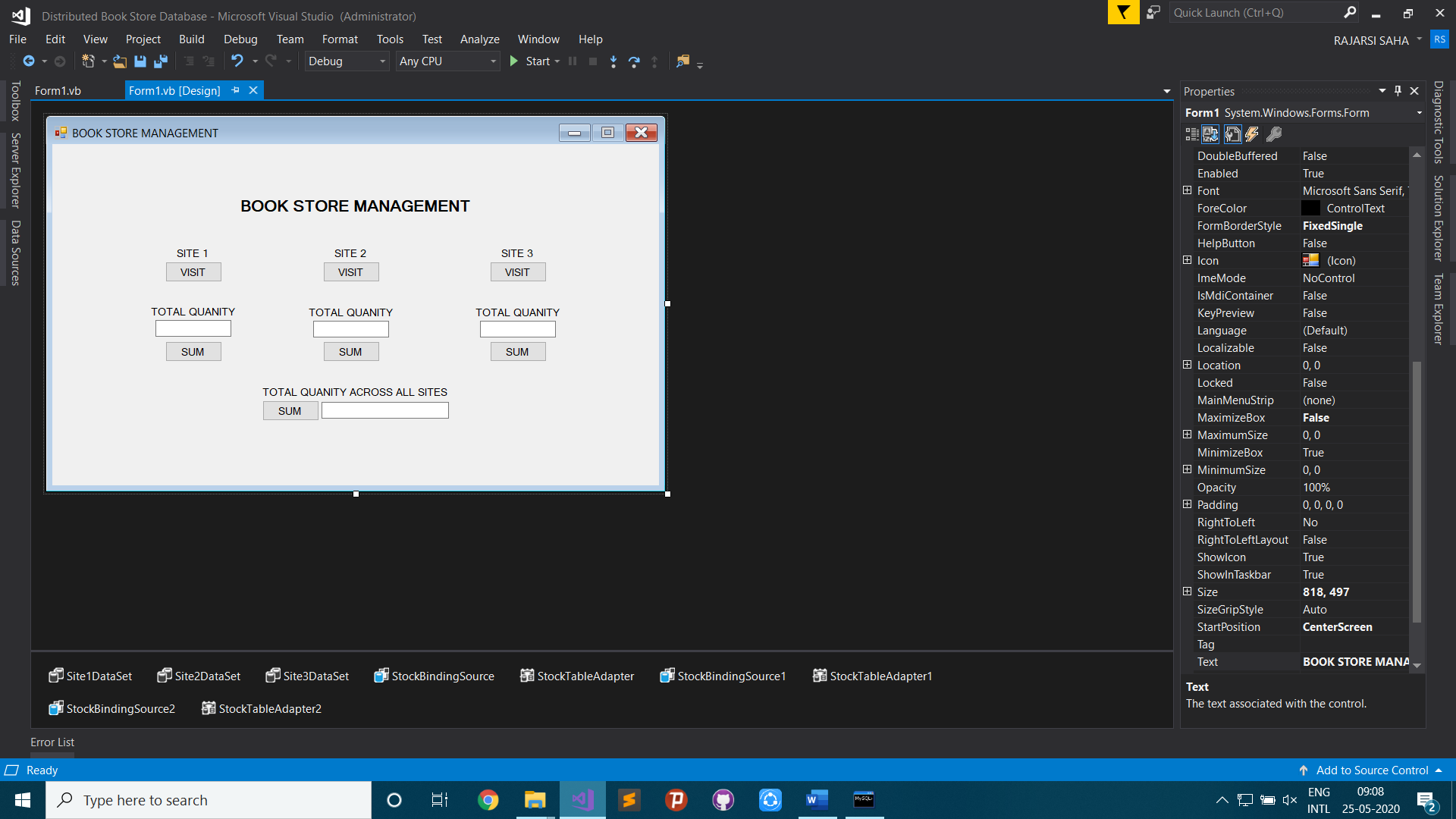
1. Create Database Connection



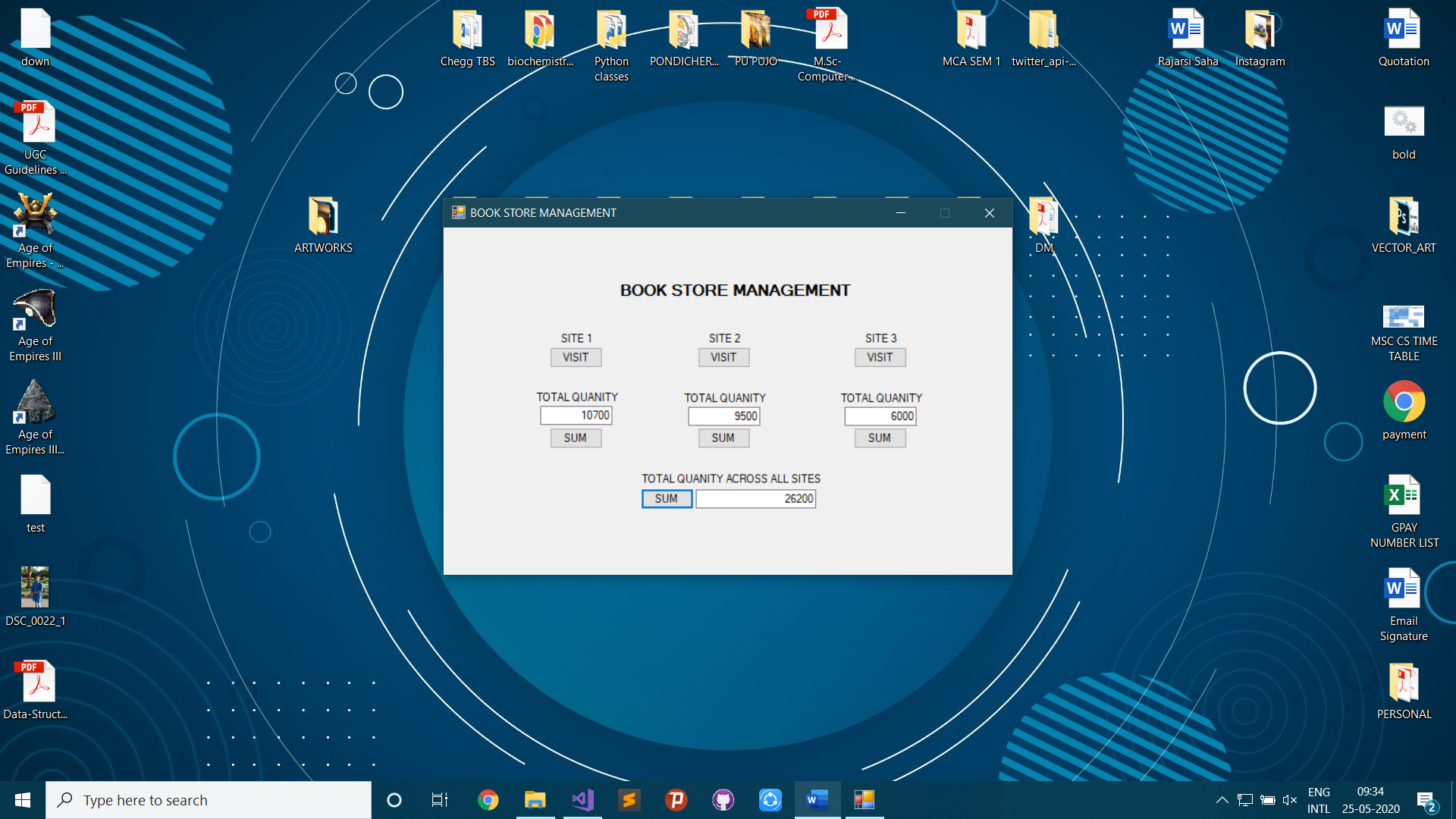
1. Add Dataset



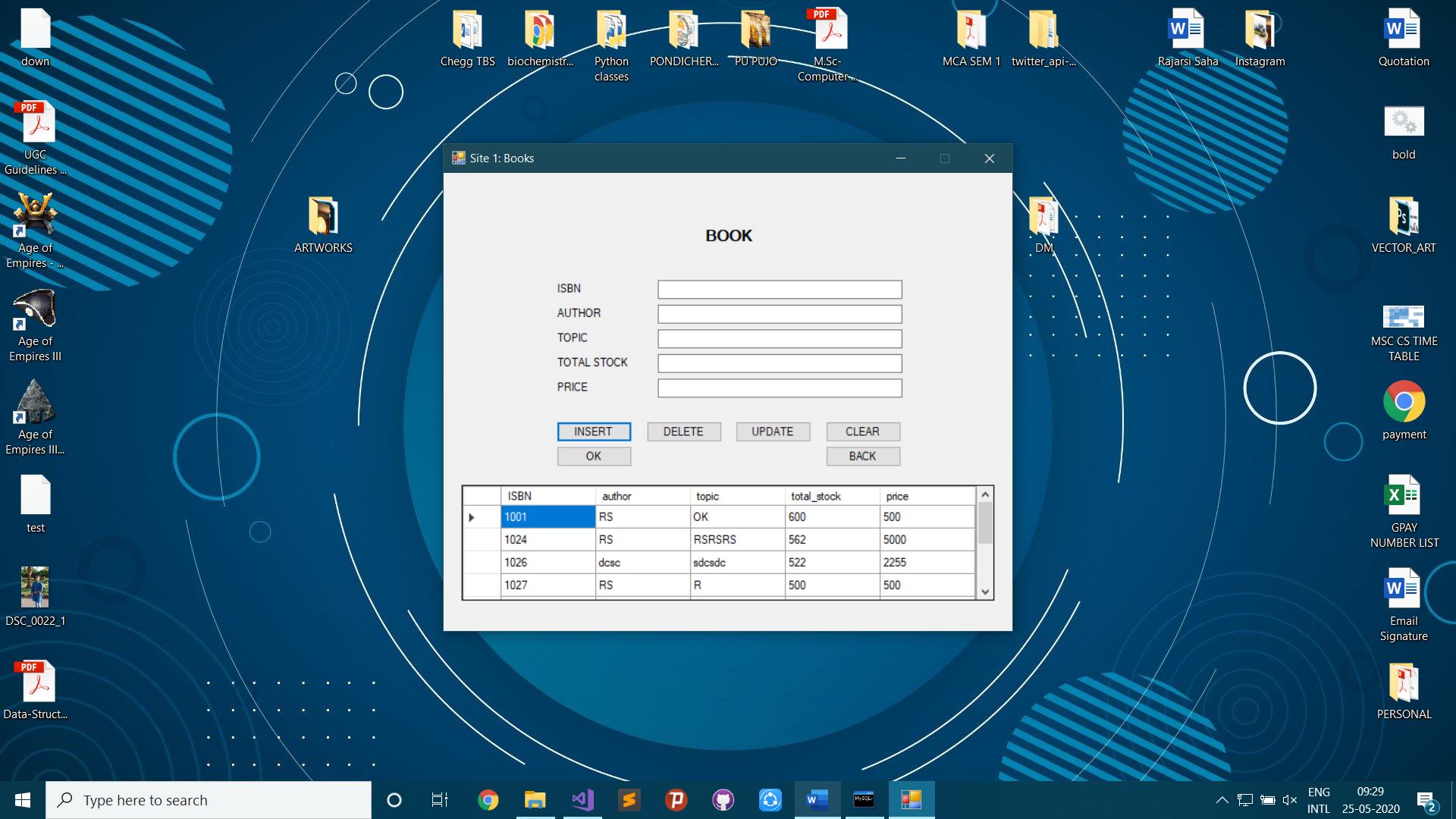
1. Front-end Design



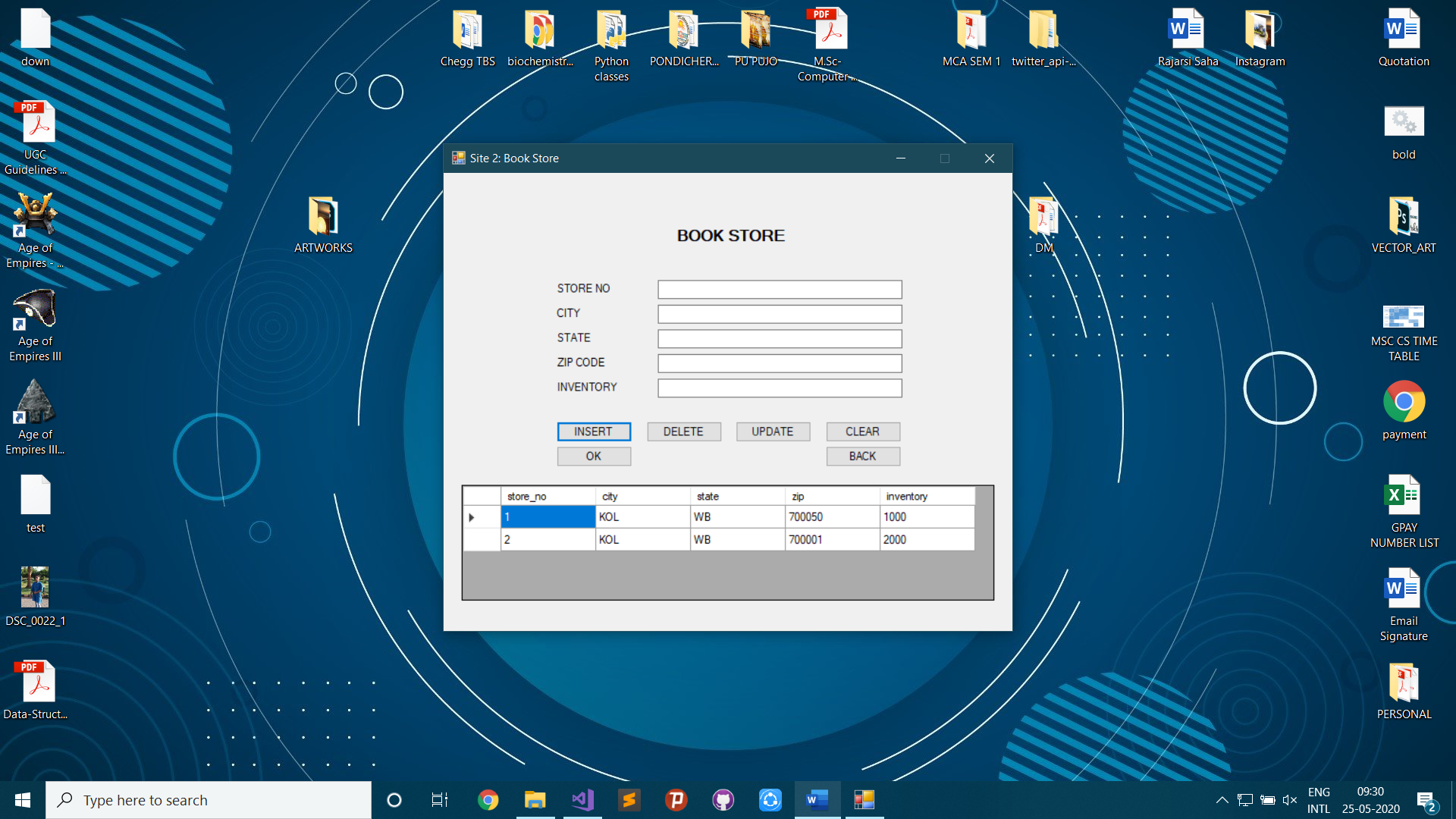
1. Distributed Database handling



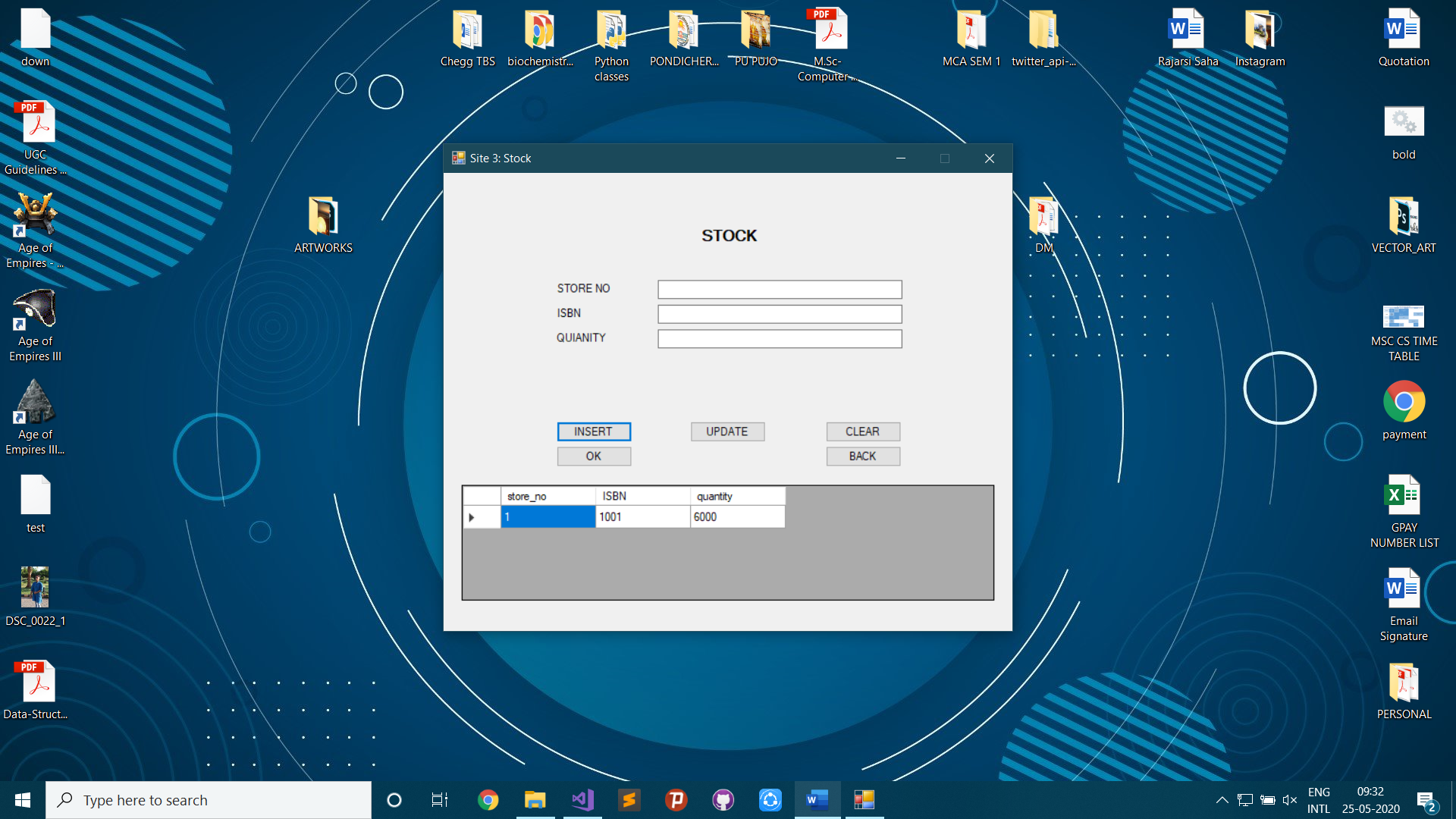
1. Insert Books in Site 1



1. Insert Book Store in Site 2



1. Insert Stock in Site 3



**VISUAL BASIC CODE:**

Form 1:

Public Class Form1

Public Shared Base As Integer

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Base = 1

Me.Hide()

Form2.Show()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

Base = 2

Me.Hide()

Form2.Show()

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

Base = 3

Me.Hide()

Form2.Show()

End Sub

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'Site3DataSet.stock' table. You can move, or remove it, as needed.

Me.StockTableAdapter2.Fill(Me.Site3DataSet.stock)

'TODO: This line of code loads data into the 'Site2DataSet.stock' table. You can move, or remove it, as needed.

Me.StockTableAdapter1.Fill(Me.Site2DataSet.stock)

'TODO: This line of code loads data into the 'Site1DataSet.stock' table. You can move, or remove it, as needed.

Me.StockTableAdapter.Fill(Me.Site1DataSet.stock)

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

TextBox1.Text = StockTableAdapter.ScalarQuery()

End Sub

Private Sub Button5\_Click(sender As Object, e As EventArgs) Handles Button5.Click

TextBox2.Text = StockTableAdapter1.ScalarQuery()

End Sub

Private Sub Button6\_Click(sender As Object, e As EventArgs) Handles Button6.Click

TextBox3.Text = StockTableAdapter2.ScalarQuery()

End Sub

Private Sub Button7\_Click(sender As Object, e As EventArgs) Handles Button7.Click

TextBox4.Text = Val(StockTableAdapter.ScalarQuery() + StockTableAdapter1.ScalarQuery() + StockTableAdapter2.ScalarQuery())

End Sub

End Class

Form 2:

Public Class Form2

Dim ButtonClicked As String

Private Sub Button7\_Click(sender As Object, e As EventArgs) Handles Button7.Click

ButtonClicked = Button7.Text

Me.Close()

End Sub

Private Sub Button8\_Click(sender As Object, e As EventArgs) Handles Button8.Click

ButtonClicked = Button8.Text

Me.Close()

End Sub

Private Sub Form2\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Me.Text = "Site " & Form1.Base & ": Books"

'TODO: This line of code loads data into the 'Site3DataSet.books' table. You can move, or remove it, as needed.

Me.BooksTableAdapter2.Fill(Me.Site3DataSet.books)

'TODO: This line of code loads data into the 'Site2DataSet.books' table. You can move, or remove it, as needed.

Me.BooksTableAdapter1.Fill(Me.Site2DataSet.books)

'TODO: This line of code loads data into the 'Site1DataSet.books' table. You can move, or remove it, as needed.

Me.BooksTableAdapter.Fill(Me.Site1DataSet.books)

If Form1.Base = 1 Then

Me.DataGridView1.DataSource = Site1DataSet

Me.DataGridView1.DataMember = BooksBindingSource.DataMember

ElseIf Form1.Base = 2 Then

Me.DataGridView1.DataSource = Site2DataSet

Me.DataGridView1.DataMember = BooksBindingSource1.DataMember

ElseIf Form1.Base = 3 Then

Me.DataGridView1.DataSource = Site3DataSet

Me.DataGridView1.DataMember = BooksBindingSource2.DataMember

End If

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

DataGridView1.ReadOnly = True

Try

If Form1.Base = 1 Then

BooksTableAdapter.Insert(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data inserted!", "Message")

Me.Site1DataSet.books.AcceptChanges()

Me.BooksTableAdapter.Fill(Me.Site1DataSet.books)

ElseIf Form1.Base = 2 Then

BooksTableAdapter1.Insert(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data inserted!", "Message")

Me.Site2DataSet.books.AcceptChanges()

Me.BooksTableAdapter1.Fill(Me.Site2DataSet.books)

ElseIf Form1.Base = 3 Then

BooksTableAdapter2.Insert(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data inserted!", "Message")

Me.Site3DataSet.books.AcceptChanges()

Me.BooksTableAdapter2.Fill(Me.Site3DataSet.books)

End If

Catch ex As Exception

MessageBox.Show(ex.Message, "Error")

End Try

TextBox1.Clear()

TextBox2.Clear()

TextBox3.Clear()

TextBox4.Clear()

TextBox5.Clear()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

DataGridView1.ReadOnly = True

Try

If Form1.Base = 1 Then

BooksTableAdapter.Delete(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data deleted!", "Message")

Me.Site1DataSet.books.AcceptChanges()

Me.BooksTableAdapter.Fill(Me.Site1DataSet.books)

ElseIf Form1.Base = 2 Then

BooksTableAdapter1.Delete(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data deleted!", "Message")

Me.Site2DataSet.books.AcceptChanges()

Me.BooksTableAdapter1.Fill(Me.Site2DataSet.books)

ElseIf Form1.Base = 3 Then

BooksTableAdapter2.Delete(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data deleted!", "Message")

Me.Site3DataSet.books.AcceptChanges()

Me.BooksTableAdapter2.Fill(Me.Site3DataSet.books)

End If

Catch ex As Exception

MessageBox.Show(ex.Message, "Error")

End Try

TextBox1.Clear()

TextBox2.Clear()

TextBox3.Clear()

TextBox4.Clear()

TextBox5.Clear()

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

DataGridView1.ReadOnly = True

TextBox1.Clear()

TextBox2.Clear()

TextBox3.Clear()

TextBox4.Clear()

TextBox5.Clear()

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

DataGridView1.ReadOnly = False

End Sub

Private Sub Form2\_FormClosing(sender As Object, e As FormClosingEventArgs) Handles Me.FormClosing

If ButtonClicked = "OK" Then

Form3.Show()

Else

Form1.Show()

End If

End Sub

End Class

Form 3:

Public Class Form3

Dim ButtonClicked As String

Private Sub Button7\_Click(sender As Object, e As EventArgs) Handles Button7.Click

ButtonClicked = Button7.Text

Me.Close()

End Sub

Private Sub Button8\_Click(sender As Object, e As EventArgs) Handles Button8.Click

ButtonClicked = Button8.Text

Me.Close()

End Sub

Private Sub Form3\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Me.Text = "Site " & Form1.Base & ": Book Store"

'TODO: This line of code loads data into the 'Site3DataSet.book\_store' table. You can move, or remove it, as needed.

Me.Book\_storeTableAdapter2.Fill(Me.Site3DataSet.book\_store)

'TODO: This line of code loads data into the 'Site2DataSet.book\_store' table. You can move, or remove it, as needed.

Me.Book\_storeTableAdapter1.Fill(Me.Site2DataSet.book\_store)

'TODO: This line of code loads data into the 'Site1DataSet.book\_store' table. You can move, or remove it, as needed.

Me.Book\_storeTableAdapter.Fill(Me.Site1DataSet.book\_store)

If Form1.Base = 1 Then

Me.DataGridView1.DataSource = Site1DataSet

Me.DataGridView1.DataMember = BookstoreBindingSource.DataMember

ElseIf Form1.Base = 2 Then

Me.DataGridView1.DataSource = Site2DataSet

Me.DataGridView1.DataMember = BookstoreBindingSource1.DataMember

ElseIf Form1.Base = 3 Then

Me.DataGridView1.DataSource = Site3DataSet

Me.DataGridView1.DataMember = BookstoreBindingSource2.DataMember

End If

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

DataGridView1.ReadOnly = False

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

DataGridView1.ReadOnly = True

TextBox1.Clear()

TextBox2.Clear()

TextBox3.Clear()

TextBox4.Clear()

TextBox5.Clear()

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

DataGridView1.ReadOnly = True

Try

If Form1.Base = 1 Then

Book\_storeTableAdapter.Insert(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data inserted!", "Message")

Me.Site1DataSet.book\_store.AcceptChanges()

Me.Book\_storeTableAdapter.Fill(Me.Site1DataSet.book\_store)

ElseIf Form1.Base = 2 Then

Book\_storeTableAdapter1.Insert(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data inserted!", "Message")

Me.Site2DataSet.book\_store.AcceptChanges()

Me.Book\_storeTableAdapter1.Fill(Me.Site2DataSet.book\_store)

ElseIf Form1.Base = 3 Then

Book\_storeTableAdapter2.Insert(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data inserted!", "Message")

Me.Site3DataSet.book\_store.AcceptChanges()

Me.Book\_storeTableAdapter2.Fill(Me.Site3DataSet.book\_store)

End If

Catch ex As Exception

MessageBox.Show(ex.Message, "Error")

End Try

TextBox1.Clear()

TextBox2.Clear()

TextBox3.Clear()

TextBox4.Clear()

TextBox5.Clear()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

DataGridView1.ReadOnly = True

Try

If Form1.Base = 1 Then

Book\_storeTableAdapter.Delete(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data deleted!", "Message")

Me.Site1DataSet.book\_store.AcceptChanges()

Me.Book\_storeTableAdapter.Fill(Me.Site1DataSet.book\_store)

ElseIf Form1.Base = 2 Then

Book\_storeTableAdapter1.Delete(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data deleted!", "Message")

Me.Site2DataSet.book\_store.AcceptChanges()

Me.Book\_storeTableAdapter1.Fill(Me.Site2DataSet.book\_store)

ElseIf Form1.Base = 3 Then

Book\_storeTableAdapter2.Delete(TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text)

MessageBox.Show("Data deleted!", "Message")

Me.Site3DataSet.book\_store.AcceptChanges()

Me.Book\_storeTableAdapter2.Fill(Me.Site3DataSet.book\_store)

End If

Catch ex As Exception

MessageBox.Show(ex.Message, "Error")

End Try

TextBox1.Clear()

TextBox2.Clear()

TextBox3.Clear()

TextBox4.Clear()

TextBox5.Clear()

End Sub

Private Sub Form3\_FormClosing(sender As Object, e As FormClosingEventArgs) Handles Me.FormClosing

If ButtonClicked = "OK" Then

Form4.Show()

Else

Form2.Show()

End If

End Sub

End Class

Form 4:

Public Class Form4

Dim ButtonClicked As String

Private Sub Button8\_Click(sender As Object, e As EventArgs) Handles Button8.Click

ButtonClicked = Button8.Text

Me.Close()

End Sub

Private Sub Button7\_Click(sender As Object, e As EventArgs) Handles Button7.Click

ButtonClicked = Button7.Text

Me.Close()

End Sub

Private Sub Form4\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Me.Text = "Site " & Form1.Base & ": Stock"

'TODO: This line of code loads data into the 'Site3DataSet.stock' table. You can move, or remove it, as needed.

Me.StockTableAdapter2.Fill(Me.Site3DataSet.stock)

'TODO: This line of code loads data into the 'Site2DataSet.stock' table. You can move, or remove it, as needed.

Me.StockTableAdapter1.Fill(Me.Site2DataSet.stock)

'TODO: This line of code loads data into the 'Site1DataSet.stock' table. You can move, or remove it, as needed.

Me.StockTableAdapter.Fill(Me.Site1DataSet.stock)

If Form1.Base = 1 Then

Me.DataGridView1.DataSource = Site1DataSet

Me.DataGridView1.DataMember = StockBindingSource.DataMember

ElseIf Form1.Base = 2 Then

Me.DataGridView1.DataSource = Site2DataSet

Me.DataGridView1.DataMember = StockBindingSource1.DataMember

ElseIf Form1.Base = 3 Then

Me.DataGridView1.DataSource = Site3DataSet

Me.DataGridView1.DataMember = StockBindingSource2.DataMember

End If

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

DataGridView1.ReadOnly = True

TextBox1.Clear()

TextBox2.Clear()

TextBox3.Clear()

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

DataGridView1.ReadOnly = False

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

DataGridView1.ReadOnly = True

Try

If Form1.Base = 1 Then

StockTableAdapter.Insert(TextBox1.Text, TextBox2.Text, TextBox3.Text)

MessageBox.Show("Data inserted!", "Message")

Me.Site1DataSet.stock.AcceptChanges()

Me.StockTableAdapter.Fill(Me.Site1DataSet.stock)

ElseIf Form1.Base = 2 Then

StockTableAdapter1.Insert(TextBox1.Text, TextBox2.Text, TextBox3.Text)

MessageBox.Show("Data inserted!", "Message")

Me.Site2DataSet.stock.AcceptChanges()

Me.StockTableAdapter1.Fill(Me.Site2DataSet.stock)

ElseIf Form1.Base = 3 Then

StockTableAdapter2.Insert(TextBox1.Text, TextBox2.Text, TextBox3.Text)

MessageBox.Show("Data inserted!", "Message")

Me.Site3DataSet.stock.AcceptChanges()

Me.StockTableAdapter2.Fill(Me.Site3DataSet.stock)

End If

Catch ex As Exception

MessageBox.Show(ex.Message, "Error")

End Try

TextBox1.Clear()

TextBox2.Clear()

TextBox3.Clear()

End Sub

Private Sub Form4\_FormClosing(sender As Object, e As FormClosingEventArgs) Handles Me.FormClosing

If ButtonClicked = "OK" Then

Form1.Show()

Else

Form3.Show()

End If

End Sub

End Class

**FEW RELATED DEFINITIONS:**

**Database:** A **database** is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex, they are often developed using formal design and modelling techniques.

**Database Management System (DBMS):**

**The database management system (DBMS)** is the software that interacts with end users, applications, and the database itself to capture and analyse the data. The DBMS software additionally encompasses the core facilities provided to administer the database.



The sum total of the database, the DBMS and the associated applications can be referred to as a "database system". Often the term "database" is also used to loosely refer to any of the DBMS, the database system or an application associated with the database.

**Windows Forms:**

**Windows Forms (WinForms)** is a graphical (GUI) class library included as a part of Microsoft .NET Framework or Mono Framework, providing a platform to write rich client applications for desktop, laptop, and tablet PCs. While it is seen as a replacement for the earlier and more complex C++ based Microsoft Foundation Class Library, it does not offer a comparable paradigm and only acts as a platform for the user interface tier in a multi-tier solution.



At the Microsoft Connect event on December 4, 2018, Microsoft announced releasing Windows Forms as an open source project on GitHub. It is released under the MIT License. With this release, Windows Forms has become available for projects targeting the .NET Core framework. However, the framework is still available only on the Windows platform, and Mono's incomplete implementation of WinForms remains the only cross-platform implementation.