Table of Contents

Page

[**1. Project Proposals**](#_qp39sjq675l6) **1**

[(I) Project Goal](#_98g1yix40hu2) 1

[(II) Project Objectives](#_7qr7wbsk1er8) 1

[(iii) Project Scope](#_ky3y5ytlil7f) 1

[1.Context](#_ilv4wjyirwma) 1

[2. Target Language Platform](#_ck0y44cyepa1) 1

[a) Microsoft SQL Server](#_7a9a7w8x4e7k) 1

[b) ASP.NET](#_5t3cy1zeu3rc) 2

[3. Stakeholders of the product](#_u7120lu5dju6) 2

4. What is Web Application 3

[**2. Analysis Documentation**](#_vcv6djdruvcm) **4**

[a) Problem with existing System](#_c8ilrylomu98) 4

[b) Proposed System](#_lp2mnsn8fohm)

**3. Hardware Requirement 5**

**4. Software Requirement 5**

**5. Design Documentation 6**

1. Data flow diagram 6

a)Context Diagram 6

b) Level 1 6

2) ER Diagram 7

**6. Design Documentation 8**

a) [Creating ASP.NET Core Web Application in Visual Studio 2017](#_41ldwkvbutzb) 8

**7. Summary 14**

**8. References**  **15**

Introduction

# 1. Project Proposals

## **(I) Project Goal**

We will have a Web Application named ‘Book Library’ which is a database driven website.

## **(II) Project Objectives**

The project objective is to

* Design a web application that can handle and manage the activities the CRUD(Create, Read, Update, and Delete) for the administrator to easily handle the library activities in an efficient and reliable way.
* Develop a system that can replace a manual library managing system.
* Develop a database which contains book details and admin details.
* Administrators or Librarians should have logins
* Create an attractive, easy to understand, and user-friendly interface.
* Reducing time of manual work.
* Reducing human error and redundancy in the database.

## **(iii) Project Scope**

### **1.Context**

We will create a Web Application named ‘Book Library’ which is a database driven website.

We will give CRUD(Create, Read, Update, and Delete) activities to the administrator or librarian.

We will use Microsoft SQL Server database to store book details and Administrators or Librarians details.

### **2. Target Language Platform**

#### **a) Microsoft SQL Server**

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).  
Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

#### **b) 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. Moving forward, ASP.NET Core is the framework that can be used for web development with .NET. If you have any experience with MVC or Web API over the last few years, you will notice some familiar features. At the end this tutorial, you will have everything you need to start using ASP.NET Core and write an application that can create, edit, and view data from a database.

#### 

### **3. Stakeholders of the product**

The end user of this project and web application will be the administrators and librarians.

**4. What is Web Application**

A web application is a computer program that utilizes web browsers and web technology to perform tasks over the Internet.

**OVERVIEW**

Millions of businesses use the Internet as a cost-effective communications channel. It lets them exchange information with their target market and make fast, secure transactions. However, effective engagement is only possible when the business is able to capture and store all the necessary data, and have a means of processing this information and presenting the results to the user.  
  
Web applications use a combination of server-side scripts (PHP and ASP) to handle the storage and retrieval of the information, and client-side scripts (JavaScript and HTML) to present information to users. This allows users to interact with the company using online forms, content management systems, shopping carts and more. In addition, the applications allow employees to create documents, share information, collaborate on projects, and work on common documents regardless of location or device.

**HOW A WEB APPLICATION WORKS**

Web applications are usually coded in browser-supported language such as JavaScript and HTML as these languages rely on the browser to render the program executable. Some of the applications are dynamic, requiring server-side processing. Others are completely static with no processing required at the server.  
  
The web application requires a web server to manage requests from the client, an application server to perform the tasks requested, and, sometimes, a database to store the information. Application server technology ranges from ASP.NET, ASP and ColdFusion, to PHP and JSP.  
  
**Here’s what a typical web application flow looks like**:  
  
User triggers a request to the web server over the Internet, either through a web browser or the application’s user interface  
Web server forwards this request to the appropriate web application server  
Web application server performs the requested task – such as querying the database or processing the data – then generates the results of the requested data  
Web application server sends results to the web server with the requested information or processed data  
Web server responds back to the client with the requested information that then appears on the user’s display

**EXAMPLE OF A WEB APPLICATION**

Web applications include online forms, shopping carts, word processors, spreadsheets, video and photo editing, file conversion, file scanning, and email programs such as Gmail, Yahoo and AOL. Popular applications include Google Apps and Microsoft 365.  
  
Google Apps for Work has Gmail, Google Docs, Google Sheets, Google Slides, online storage and more. Other functionalities include online sharing of documents and calendars. This lets all team members access the same version of a document simultaneously.

# 2. Analysis Documentation

Analysis of the project is to do a detailed study of the various operations performed  
by a web application. Here the key question is- what all problems exist in the present system? What must be done to solve the problem? Analysis begins when an admin begins a study of the program using the existing system.

## a) Problem with existing System

In our existing system all the transaction of books are done manually, So taking  
more time for a transaction like performing CRUD(Create, Read, Update, and Delete) activities on books present in the database. So after conducting the feasibility study, we decided to make the manual Library management system to be computerized.

## b) Proposed System

The proposed system is to:-

* Design a web application that can handle and manage the activities the CRUD(Create, Read, Update, and Delete) for the administrator to easily handle the library activities in an efficient and reliable way.
* Develop a system that can replace a manual library managing system.
* Develop a database which contains book details and admin details.
* Administrators or Librarians should have logins
* Create an attractive, easy to understand, and user-friendly interface.
* Reducing time of manual work.
* Reducing human error and redundancy in the database.
* More Storage Capacity
* Fast access to the database

3. Hardware Configuration

Processor: Intel Core i3 3.24GHz  
RAM: 2 GB

4. Software Configuration

Operating System: Windows 7, Window 10   
Language: ASP.NET

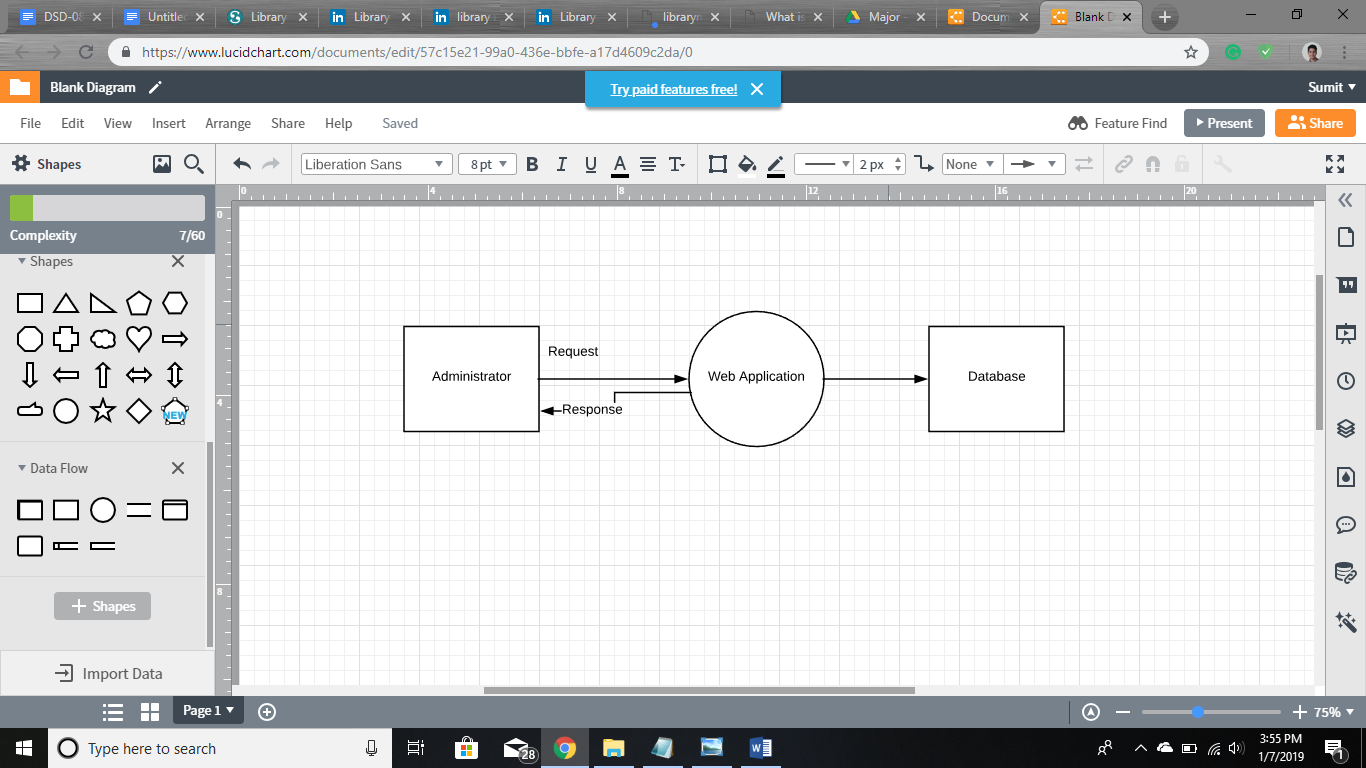
Database: Microsoft SQL Server

Application: Visual Studio 2017

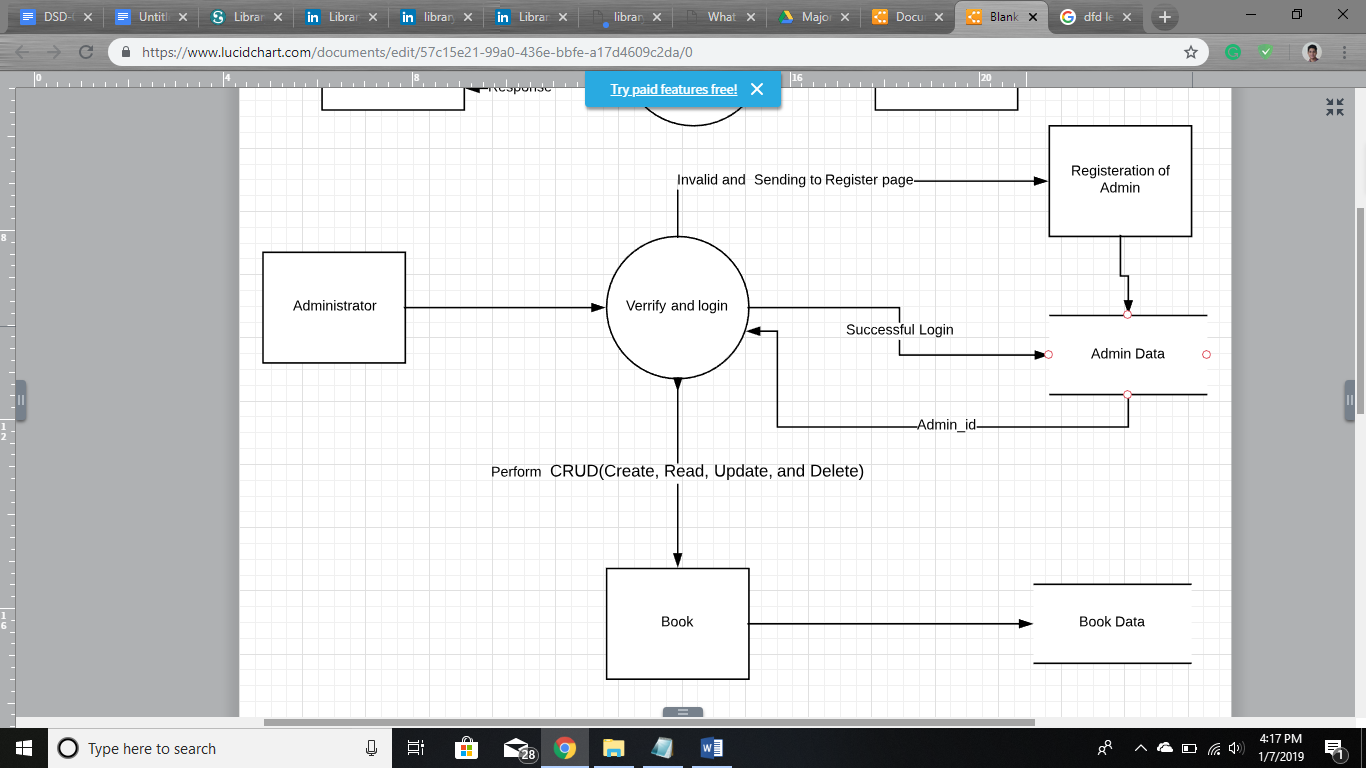
**5. Design Documentation**

**Data Flow diagram**

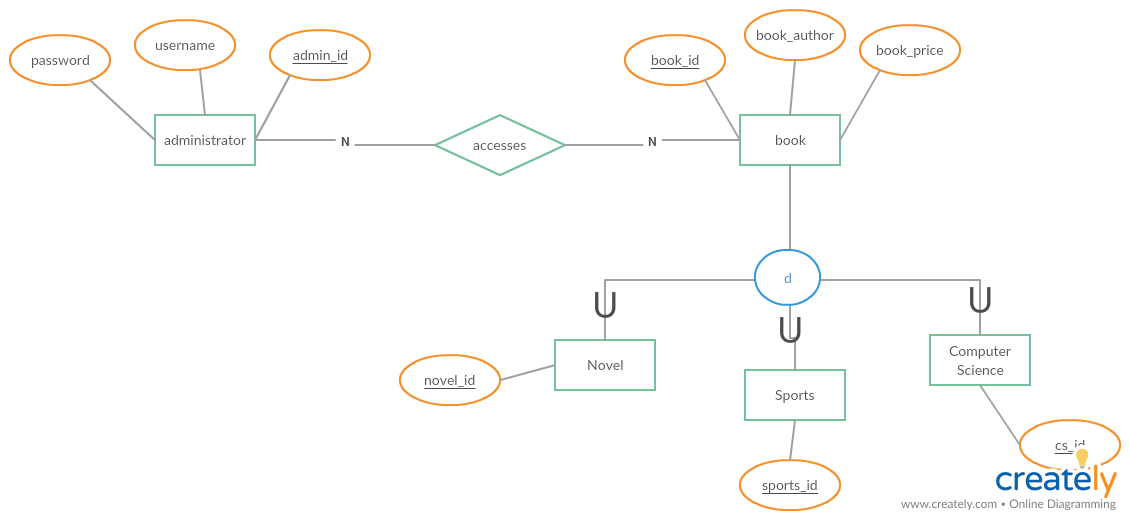
**a) Context Diagram**



b) Level 1



c) Entity- Relationship Diagram



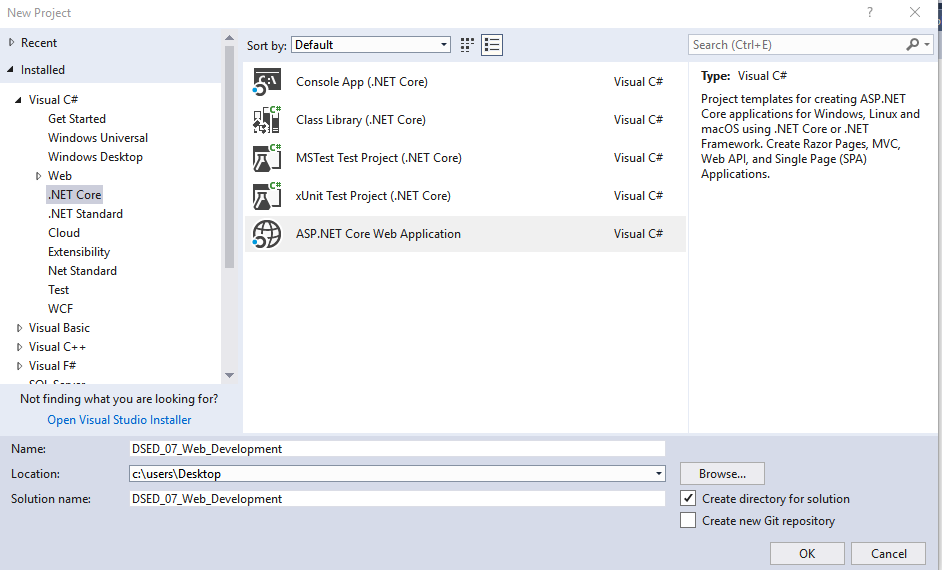
Customer

6. Design Documentation

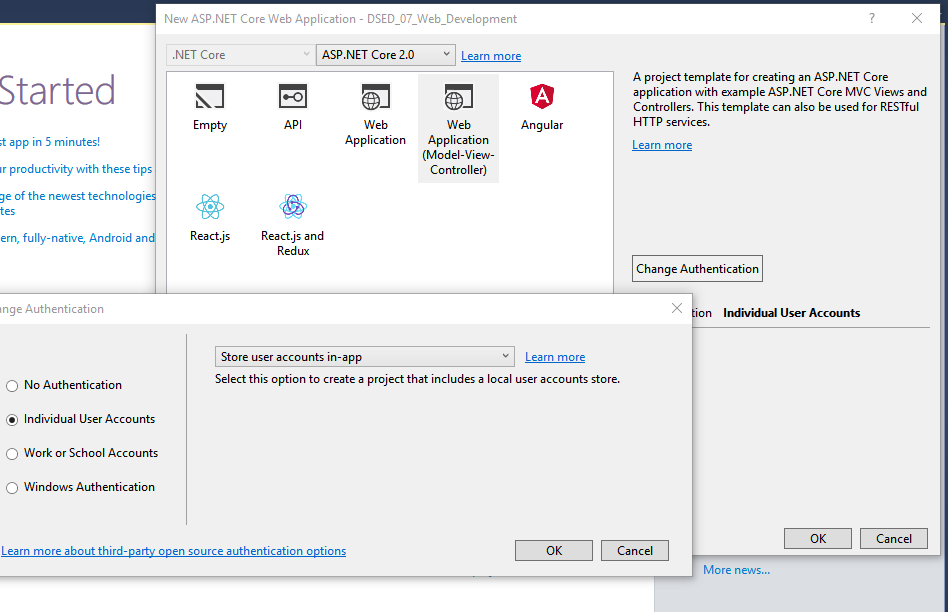
## 

## **Creating ASP.NET Core Web Application in Visual Studio 2017**

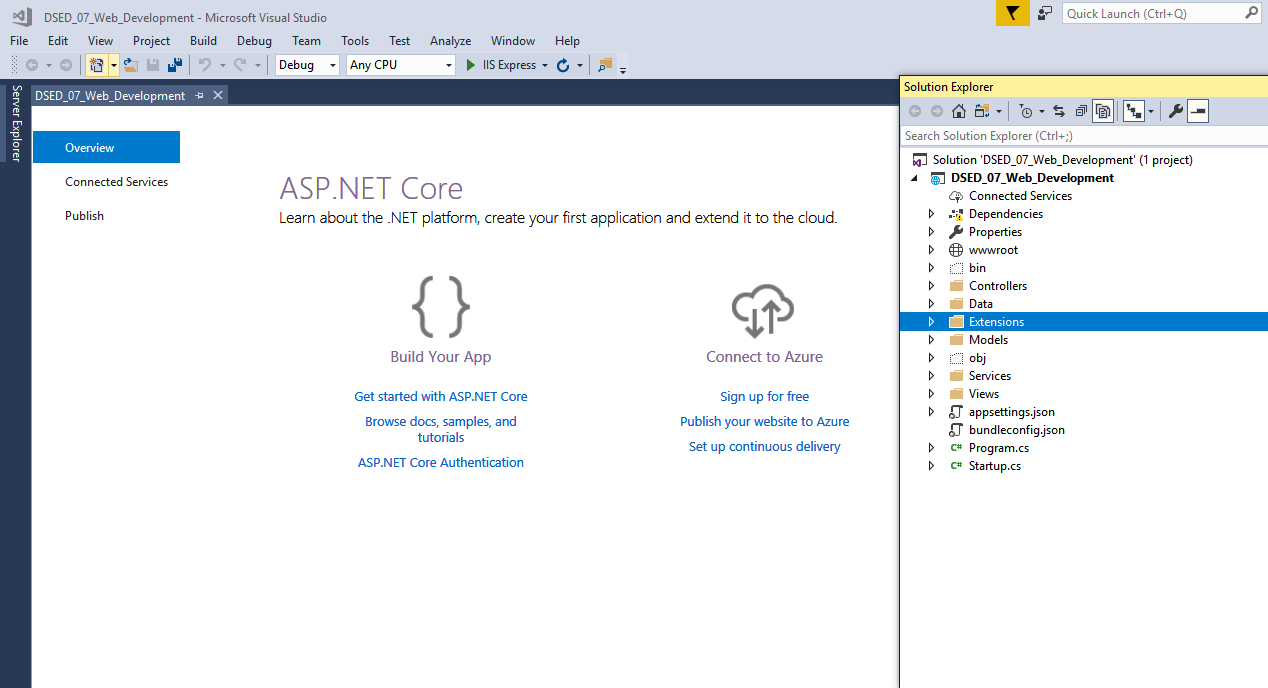
Firstly, Launch Microsoft Visual Studio 2017. Create a New Project, when prompted choose ASP.NET Core Web Application.



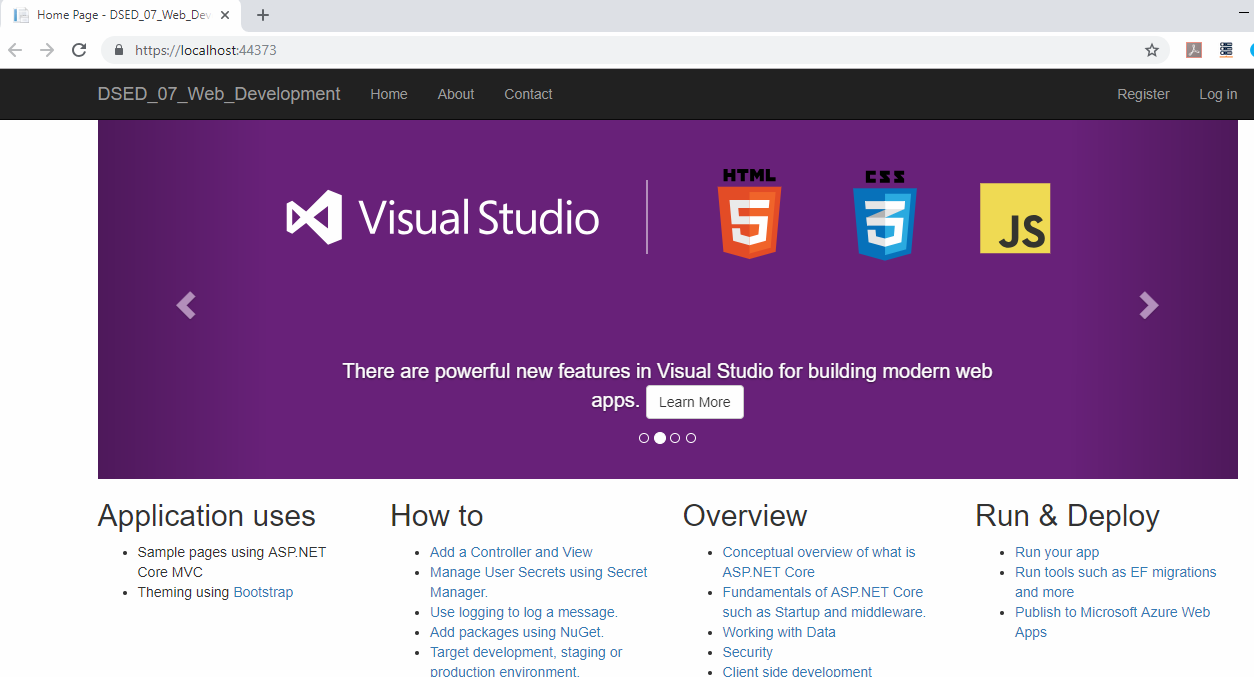
When prompted choose ASP.NET Core 2.0, the Web Application MVC template and change Authentication to Individual User Accounts for Login and Register, in the lab computers do not “Configure for HTTPS” for now, and then click OK.



After Creating ASP.Net Web Project, Below Solution Explorer will create : -



Build and Run this default website on IIS Express:-

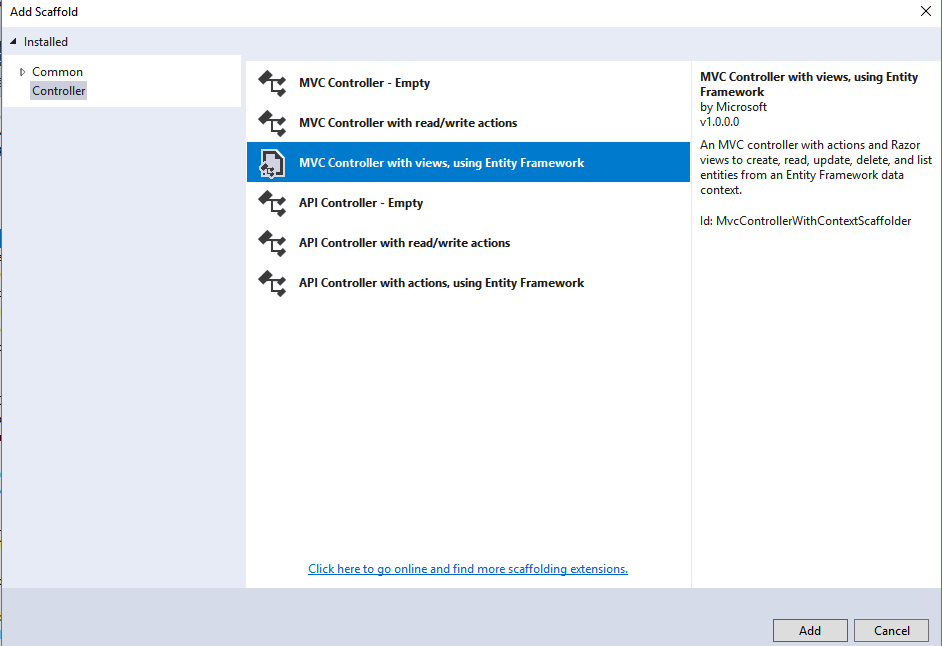


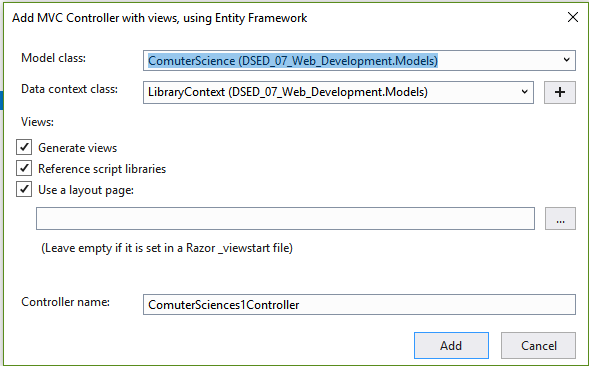
#### **Adding Scaffolding to Create CRUD :-**

Asp.net will provide a template to generate controllers and views based on the database.

Right-click the Controllers folder in Solution Explorer, select **Add | Controller…**

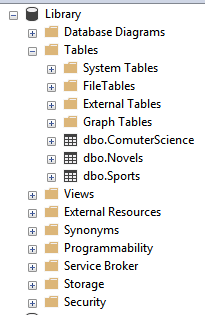
Add an **MVC Controller with views using Entity Framework** called ComuterScienceController





### **Creating a Database First ASP.NET Core Project Using MSSQL-SERVER :-**

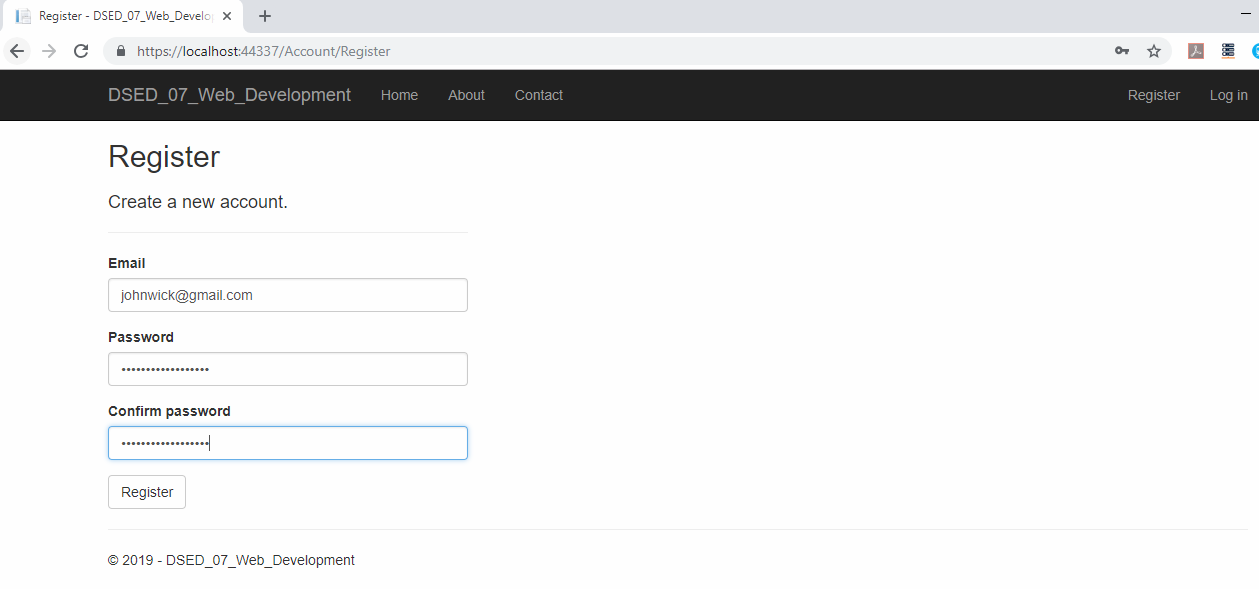
1. Create New DATABASE Name Library.
2. Add Tables to DATABASE (Created Table computer science, Novels and Sports)
3. Fill all Details in tables.

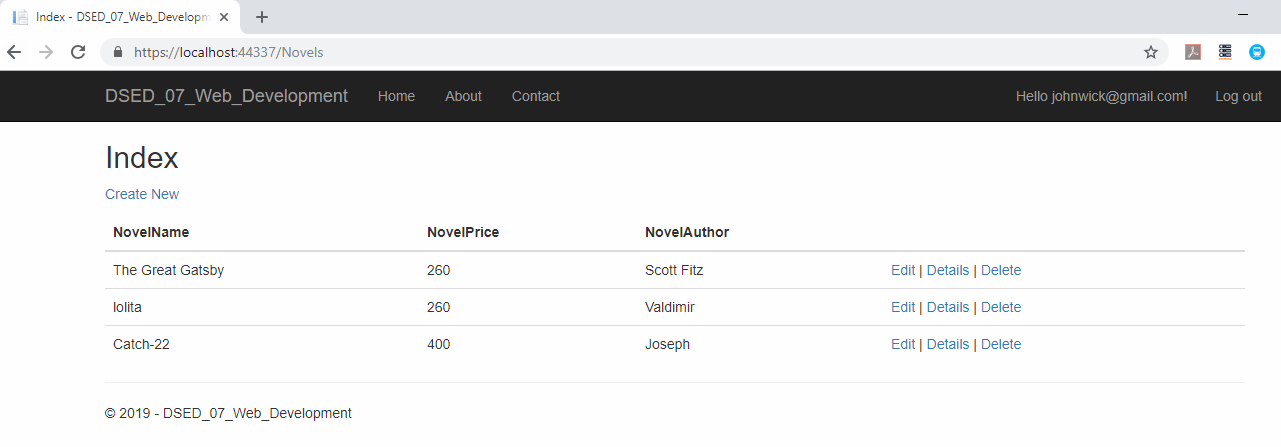


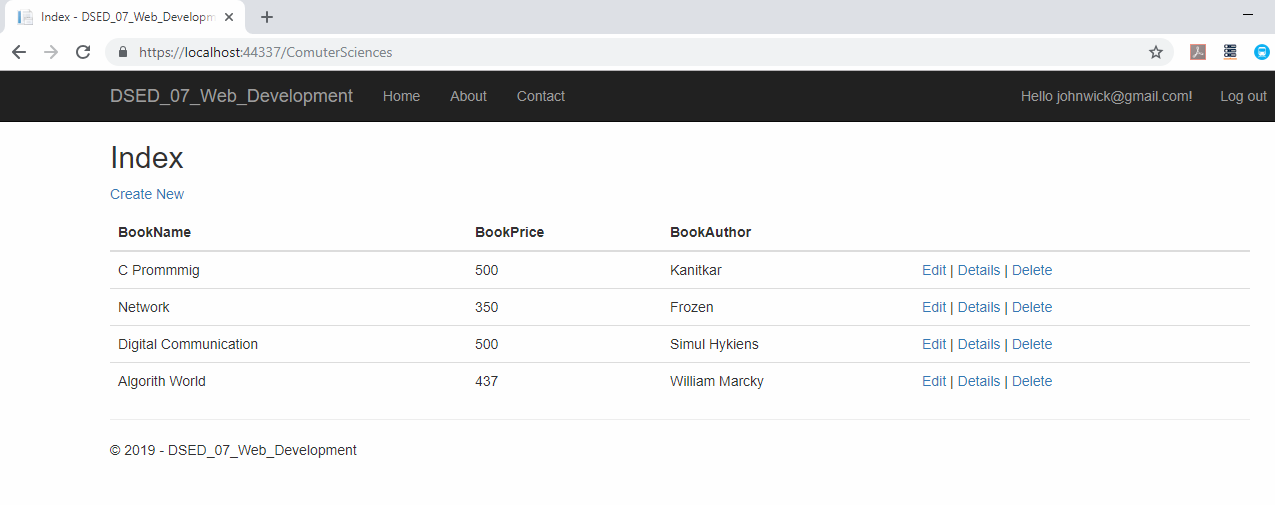
**After Running Projects following views will Show:**

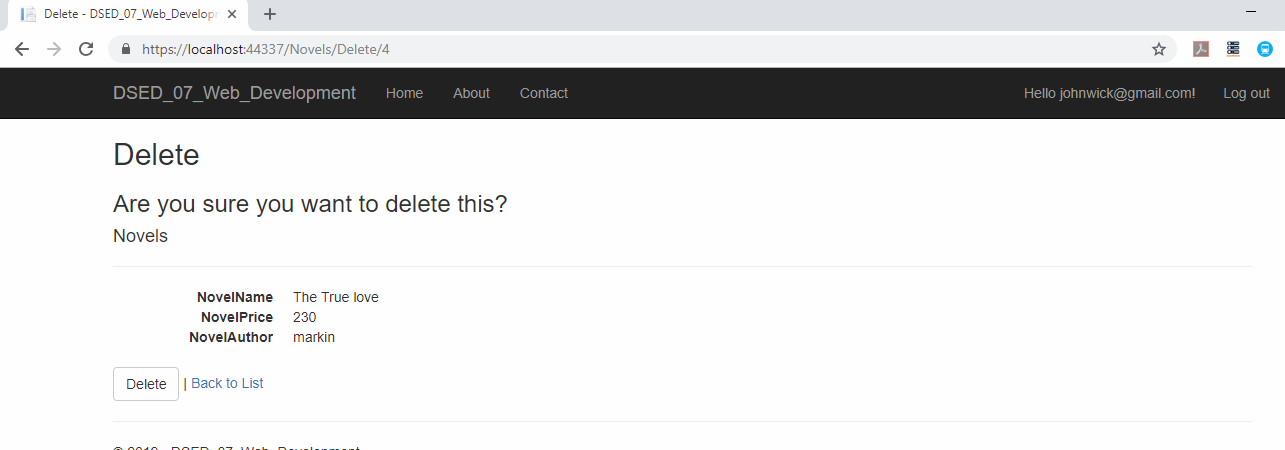
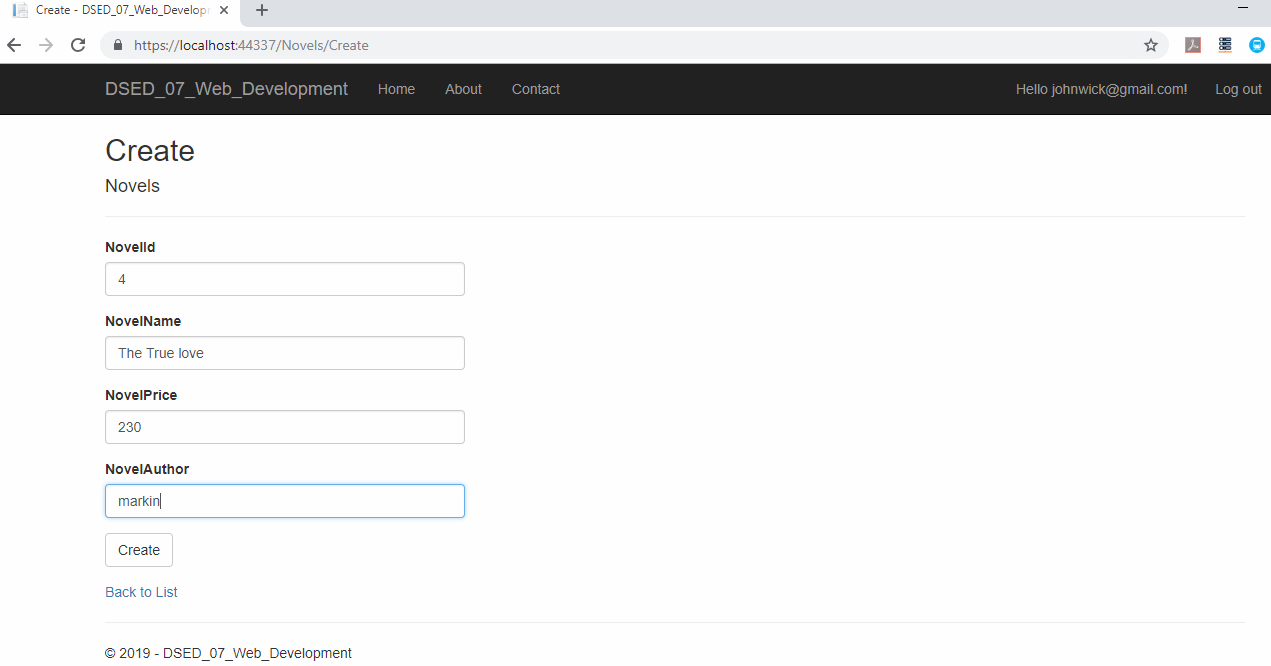
**A screenshot of a cell phone

Description automatically generated**









7. Summary:

This project is Created in ASP.net Core for web application with individual user account.it is created on MVC Framework with Database back End using Entity Framework, With LINQ. In Web Application, Admin Can Handles All Details About Database. Admin Can Edit, View, Add and Delete. Database Backend for this project is created using MSSQL-Server.

**8. References**

1.https://www.maxcdn.com/one/visual-glossary/web-application/  
2.https://www.tutorialspoint.com/asp.net\_core/asp.net\_core\_overview.htm  
3.https://en.wikipedia.org/wiki/Microsoft\_SQL\_Server  
4.https://en.wikipedia.org/wiki/C\_Sharp\_(programming\_language)