

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN
KHOA CÔNG NGHỆ THÔNG TIN

----- oOo -----



BÀI TẬP THỰC HÀNH

LẬP TRÌNH WEB API

Phiên bản: 2.0

Hưng Yên – Tháng 9 năm 2020

BÀI THỰC HÀNH SỐ 1: TẠO VÀ KIỂM TRA ỨNG DỤNG WEB API

A. MỤC TIÊU BÀI THỰC HÀNH

Sau bài thực hành này sinh viên có thể:

- Trình bày các bước tạo một dự án Web API
- Viết được các hàm GET, POST, PUT, DELETE đơn giản trong Web API
- Sử dụng thành thạo các công cụ kiểm tra Web API

B. ĐIỀU KIỆN THỰC HÀNH

Trong bài thực hành này, sinh viên cần kiểm tra và chắc chắn các phần mềm sau trên máy tính còn hoạt động tốt:

1. Visual Studio 2017/2019
2. Postman
3. Fiddler

C. TÀI NGUYÊN THAM CHIẾU

Để hoàn thành tốt bài thực hành này sinh viên nên tham khảo các tài nguyên sau:

STT	Tên tài nguyên	Mô tả tài nguyên
1	ASP.NET Web API Tutorial for Beginners and Professional	Tài liệu tham khảo về ASP.NET Web API

D. YÊU CẦU BÀI TẬP THỰC HÀNH

Xây dựng Web API cho ứng dụng quản lý bán hàng Online với các yêu cầu sau:

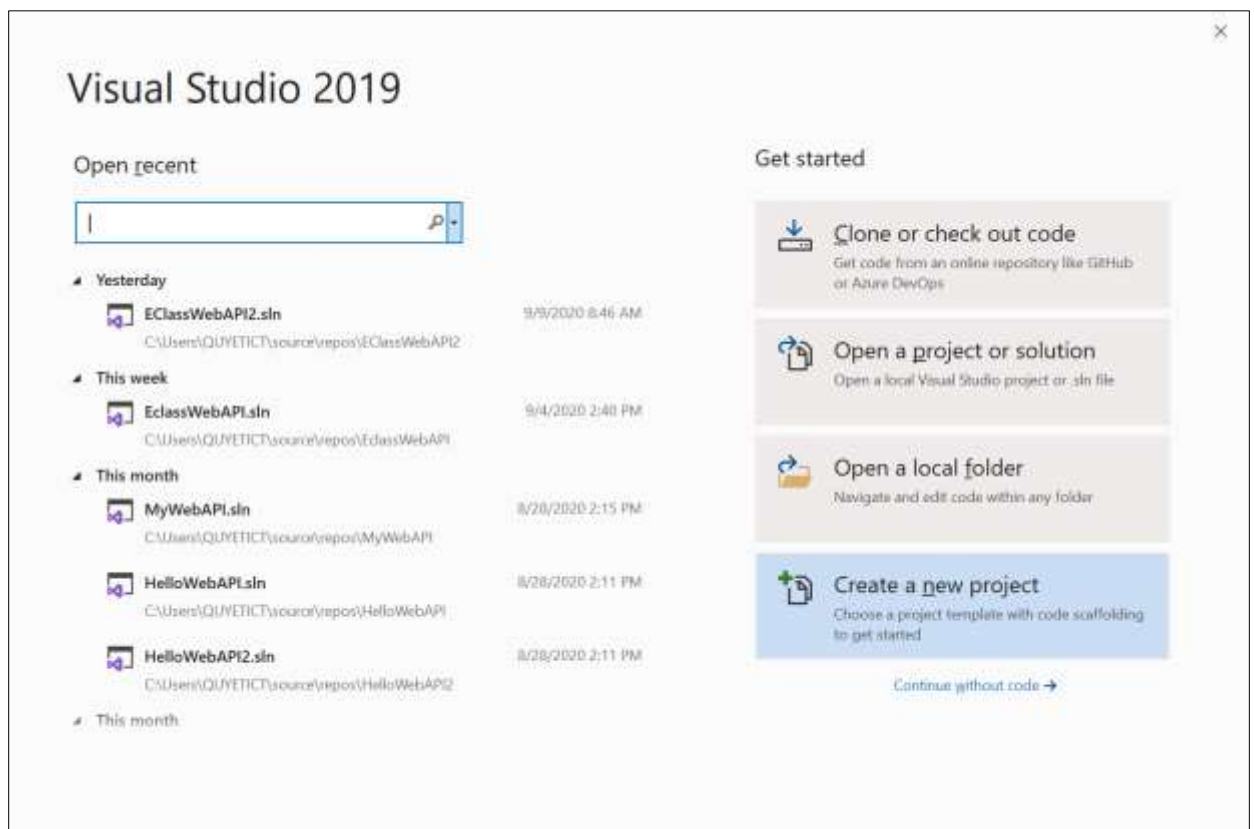
1. Tạo dự án ASP.NET Web API có tên **EShopWebAPI**
2. Tạo ra các lớp đối tượng để quản lý các loại sản phẩm và sản phẩm, trong đó:
 - Loại sản phẩm (Category) có các thuộc tính sau: Mã loại (CategoryID), tên loại (CategoryName) và mô tả (Description)

- Sản phẩm có các thuộc tính: Mã sản phẩm (ProductID), Mã loại (CategoryID), ProductName (Tên sản phẩm), UnitPrice (Đơn giá), Số lượng (Quantity).
3. Tạo Web API Controller để thực hiện việc lấy về, thêm mới, chỉnh sửa, hoặc xóa thông tin của các **loại sản phẩm**.
 4. Thực hiện kiểm tra các hàm API đã xây dựng ở **yêu cầu 3** sử dụng công cụ Swagger.
 5. Tạo Web API Controller để thực hiện việc lấy về, thêm mới, chỉnh sửa, hoặc xóa thông tin của các **sản phẩm**.
 6. Thực hiện kiểm tra các hàm API đã xây dựng ở **yêu cầu 5** sử dụng các công cụ Postman, và Fiddler.

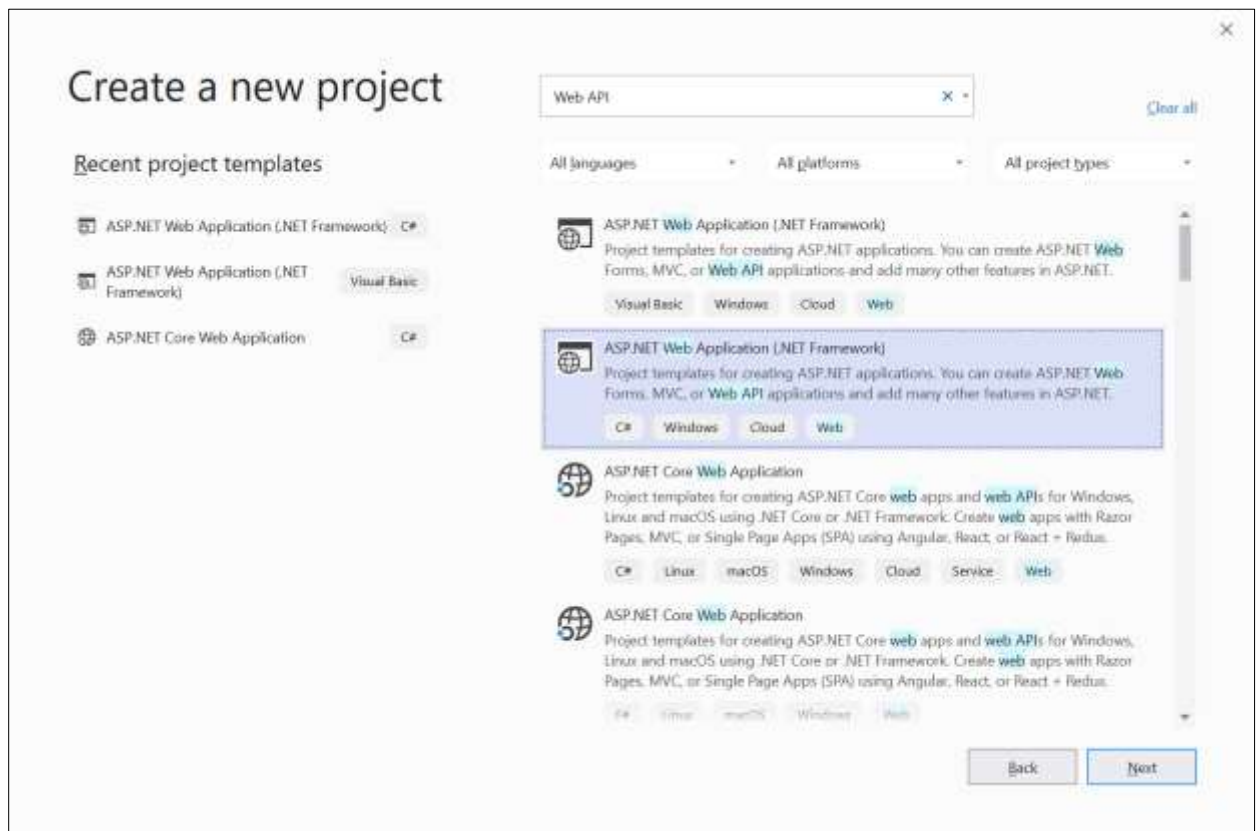
E. HƯỚNG DẪN THỰC HIỆN

1. Tạo dự án ASP.NET Web API có tên HomeShopping

Bước 1: Chọn **Create a new project** từ giao diện trang bắt đầu (Start Page) của Visual Studio 2019

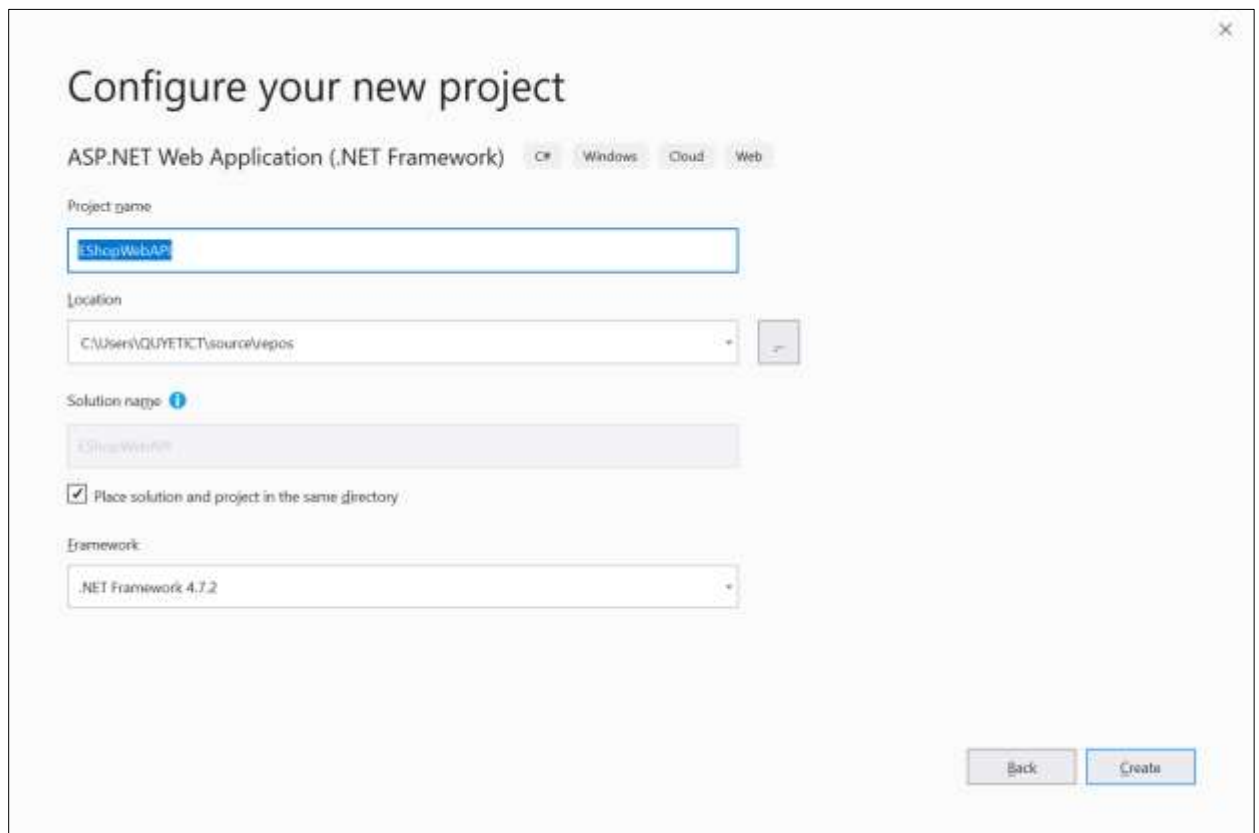


Bước 2: Giao diện **Create a new project** hiển thị, bạn nhập vào ô tìm kiếm từ khóa “Web API”, danh sách các công nghệ hỗ trợ ứng dụng Web API hiện ra, bạn chọn **ASP.NET Web Application (.NET Framework)** với C# → nhấn **Next**.



Bước 3: Giao diện **Configure your new project** hiển thị, bạn đặt tên dự án là **“EShopWebAPI”** → nhấn **Create**.

Ngoài ra bạn có thể chọn thay đổi nơi lưu trữ dự án ở ô chọn **Location** trước khi nhấn **Create**.



Configure your new project

ASP.NET Web Application (.NET Framework) **CF** Windows Cloud Web

Project name

ShopWebAPI

Location

C:\Users\QUYETICT\source\repos

Solution name ⓘ

ShopWebAPI

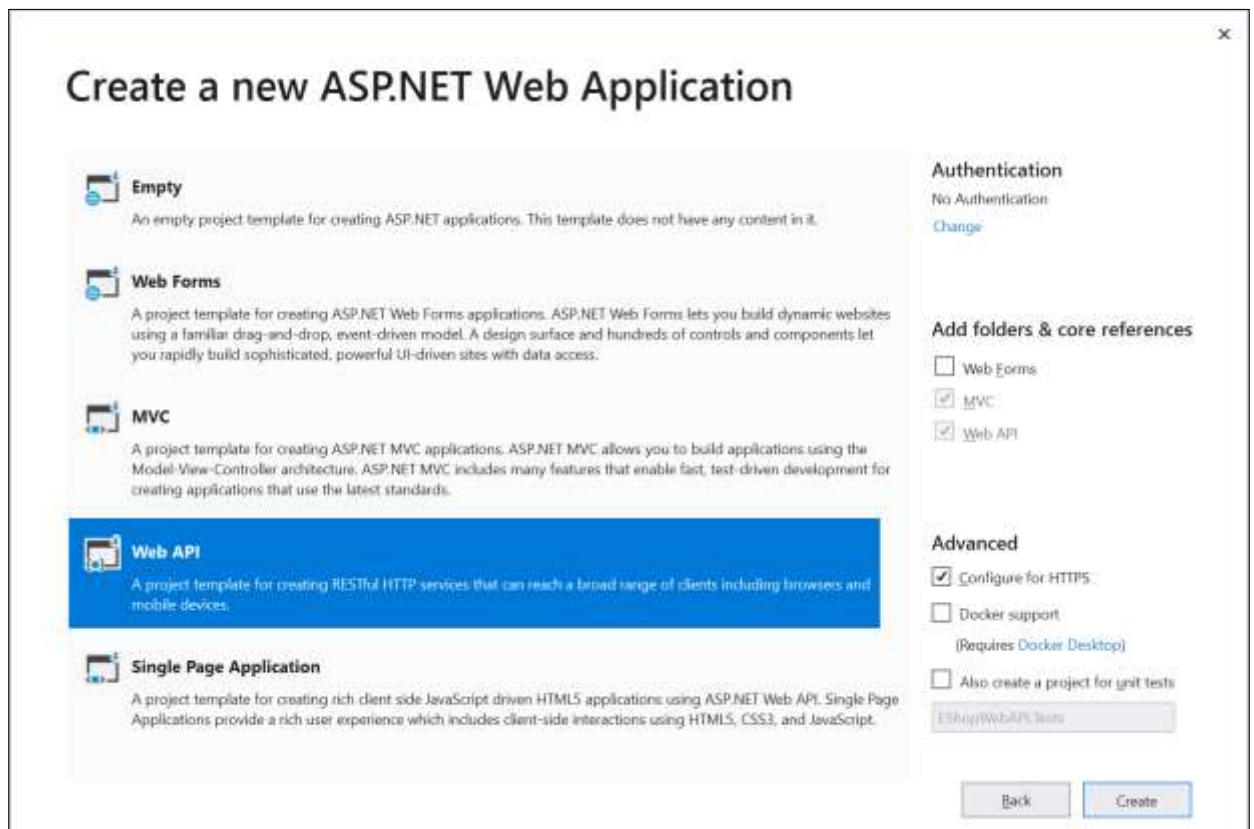
☒ Place solution and project in the same directory

Framework

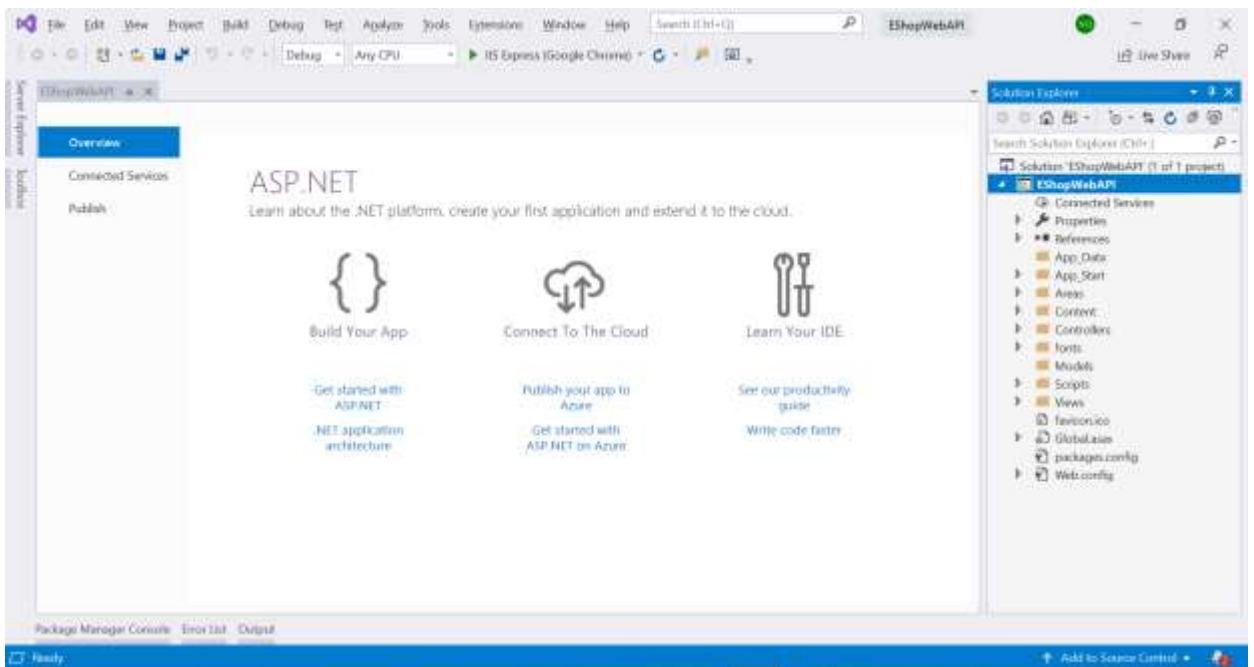
.NET Framework 4.7.2

Back Create

Bước 4: Giao diện **Create a new ASP.NET Web Application** hiển thị, bạn chọn **Web API**. Ở phần cấu hình bên phải, trong bài thực hành này, bạn có thể chọn **No Authentication; MVC** và **Web API**; bỏ qua **Configure for HTTPS** → nhấn **Create**.



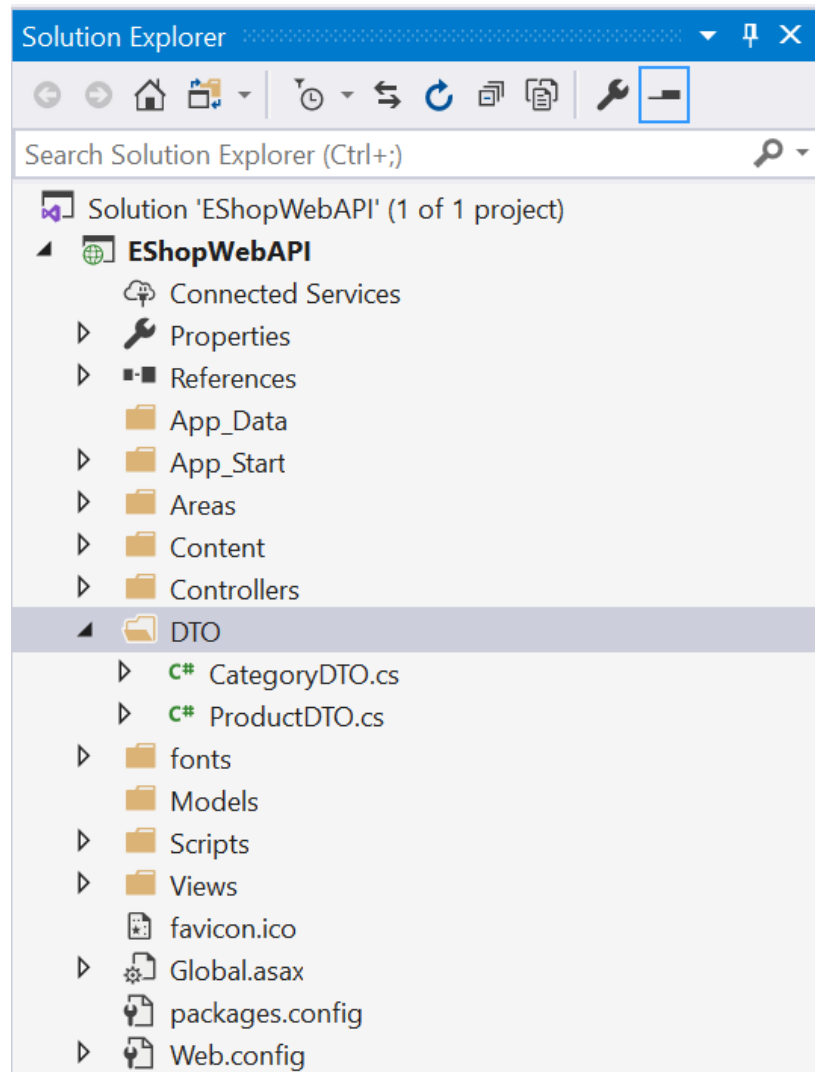
Giao diện dự án Web API sẽ hiển thị như hình dưới.



2. Tạo ra các lớp đối tượng để quản lý các loại sản phẩm và sản phẩm

Bước 1: Tạo thư mục **DTO** trong dự án **EShopWebAPI**

Bước 2: Tạo các lớp **ProductDTO** và **CategoryDTO** trong thư mục **DTO**



Bước 3: Lập trình tạo các lớp đối tượng như dưới đây:

CategoryDTO.cs

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

namespace EShopWebAPI.DTO
{
    public class CategoryDTO
    {
        private int categoryID;
        private string categoryName;
    }
}
```

```
        private string description;

        public int CategoryID { get => categoryID; set => categoryID = value; }
        public string CategoryName { get => categoryName; set => categoryName = value; }
        public string Description { get => description; set => description = value; }

        public CategoryDTO()
        {
        }
        public CategoryDTO(int categoryID, string categoryName, string description)
        {
            this.categoryID = categoryID;
            this.categoryName = categoryName;
            this.description = description;
        }
    }
}
```

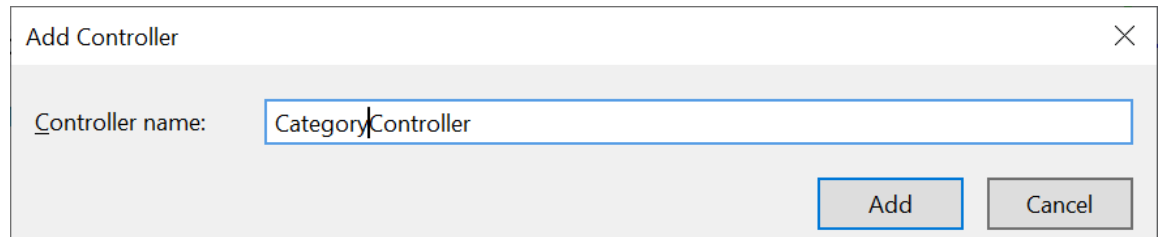
ProductDTO.cs

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

namespace EShopWebAPI.DTO
{
    public class ProductDTO
    {
        private int productID;
        private string productName;
        private int categoryID;
        private int unitPrice;
        private int quantity;
        public int ProductID { get => productID; set => productID = value; }
        public string ProductName { get => productName; set => productName = value; }
        public int CategoryID { get => categoryID; set => categoryID = value; }
        public int UnitPrice { get => unitPrice; set => unitPrice = value; }
        public int Quantity { get => quantity; set => quantity = value; }
        public ProductDTO()
        {
        }
        public ProductDTO(int productID, string productName, int categoryID, int
unitPrice, int quantity)
        {
            this.productID = productID;
            this.productName = productName;
            this.categoryID = categoryID;
            this.unitPrice = unitPrice;
            this.quantity = quantity;
        }
    }
}
```


3. Tạo Web API Controller

- Tạo CategoryController:
 - *Bước 1:* Right-Click vào thư mục **Controllers** của dự án, chọn **Add → Controller**, giao diện **Add New Scaffolded Item** hiển thị ra, chọn **Web API 2 Controller with read/write actions** → nhấn **Add**.
 - *Bước 2:* Đặt tên Controller là CategoryController



The screenshot shows a dialog box titled "Add Controller" with a close button (X) in the top right corner. Below the title bar, there is a label "Controller name:" followed by a text input field containing the text "CategoryController". At the bottom right of the dialog, there are two buttons: "Add" and "Cancel".

- *Bước 3:* Cài đặt CategoryController

CategoryController.cs

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

using EShopWebAPI.DTO;

namespace EShopWebAPI.Controllers
{
    public class CategoryController : ApiController
    {
        List<CategoryDTO> categories = new List<CategoryDTO>()
        {
            new CategoryDTO(1, "Laptop", "Máy xách tay"),
            new CategoryDTO(2, "Mobile", "Điện thoại"),
            new CategoryDTO(3, "TV", "Tivi")
        };

        // GET: api/Category
        public IEnumerable<CategoryDTO> Get()
        {
            return categories;
        }

        // GET: api/Category/5
        public CategoryDTO Get(int id)
        {
            return categories[id];
        }

        // POST: api/Category
        public List<CategoryDTO> Post([FromBody]CategoryDTO category)
        {
        }
```

```
        categories.Add(category);  
        return categories;  
    }  
  
    // PUT: api/Category/5  
    public List<CategoryDTO> Put(int id, [FromBody]CategoryDTO category)  
    {  
        categories[id] = category;  
        return categories;  
    }  
  
    // DELETE: api/Category/5  
    public List<CategoryDTO> Delete(int id)  
    {  
        categories.RemoveAt(id);  
        return categories;  
    }  
}
```

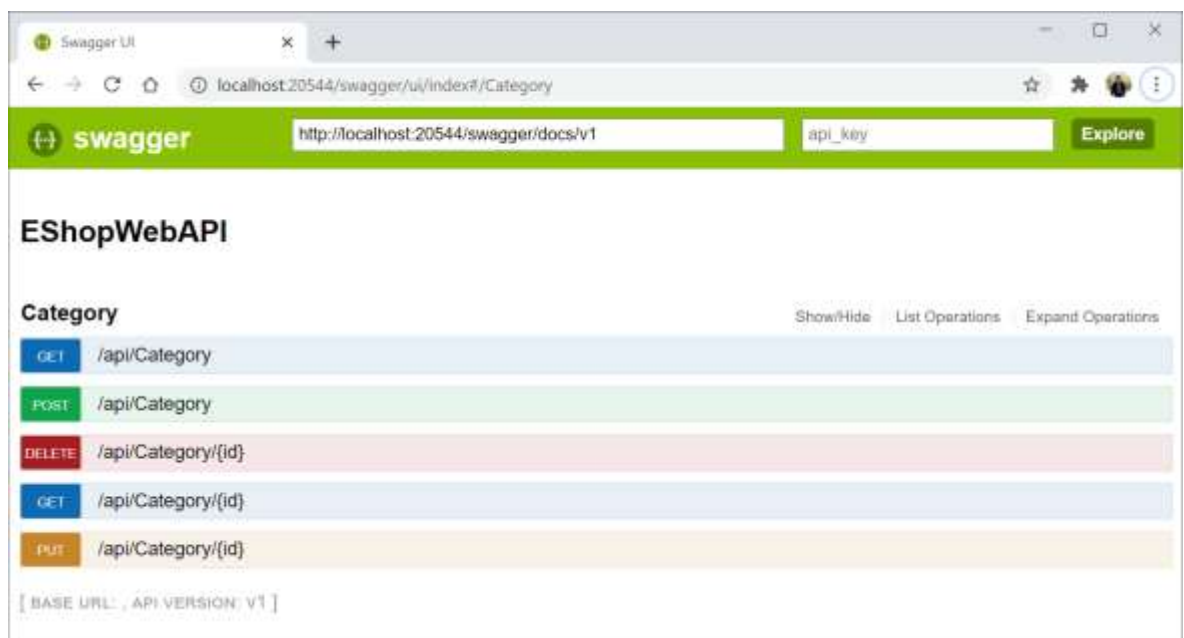
4. Kiểm tra Web API với các công cụ Swagger, Postman, Fiddler

Bạn cần chạy dự án Web API để có được URL tương tự sau:

<http://localhost:20544/>

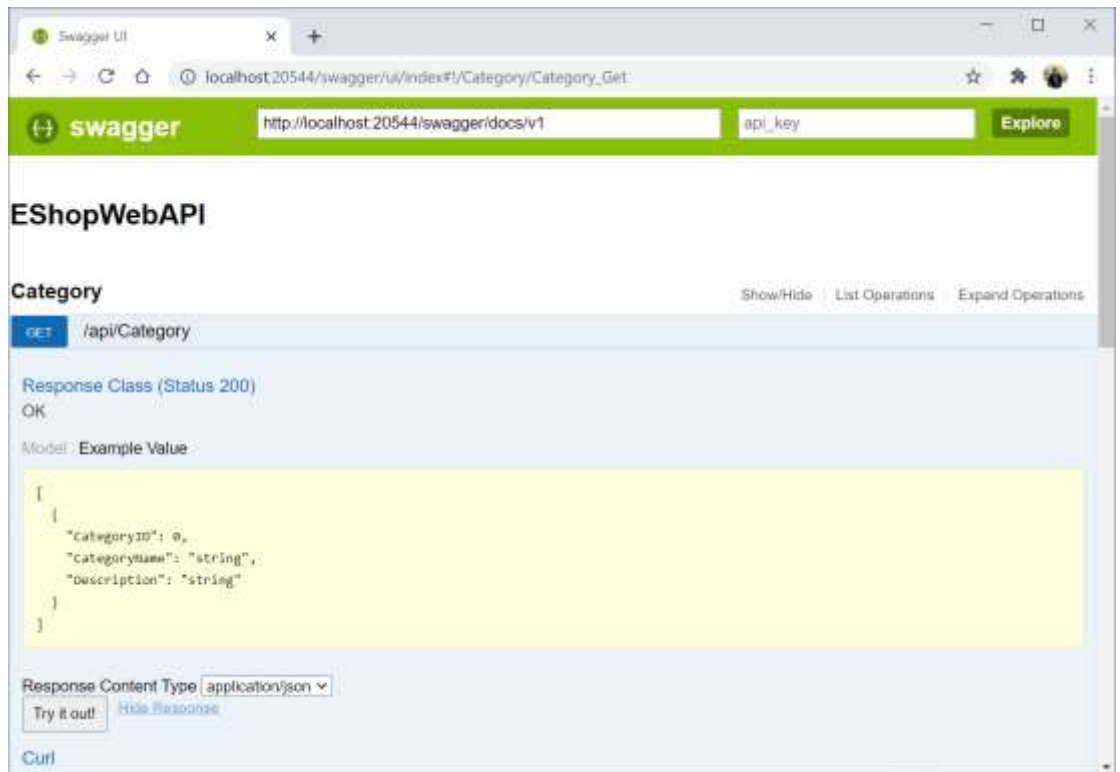
- Kiểm tra Web API với Swagger
 - *Bước 1:* Bạn cần đảm bảo Swagger đã được cài đặt vào dự án
 - *Bước 2:* Truy cập vào trang

<http://localhost:20544/swagger>

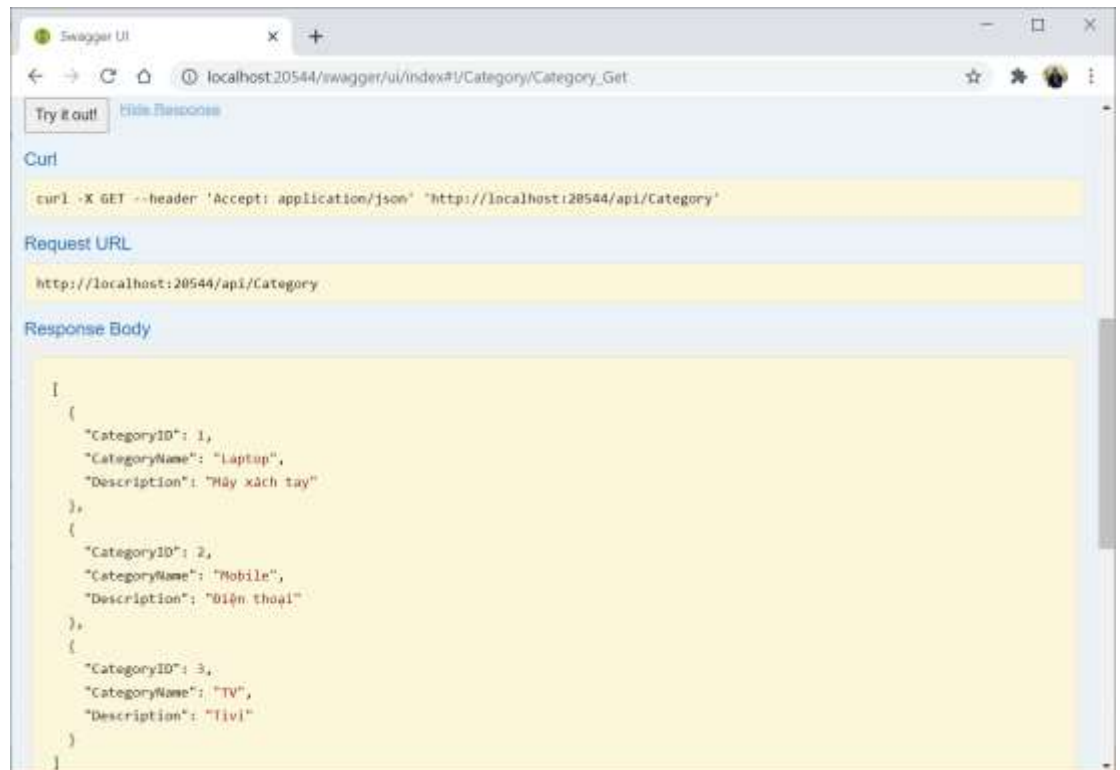


- *Bước 3:* Kiểm tra từng hàm API
 - Kiểm tra hàm GET: /api/Category

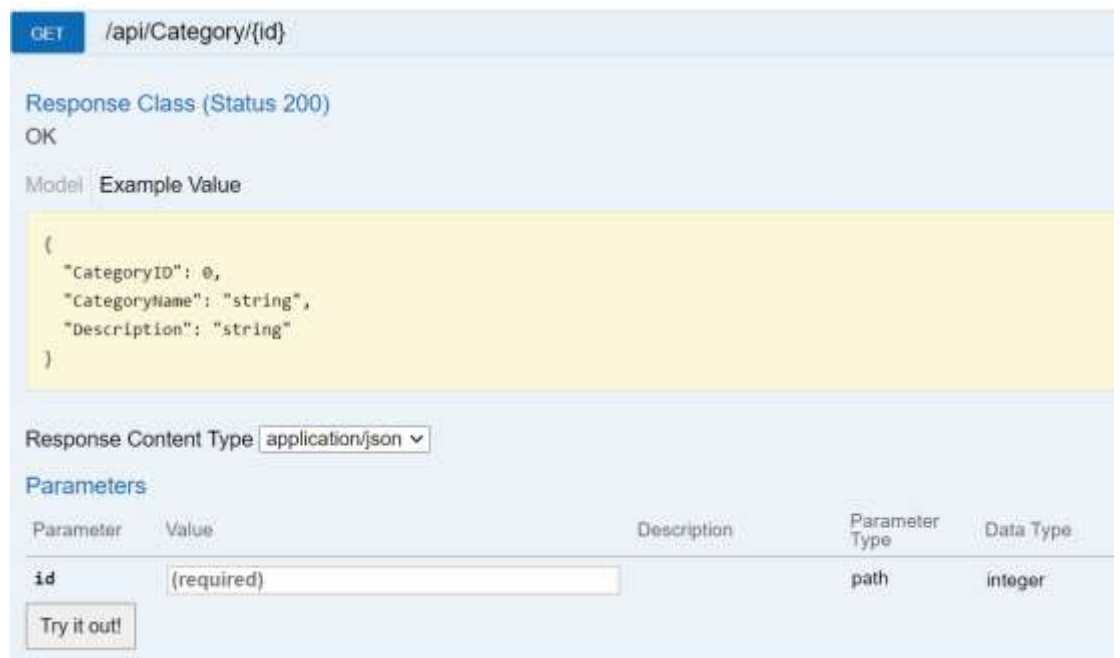
Click **Try it out**



Kết quả tương tự sau:



- Kiểm tra hàm GET: `/api/Category/{id}`



Nhập “1” vào ô **(required)** → Nhấn **Try it out!**

Kết quả tương tự hình sau:

Request URL

http://localhost:20544/api/Category/1

Response Body

```
{
  "CategoryID": 2,
  "CategoryName": "Mobile",
  "Description": "Điện thoại"
}
```

Response Code

200

- Kiểm tra hàm POST:

POST /api/Category

Response Class (Status 200)
OK

Model: Example Value

```
{
  "CategoryID": 8,
  "CategoryName": "string",
  "Description": "string"
}
```

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
category	(required)		body	Model: Example Value

Parameter content type: application/json

Try it out!

```
{
  "CategoryID": 8,
  "CategoryName": "string",
  "Description": "string"
}
```

Nhập vào ô (required) nội dung một loại sản phẩm dưới dạng JSON như sau:

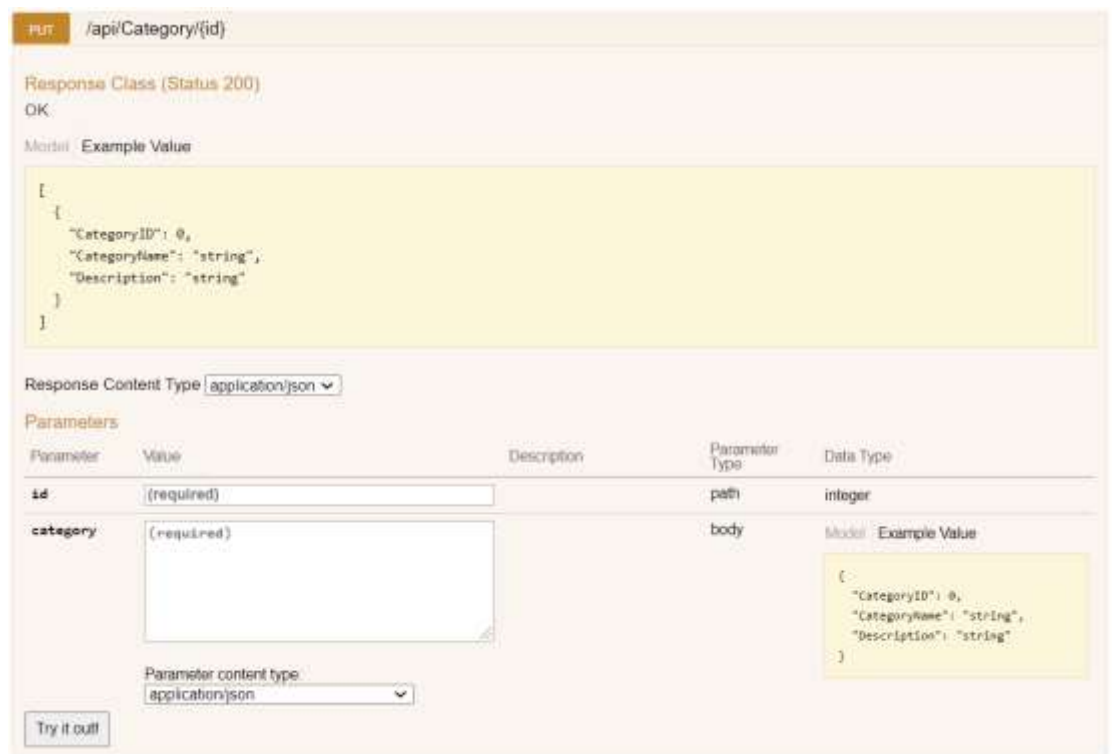
```
{
  "CategoryID": 4,
  "CategoryName": "Printer",
  "Description": "Máy in"
}
```

Nhấn **Try it out!**

Kết quả sẽ tương tự hình sau:



- Kiểm tra hàm PUT: /api/Category/{id}



Nhập **“1”** vào ô **id (required)** và đoạn mã chứa thông tin sản phẩm cần sửa vào ô **category (required)** như sau:

```
{  
  "CategoryID": 5,  
  "CategoryName": "Speaker",  
  "Description": "Loa"  
}
```



- Kiểm tra hàm DELETE: /api/Category/{id}

DELETE /api/Category/{id}

Response Class (Status 200)
OK

Model Example Value

```
{
  {
    "CategoryID": 0,
    "CategoryName": "string",
    "Description": "string"
  }
}
```

Response Content Type

Parameters:

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="(required)"/>		path	integer

Nhập “1” vào ô **id (required)**

Nhấn **Try it out!**

Kết quả tương tự sau:

Response Body

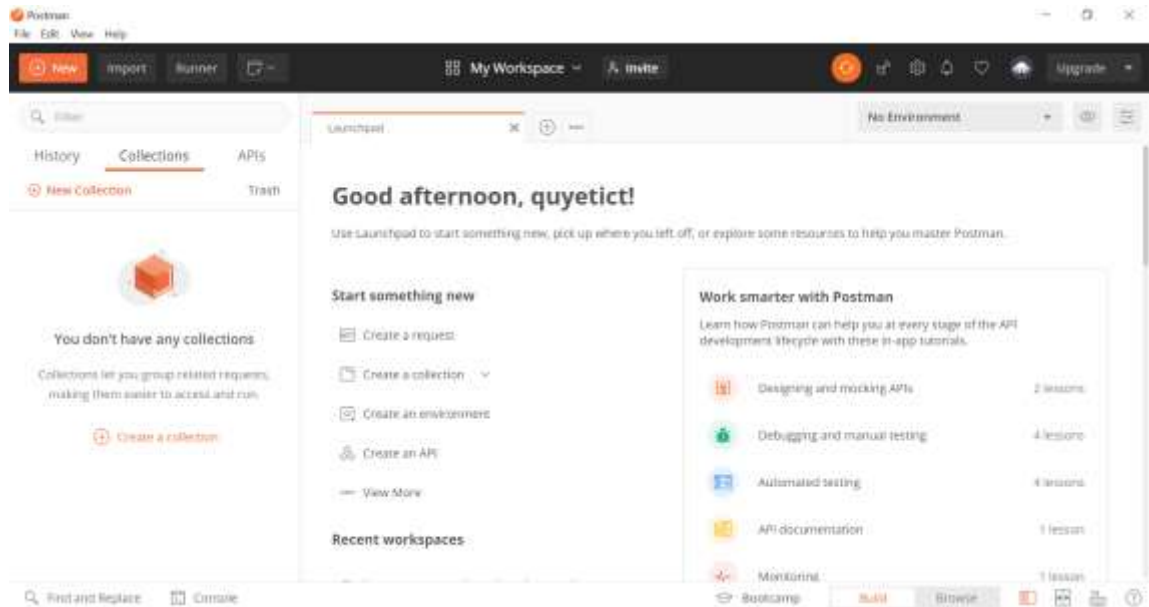
```
[
  {
    "CategoryID": 1,
    "CategoryName": "Laptop",
    "Description": "Máy xách tay"
  },
  {
    "CategoryID": 3,
    "CategoryName": "TV",
    "Description": "Tivi"
  }
]
```

Response Code

200

- Kiểm tra Web API với POSTMAN

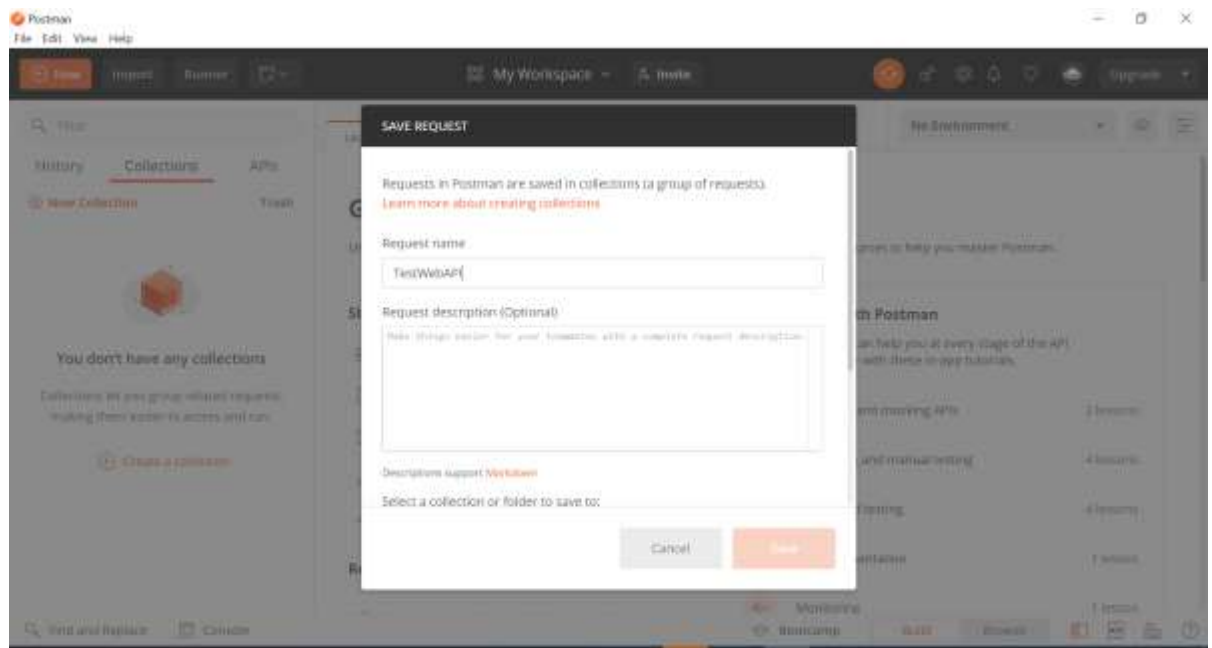
Bước 1: Khởi động Postman



Bước 2: Tạo mới dự án kiểm tra Web API

Chọn New → Request

Đặt tên yêu cầu: TestWebAPI



Tạo mới 1 Collection với tên: MyWebAPI

SAVE REQUEST

Make things easier for your teammates with a complete request description.

Descriptions support **Markdown**

Select a collection or folder to save to:

Search for a collection or folder

All Collections

+ Create Collection

MyWebAPI

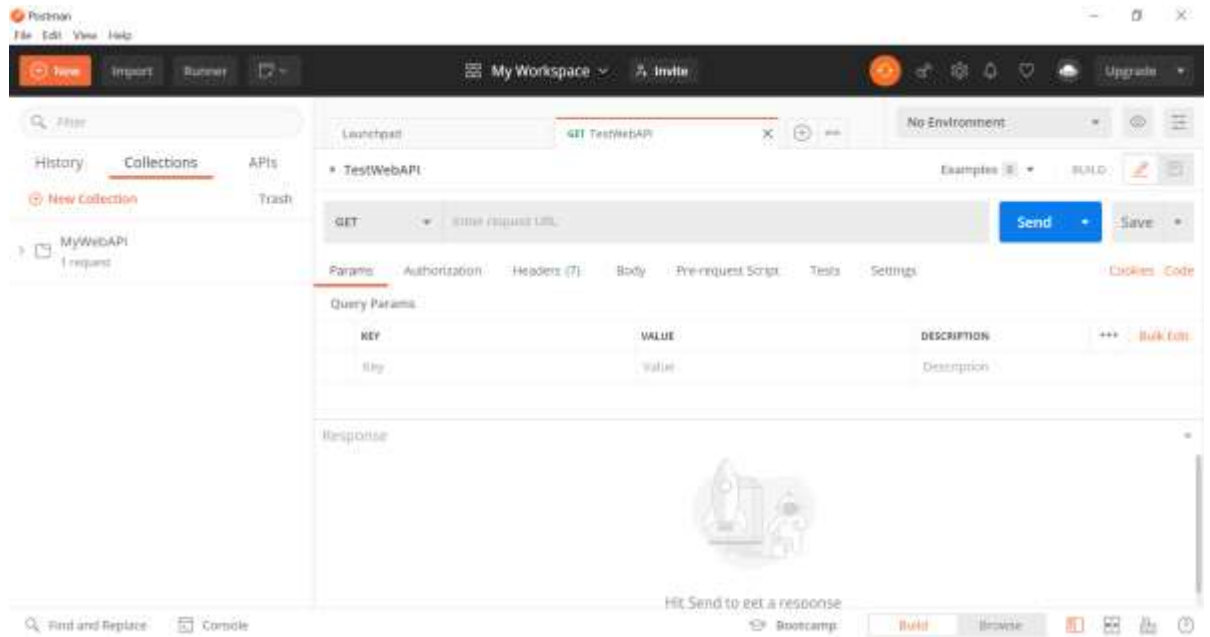
×

✓

Cancel

Save

Chọn **Save**, bạn sẽ thấy giao diện như dưới đây.



Sau đó thực hiện kiểm tra các phương thức Web API với các đường link và dữ liệu tương ứng (tương tự với Swagger).

5. Sinh viên làm tương tự theo hướng dẫn thực hiện yêu cầu 3

6. Sinh viên làm tương tự theo hướng dẫn thực hiện yêu cầu 4

BÀI THỰC HÀNH SỐ 2: XÂY DỰNG ỨNG DỤNG WEB API VỚI CRUD (1)

A. MỤC TIÊU BÀI THỰC HÀNH

Sau bài thực hành này sinh viên có thể:

- Tạo được mô hình dữ liệu cho ứng dụng Web API sử dụng ADO.NET Entity Framework
- Viết được các hàm GET, POST, PUT, DELETE thao tác với cơ sở dữ liệu SQL Server
- Sử dụng thành thạo các công cụ kiểm tra Web API

B. ĐIỀU KIỆN THỰC HÀNH

Trong bài thực hành này, sinh viên cần kiểm tra và chắc chắn các phần mềm sau trên máy tính còn hoạt động tốt:

1. Visual Studio 2017/2019
2. Postman

C. TÀI NGUYÊN THAM CHIẾU

Để hoàn thành tốt bài thực hành này sinh viên nên tham khảo các tài nguyên sau:

STT	Tên tài nguyên	Mô tả tài nguyên
1	ASP.NET Web API Tutorial for Beginners and Professional	Tài liệu tham khảo về ASP.NET Web API

D. YÊU CẦU BÀI TẬP THỰC HÀNH

Xây dựng Web API cho ứng dụng quản lý bán hàng Online với các yêu cầu sau:

1. Tạo cơ sở dữ liệu SQL Server có tên “**EShop**” với các bảng dữ liệu như sau:

Bảng **Categories**

Tên thuộc tính	Kiểu dữ liệu	Mô tả
CategoryID	int	Mã loại sản phẩm, khóa chính
CategoryName	nvarchar(100)	Tên loại sản phẩm

Description	nvarchar(255)	Chi chú
-------------	---------------	---------

Bảng **Products**

Tên thuộc tính	Kiểu dữ liệu	Mô tả
ProductID	int	Mã sản phẩm, khóa chính
CategoryID	int	Mã loại sản phẩm, khóa ngoại
ProductName	nvarchar(255)	Tên sản phẩm
UnitPrice	int	Đơn giá
Quantity	int	Số lượng

- Tạo mô hình dữ liệu có tên **EShopModel** trong thư mục **Models** của dự án **EShopWebAPI** (đã tạo trong bài thực hành trước).
- Tạo các phương thức GET trong các Controller tương ứng (đã tạo ở bài trước) thực hiện:
 - Lấy về toàn bộ loại sản phẩm
 - Lấy về một loại sản phẩm theo mã loại
 - Lấy về toàn bộ sản phẩm trong bảng **Products**
 - Lấy về của một sản phẩm theo mã sản phẩm
- Tạo các phương thức POST cho phép:
 - Thêm mới một loại sản phẩm
 - Thêm mới một sản phẩm
- Tạo các phương thức PUT cho phép
 - Sửa thông tin một loại sản phẩm
 - Sửa thông tin một sản phẩm
- Tạo các phương thức DELETE cho phép
 - Xóa thông tin một loại sản phẩm
 - Xóa thông tin một sản phẩm
- Kiểm tra các phương thức trên sử dụng công cụ POSTMAN

E. HƯỚNG DẪN THỰC HIỆN

1. Tạo cơ sở dữ liệu “EShop”

Bước 1: Sử dụng lệnh T-SQL để tạo cơ sở dữ liệu

```
CREATE DATABASE EShop
```

Bước 2: Bạn có thể tự thiết kế các bảng cơ sở dữ liệu theo mô tả ở trên hoặc sử dụng các lệnh T-SQL dưới đây để tạo bảng và dữ liệu minh họa:

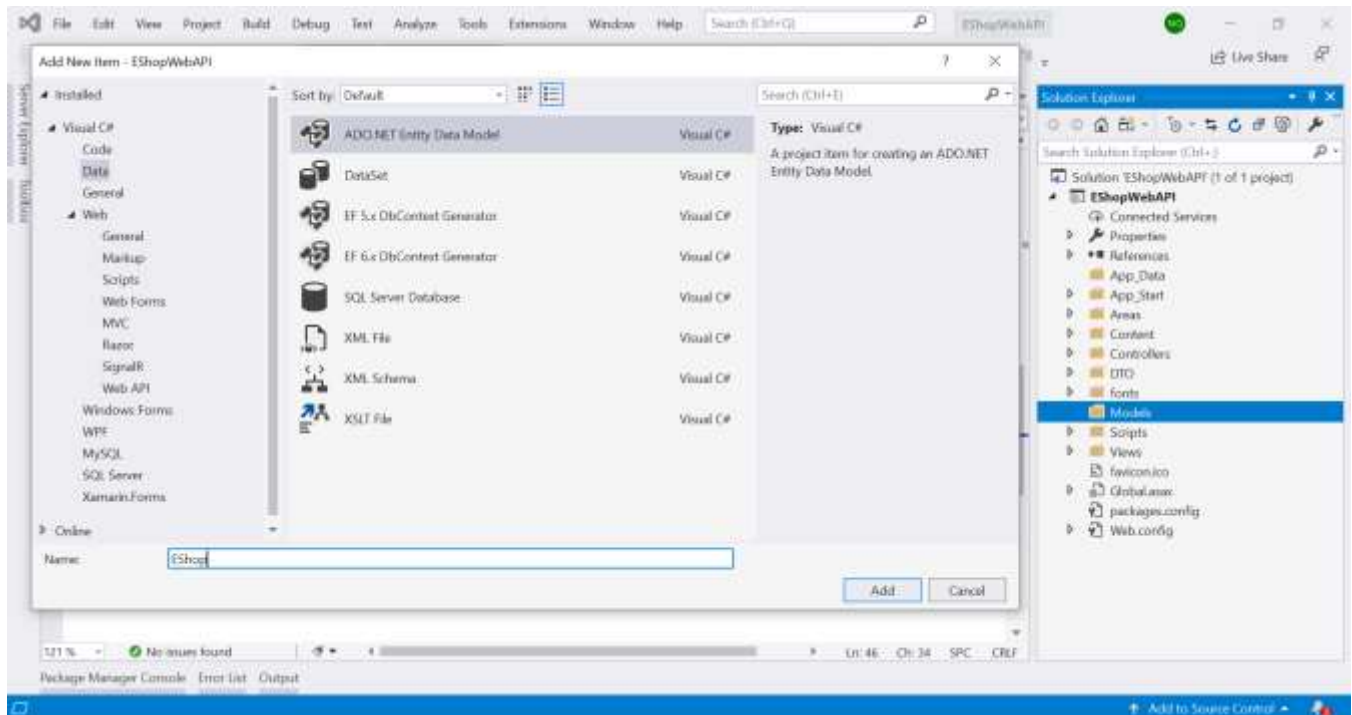
```
USE [EShop]
GO
/***** Object: Table [dbo].[Categories]    Script Date: 9/17/2020 10:54:03
AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Categories](
    [CategoryID] [int] NOT NULL,
    [CategoryName] [nvarchar](100) NOT NULL,
    [Description] [nvarchar](255) NULL,
    CONSTRAINT [PK_Categories] PRIMARY KEY CLUSTERED
(
    [CategoryID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Products]    Script Date: 9/17/2020 10:54:03 AM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Products](
    [ProductID] [int] NOT NULL,
    [CategoryID] [int] NOT NULL,
    [ProductName] [nvarchar](255) NOT NULL,
    [UnitPrice] [int] NOT NULL,
    [Quantity] [int] NOT NULL,
    CONSTRAINT [PK_Products] PRIMARY KEY CLUSTERED
(
    [ProductID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
INSERT [dbo].[Categories] ([CategoryID], [CategoryName], [Description]) VALUES
(1, N'Mobile', N'Điện thoại')
```

```
GO
INSERT [dbo].[Categories] ([CategoryID], [CategoryName], [Description]) VALUES
(2, N'Tivi', N'Ti vi')
GO
INSERT [dbo].[Categories] ([CategoryID], [CategoryName], [Description]) VALUES
(3, N'Computer', N'Máy tính')
GO
INSERT [dbo].[Products] ([ProductID], [CategoryID], [ProductName],
[UnitPrice], [Quantity]) VALUES (1, 1, N'Samsung Galaxy S8', 7500000, 50)
GO
INSERT [dbo].[Products] ([ProductID], [CategoryID], [ProductName],
[UnitPrice], [Quantity]) VALUES (2, 1, N'Samsung Galaxy S9', 1050000, 75)
GO
INSERT [dbo].[Products] ([ProductID], [CategoryID], [ProductName],
[UnitPrice], [Quantity]) VALUES (3, 1, N'iPhone 8', 18000000, 20)
GO
INSERT [dbo].[Products] ([ProductID], [CategoryID], [ProductName],
[UnitPrice], [Quantity]) VALUES (4, 1, N'Xiaomi Redmi 8', 4990000, 100)
GO
INSERT [dbo].[Products] ([ProductID], [CategoryID], [ProductName],
[UnitPrice], [Quantity]) VALUES (5, 1, N'Xiaomi Note 8', 5990000, 100)
GO
INSERT [dbo].[Products] ([ProductID], [CategoryID], [ProductName],
[UnitPrice], [Quantity]) VALUES (6, 2, N'LG SMART TV 4K 43INCH', 8490000, 15)
GO
INSERT [dbo].[Products] ([ProductID], [CategoryID], [ProductName],
[UnitPrice], [Quantity]) VALUES (7, 2, N'Smart TV Samsung 49inch', 7780000,
20)
GO
INSERT [dbo].[Products] ([ProductID], [CategoryID], [ProductName],
[UnitPrice], [Quantity]) VALUES (8, 2, N'Smart TV TCL 55inch', 5950000, 30)
GO
INSERT [dbo].[Products] ([ProductID], [CategoryID], [ProductName],
[UnitPrice], [Quantity]) VALUES (9, 3, N'Acer TravelMate X', 25800000, 16)
GO
INSERT [dbo].[Products] ([ProductID], [CategoryID], [ProductName],
[UnitPrice], [Quantity]) VALUES (10, 3, N'MacBook Air 2020', 28000000, 12)
GO
ALTER TABLE [dbo].[Products] WITH CHECK ADD CONSTRAINT
[FK_Products_Categories] FOREIGN KEY([CategoryID])
REFERENCES [dbo].[Categories] ([CategoryID])
GO
ALTER TABLE [dbo].[Products] CHECK CONSTRAINT [FK_Products_Categories]
GO
```

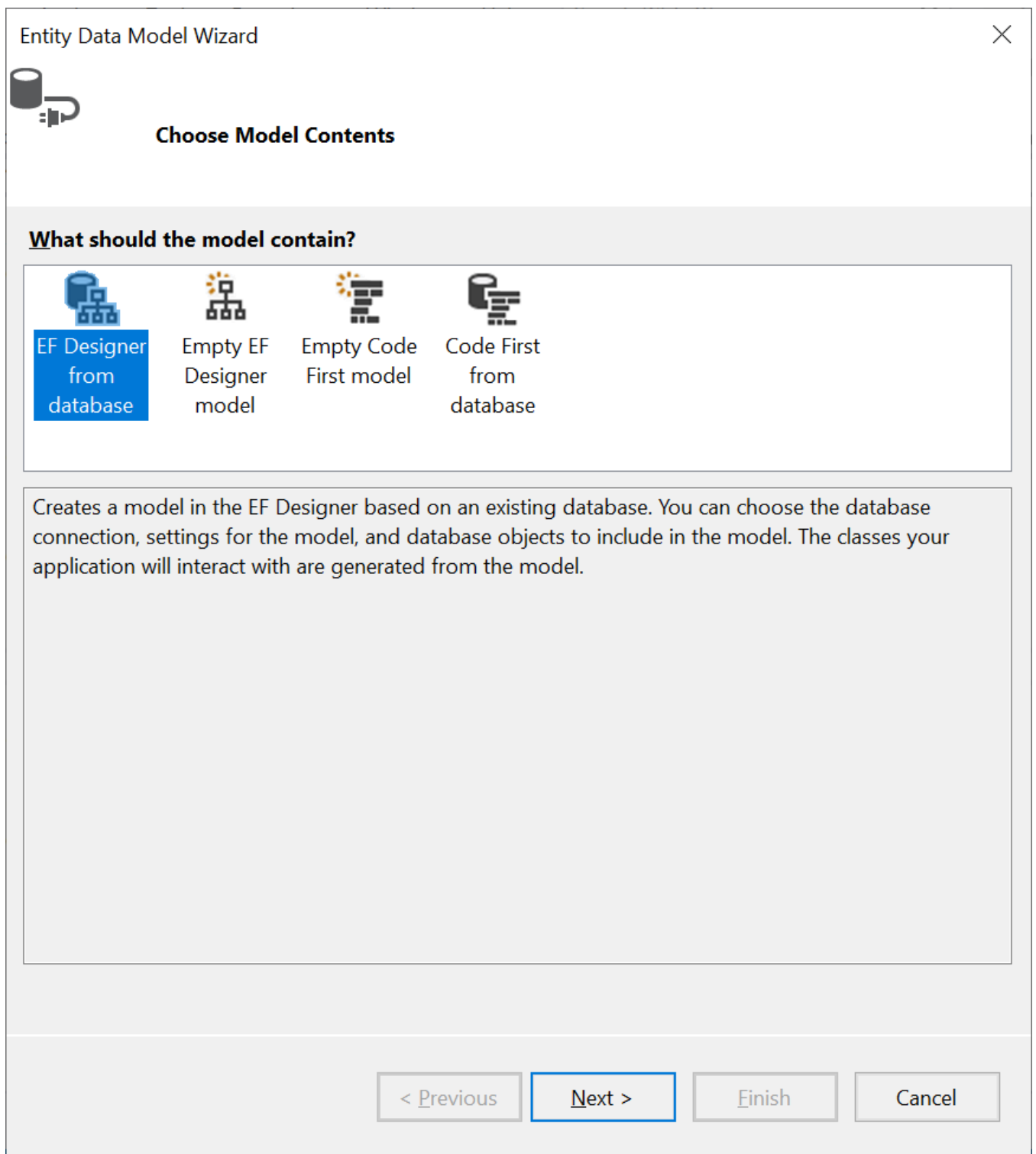
2. Tạo mô hình dữ liệu “EShopModel”

Bước 1: Chuột phải vào thư mục **Models** → **Add New Item** → **Data** → **ADO.NET Entity Data Model**.

Bước 2: Đặt tên cho model là **EShop** → **Add**



Bước 3: Chọn **EF Designer from database** → **Next**



Bước 4: Chọn New Connection

Entity Data Model Wizard



Choose Your Data Connection

Which data connection should your application use to connect to the database?

New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

☒ Save connection settings in Web.Config as:

< Previous

Next >

Finish

Cancel

Giao diện kết nối tới CSDL hiện ra, tại ô **Server Name** nhập: **.\SQLEXPRESS** hoặc **(local)**; tại mục **Select or enter a database name** nhập: **EShop** → **OK**

26

Connection Properties ? X

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:

Microsoft SQL Server (SqlClient) Change...

Server name:

.\SQLEXPRESS Refresh

Log on to the server

Authentication: Windows Authentication

User name: Password:

☐ Save my password

Connect to a database

☒ Select or enter a database name:

EShop

☐ Attach a database file:

Browse...


Logical name:

Advanced...

Test Connection OK Cancel

Bước 5: Giao diện hiển thị thông tin kết nối đến cơ sở dữ liệu hiển thị ra, chọn **Next** → **Next**

Entity Data Model Wizard



Choose Your Data Connection

Which data connection should your application use to connect to the database?

quyetnv-laptop\sqlexpress.EShop.dbo

New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

```

metadata=res:/**/Models.EShop.csdl|res:/**/Models.EShop.ssdl|
res:/**/Models.EShop.msl;provider=System.Data.SqlClient;provider connection string="data
source=.SQLEXPRESS;initial catalog=EShop;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework"

```

☒ Save connection settings in Web.Config as:

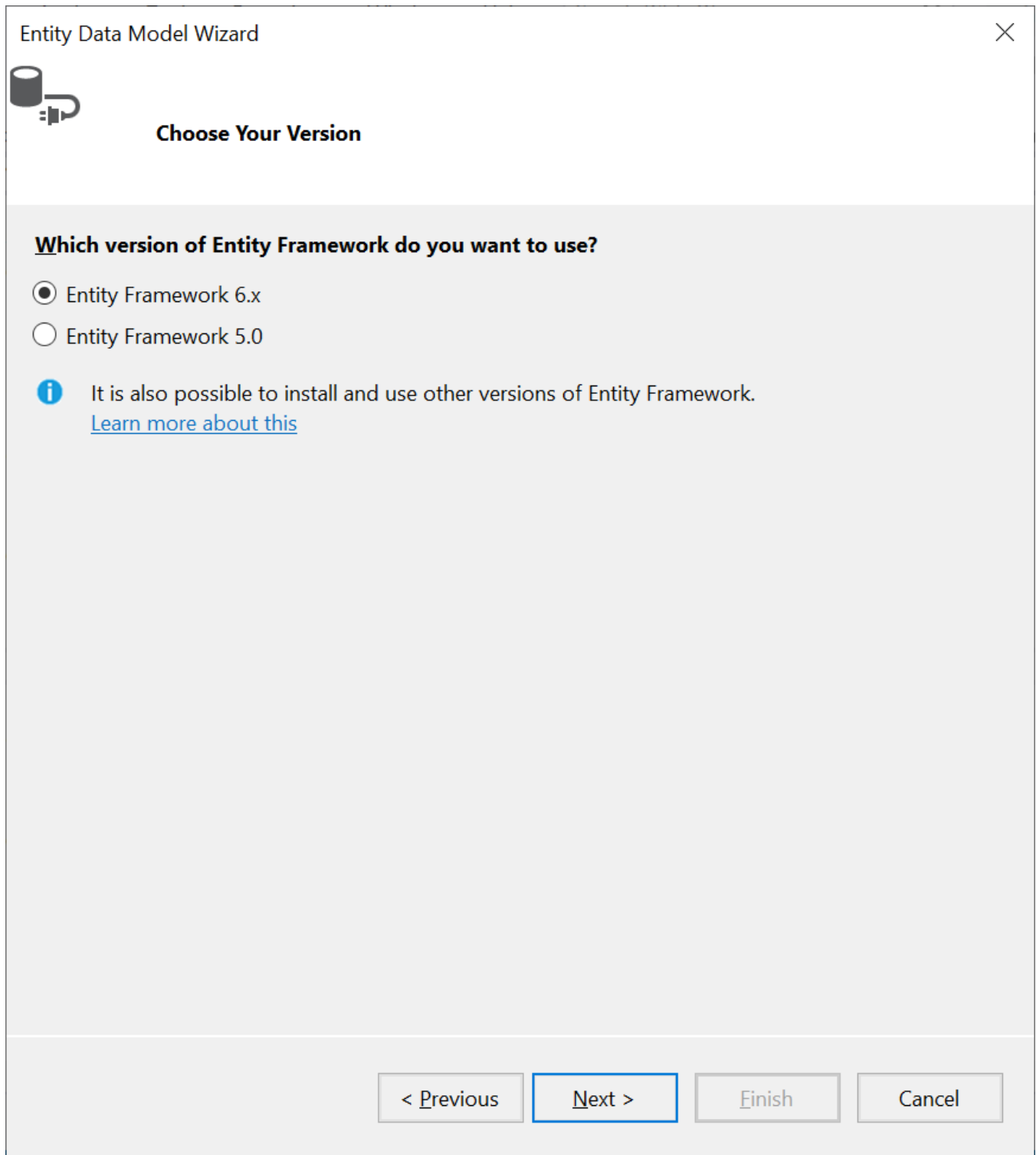
EShopEntities

< Previous

Next >

Finish


Cancel



The image shows a screenshot of the 'Entity Data Model Wizard' dialog box. The title bar at the top reads 'Entity Data Model Wizard' with a close button (X) on the right. Below the title bar is a header area with a database icon and the text 'Choose Your Version'. The main content area has a question: 'Which version of Entity Framework do you want to use?'. There are two radio button options: 'Entity Framework 6.x' (which is selected) and 'Entity Framework 5.0'. Below these options is an information icon (i) followed by the text: 'It is also possible to install and use other versions of Entity Framework. [Learn more about this](#)'. At the bottom of the dialog, there are four buttons: '< Previous', 'Next >' (which is highlighted with a blue border), 'Finish', and 'Cancel'.


Bước 6: Giao diện chọn các thực thể hiển thị ra, chọn **Tables**, sau đó chọn 2 bảng **Products** và **Categories**. Cuối cùng nhấn **Finish**.


Entity Data Model Wizard





Choose Your Database Objects and Settings


Which database objects do you want to include in your model?


☒  Tables


☒  dbo

☒  Categories

☒  Products

☐  sysdiagrams

☐  Views

☐  Stored Procedures and Functions

☒ Pluralize or singularize generated object names

☒ Include foreign key columns in the model

☒ Import selected stored procedures and functions into the entity model

Model Namespace:

EShopModel

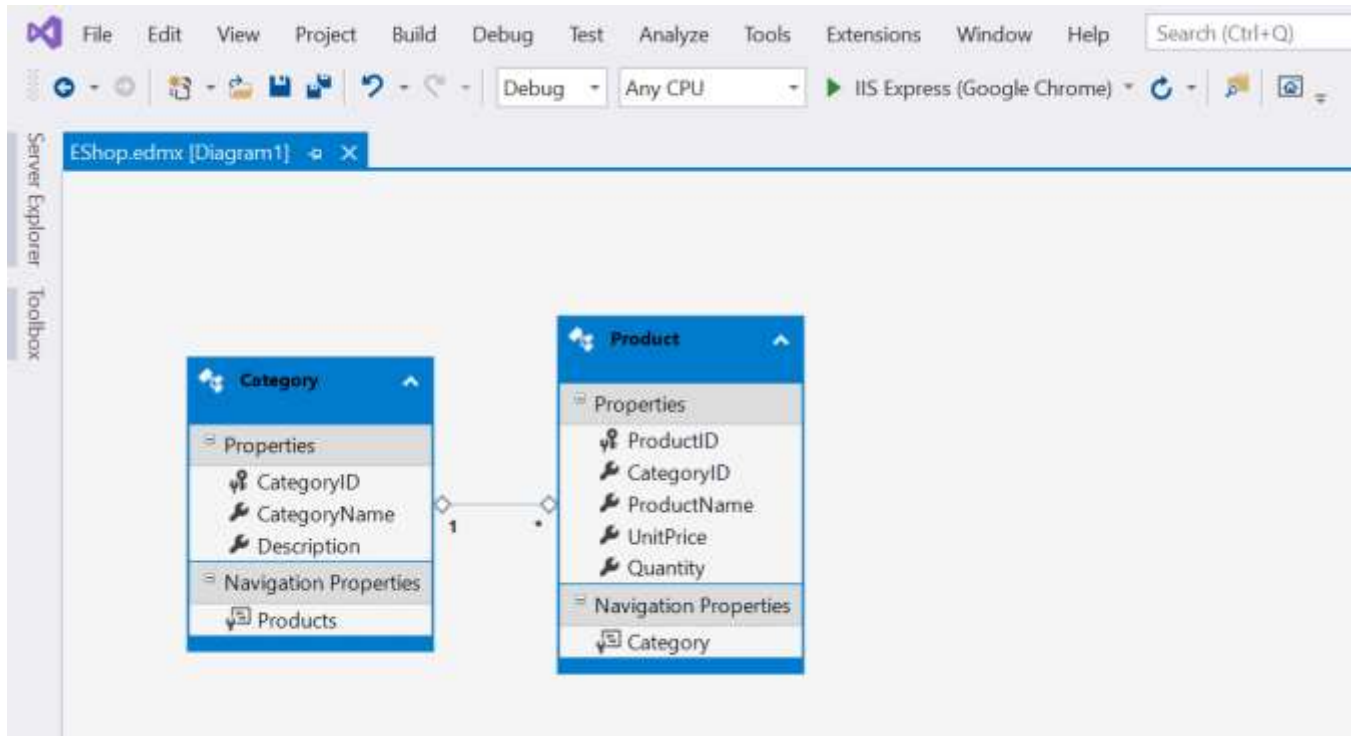
< Previous

Next >

Finish

Cancel

Kết quả hiển thị ra như hình dưới đây:



3. Sinh viên tự làm theo bài tập mẫu (trên lớp)
 - Cần **using EShopWebAPI.Models** trong các Controller
 - Cần sử dụng các thuộc tính **JsonIgnore** và **XmlIgnore**
4. Sinh viên tự làm theo bài tập mẫu (trên lớp)
5. Sinh viên tự làm theo bài tập mẫu (trên lớp)
6. Sinh viên tự làm theo bài tập mẫu (trên lớp)
7. Sinh viên tự làm theo bài tập mẫu (trên lớp)

Lưu ý: Các yêu cầu 3, 4, 5, 6 nên viết trong các Controller tương ứng là **CategoryController** và **ProductController**.

BÀI THỰC HÀNH SỐ 3: XÂY DỰNG ỨNG DỤNG WEB API VỚI CRUD (2)

A. MỤC TIÊU BÀI THỰC HÀNH

Sau bài thực hành này sinh viên có thể:

- Tạo được mô hình dữ liệu cho ứng dụng Web API sử dụng ADO.NET Entity Framework
- Viết được các hàm tùy biến sử dụng các thuộc tính HttpGet, HttpPost, HttpPut, HttpDelete để thao tác với cơ sở dữ liệu SQL Server
- Sử dụng thành thạo các công cụ kiểm tra Web API

B. ĐIỀU KIỆN THỰC HÀNH

Trong bài thực hành này, sinh viên cần kiểm tra và chắc chắn các phần mềm sau trên máy tính còn hoạt động tốt:

3. Visual Studio 2017/2019
4. Postman

C. TÀI NGUYÊN THAM CHIẾU

Để hoàn thành tốt bài thực hành này sinh viên nên tham khảo các tài nguyên sau:

STT	Tên tài nguyên	Mô tả tài nguyên
1	ASP.NET Web API Tutorial for Beginners and Professional	Tài liệu tham khảo về ASP.NET Web API

D. YÊU CẦU BÀI TẬP THỰC HÀNH

Tiếp tục xây dựng Web API cho ứng dụng quản lý bán hàng Online với các yêu cầu sau:

1. Tạo các bảng cơ sở dữ liệu sau vào CSDL “EShop” (đã tạo ở bài thực hành trước):

Bảng Customers

Tên thuộc tính	Kiểu dữ liệu	Mô tả
<u>CustomerID</u>	int	Mã khách hàng, khóa chính

Fullname	nvarchar(100)	Họ tên khách hàng
Gender	bit	Giới tính
Birthday	DateTime	Ngày sinh
Address	nvarchar(255)	Quê quán
Email	varchar(100)	Email
PhoneNumber	char(11)	Số điện thoại

Bảng **Orders**

Tên thuộc tính	Kiểu dữ liệu	Mô tả
<u>OrderID</u>	int	Mã hóa đơn, khóa chính
<u>CustomerID</u>	int	Mã khách hàng, khóa ngoại
OrderDate	DateTime	Thời gian đặt
ShipAddress	nvarchar(255)	Đơn giá

Bảng **OrderDetails**

Tên thuộc tính	Kiểu dữ liệu	Mô tả
<u>OrderID</u>	int	Mã hóa đơn
<u>ProductID</u>	int	Mã sản phẩm
Quantity	int	Số lượng
Discount	int	Chiết khấu

- Cập nhật mô hình dữ liệu có tên **EShopModel** trong thư mục **Models** của dự án **EShopWebAPI** (đã tạo trong bài thực hành ở những trước).
- Tạo **CustomerController** với các phương thức sau:
 - AddCustomer** cho phép thêm mới một khách hàng
 - GetAllCustomers** lấy về toàn bộ danh sách khách hàng

- c) **GetCustomerByID** lấy về một khách hàng theo mã
 - d) **GetCustomerByAddress** lấy về danh sách khách hàng theo địa chỉ truyền vào
4. Tạo **OrderController** với các phương thức sau:
- a) **AddOrder** cho phép thêm mới một hóa đơn với các sản phẩm thuộc hóa đơn đó
 - b) **GetAllOrders** cho phép lấy về toàn bộ hóa đơn (không bao gồm thông tin chi tiết)
 - c) **GetOrderByID** cho phép lấy thông tin chi tiết một hóa đơn
 - d) **EditOrder** cho phép sửa thông tin một hóa đơn
 - e) **DeleteOrder** cho phép xóa một hóa đơn theo mã
5. Kiểm tra các phương thức trên sử dụng công cụ POSTMAN

E. HƯỚNG DẪN THỰC HIỆN

1. Tạo các bảng cơ sở dữ liệu theo yêu cầu

Sử dụng các lệnh T-SQL để tạo bảng và thêm dữ liệu vào CSDL

```
USE [EShop]
GO
/***** Object: Table [dbo].[Customers]    Script Date: 9/26/2020 6:46:04 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Customers](
    [CustomerID] [int] NOT NULL,
    [Fullname] [nvarchar](100) NULL,
    [Gender] [bit] NULL,
    [BirthDay] [datetime] NULL,
    [Address] [nvarchar](255) NULL,
    [Email] [varchar](100) NULL,
    [PhoneNumber] [char](11) NULL,
    CONSTRAINT [PK_Customers] PRIMARY KEY CLUSTERED
(
    [CustomerID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[OrderDetails]    Script Date: 9/26/2020 6:46:04 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[OrderDetails](
    [OrderID] [int] NOT NULL,
    [ProductID] [int] NOT NULL,
    [Quantity] [int] NULL,
    [Discount] [int] NULL,
```

```

CONSTRAINT [PK_OrderDetails_1] PRIMARY KEY CLUSTERED
(
    [OrderID] ASC,
    [ProductID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Orders]    Script Date: 9/26/2020 6:46:04 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Orders](
    [OrderID] [int] NOT NULL,
    [CustomerID] [int] NULL,
    [OrderDate] [datetime] NULL,
    [ShipAddress] [nvarchar](255) NULL,
    CONSTRAINT [PK_Orders] PRIMARY KEY CLUSTERED
(
    [OrderID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
INSERT [dbo].[Customers] ([CustomerID], [Fullname], [Gender], [Birthday], [Address],
[Email], [PhoneNumber]) VALUES (1, N'Nguyễn Văn Quyết', 1, CAST(N'2000-09-23T00:00:00.000'
AS DateTime), N'Hà Nội', N'quyetict@gmail.com', N'0912188636 ')
GO
INSERT [dbo].[Customers] ([CustomerID], [Fullname], [Gender], [Birthday], [Address],
[Email], [PhoneNumber]) VALUES (2, N'Nguyễn Thị Huyền', 0, CAST(N'2001-09-24T00:00:00.000'
AS DateTime), N'Hưng Yên', N'nguyenhuyen@gmail.com', N'0961348219 ')
GO
INSERT [dbo].[OrderDetails] ([OrderID], [ProductID], [Quantity], [Discount]) VALUES (1, 1,
2, 20)
GO
INSERT [dbo].[OrderDetails] ([OrderID], [ProductID], [Quantity], [Discount]) VALUES (1, 2,
5, 10)
GO
INSERT [dbo].[OrderDetails] ([OrderID], [ProductID], [Quantity], [Discount]) VALUES (1, 3,
6, 10)
GO
INSERT [dbo].[OrderDetails] ([OrderID], [ProductID], [Quantity], [Discount]) VALUES (1, 4,
2, 15)
GO
INSERT [dbo].[OrderDetails] ([OrderID], [ProductID], [Quantity], [Discount]) VALUES (2, 3,
6, 10)
GO
INSERT [dbo].[OrderDetails] ([OrderID], [ProductID], [Quantity], [Discount]) VALUES (2, 4,
2, 15)
GO
INSERT [dbo].[OrderDetails] ([OrderID], [ProductID], [Quantity], [Discount]) VALUES (3, 1,
2, 10)
GO
INSERT [dbo].[Orders] ([OrderID], [CustomerID], [OrderDate], [ShipAddress]) VALUES (1, 1,
CAST(N'2020-09-24T00:00:00.000' AS DateTime), N'Khoa CNTT, Trường ĐHSPKT Hưng Yên')
GO

```

```
INSERT [dbo].[Orders] ([OrderID], [CustomerID], [OrderDate], [ShipAddress]) VALUES (2, 1,
CAST(N'2020-09-24T00:00:00.000' AS DateTime), N'71 Đỗ Chính, Văn Nhuế, Bần Yên Nhân, Mỹ
Hào, Hưng Yên')
GO
INSERT [dbo].[Orders] ([OrderID], [CustomerID], [OrderDate], [ShipAddress]) VALUES (3, 1,
CAST(N'2020-09-24T00:00:00.000' AS DateTime), N'155A, Vũ Văn Cẩn, Bần Yên Nhân, Mỹ Hào,
Hưng Yên')
GO
ALTER TABLE [dbo].[OrderDetails] WITH CHECK ADD CONSTRAINT [fk_Order_Details] FOREIGN
KEY([OrderID])
REFERENCES [dbo].[Orders] ([OrderID])
GO
ALTER TABLE [dbo].[OrderDetails] CHECK CONSTRAINT [fk_Order_Details]
GO
ALTER TABLE [dbo].[OrderDetails] WITH CHECK ADD CONSTRAINT [fk_Order_Product] FOREIGN
KEY([ProductID])
REFERENCES [dbo].[Products] ([ProductID])
GO
ALTER TABLE [dbo].[OrderDetails] CHECK CONSTRAINT [fk_Order_Product]
GO
ALTER TABLE [dbo].[Orders] WITH CHECK ADD CONSTRAINT [fk_OrderCustomer] FOREIGN
KEY([CustomerID])
REFERENCES [dbo].[Customers] ([CustomerID])
GO
ALTER TABLE [dbo].[Orders] CHECK CONSTRAINT [fk_OrderCustomer]
GO
```

2. Cập nhật mô hình dữ liệu trong EshopModel

- *Bước 1:* Double-Click vào file **EShop.edmx** → Giao diện Diagram hiện lên
- *Bước 2:* Right-Click vào vùng trắng trên Diagram → Chọn **Update Model from Database**
- *Bước 3:* Giao diện Update Wizard hiện ra chọn Table và các bảng tương ứng
- *Bước 4:* Nhấn **Finish**

3. Tạo CustomerController

Làm tương tự bài thực hành số 2

Lưu ý việc bổ sung các thuộc tính `HttpGet`, `HttpPost`, `HttpPut`, `HttpDelete` và chỉnh sửa file **WebApiConfig.cs**

```
routeTemplate: "api/{controller}/{action}/{id}"
```

4. Tạo OrderController

a) *Cách 1:* Tạo phương thức **AddOrder**(`int` orderID, `int` customerID, `string` orderDate, `string` shipAddress, `[FromBody]` List<OrderDetail> details)

Cách 2: Tạo phương thức **AddOrder**(`[FromBody]` OrderDTO)

Với *Cách 2*, OrderDTO chứa 1 thuộc tính **Order** và 1 danh sách **OrderDetails**
Khi đó cần lưu ý cấu trúc viết file JSON khi Test với POSTMAN

Các phần còn lại của yêu cầu 4 sinh viên làm tương tự bài tập mẫu

5. Sinh viên tự làm theo bài tập hướng dẫn trên lớp

BÀI THỰC HÀNH SỐ 4: PHÁT TRIỂN ỨNG DỤNG WEB SỬ DỤNG WEB API (1)

A. MỤC TIÊU BÀI THỰC HÀNH

Sau bài thực hành này sinh viên có thể:

- Trình bày được cách kết nối (truy cập) một ứng dụng Web API trong ASP.NET MVC
- Thiết kế được các trang Web hiển thị danh sách các bản ghi và tạo mới một bản ghi
- Sử dụng thành thạo các hàm và các thuộc tính của đối tượng HttpClient.

B. ĐIỀU KIỆN THỰC HÀNH

Trong bài thực hành này, sinh viên cần kiểm tra và chắc chắn các phần mềm sau trên máy tính còn hoạt động tốt:

1. Visual Studio 2017/2019
2. Postman

C. TÀI NGUYÊN THAM CHIẾU

Để hoàn thành tốt bài thực hành này sinh viên nên tham khảo các tài nguyên sau:

STT	Tên tài nguyên	Mô tả tài nguyên
1	ASP.NET Web API Tutorial for Beginners and Professional	Tài liệu tham khảo về ASP.NET Web API

D. YÊU CẦU BÀI TẬP THỰC HÀNH

Xây dựng Website quản lý bán hàng Online sử dụng WebAPI (EshopWebAPI) với các yêu cầu sau:

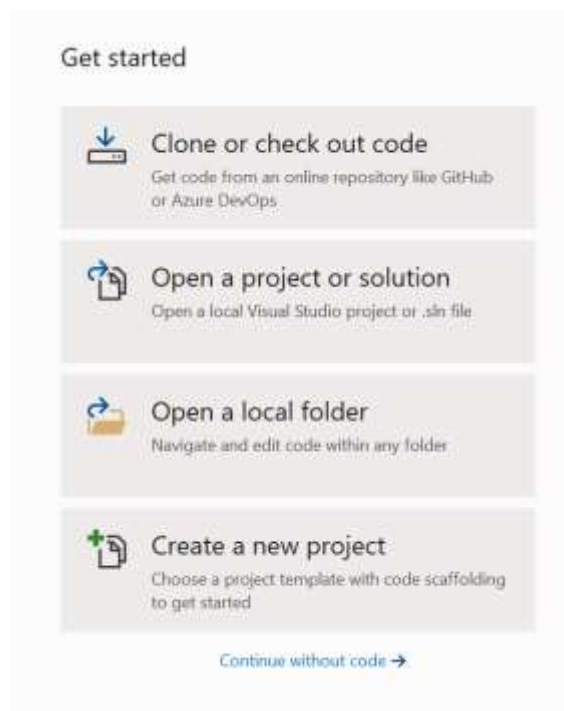
1. Tạo ứng dụng ASP.NET Web MVC có tên **EShopWebMVC**
2. Tạo lớp **Product** trong thư mục **Model** của dự án với các thuộc tính tương tự bài thực hành số 1

3. Tạo tệp **ProductController** trong thư mục Controller của dự án với các phương thức (Action) sau:
 - **Index**: cho phép lấy về toàn bộ danh sách các sản phẩm sử dụng hàm **Get** trong Web API
 - **Create**: cho phép thêm mới một sản phẩm sử dụng hàm **Post/AddProduct** trong Web API
 - **Details**: cho phép hiển thị chi tiết thông tin một sản phẩm theo mã sản phẩm
4. Tạo các View cho để hiển thị/nhập dữ liệu tương ứng với các Action ở yêu cầu 3.

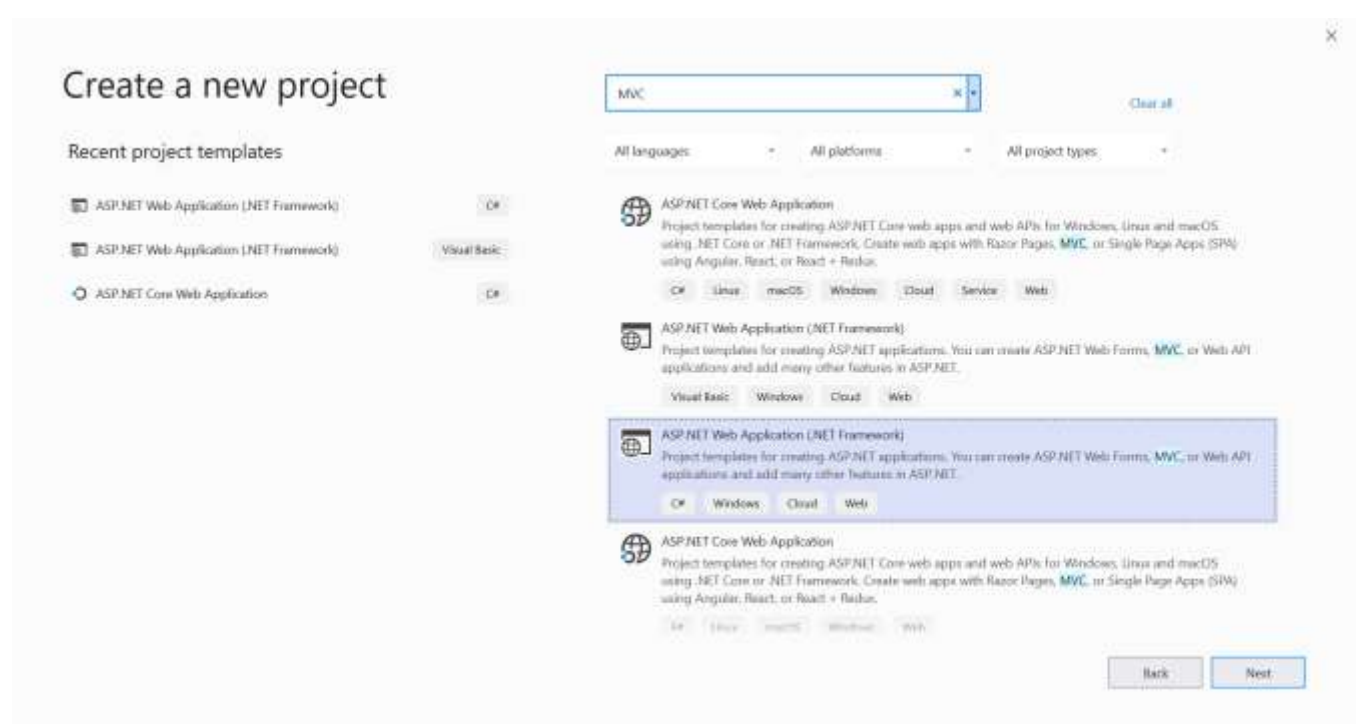
E. HƯỚNG DẪN THỰC HIỆN

1. Tạo ứng dụng EShopWebMVC

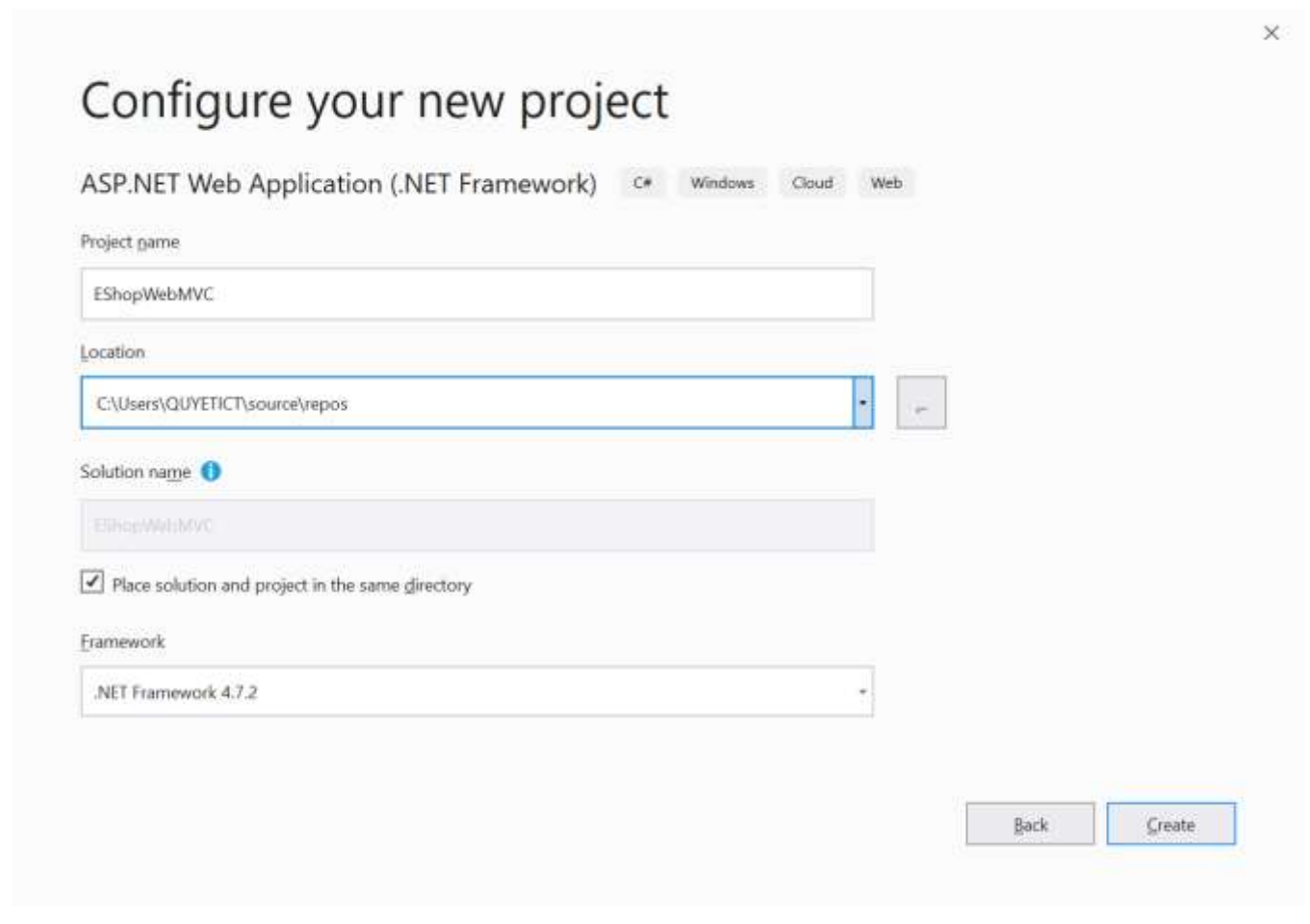
Bước 1: Từ giao diện Get Start của Visual Studio nhân Create a New Project



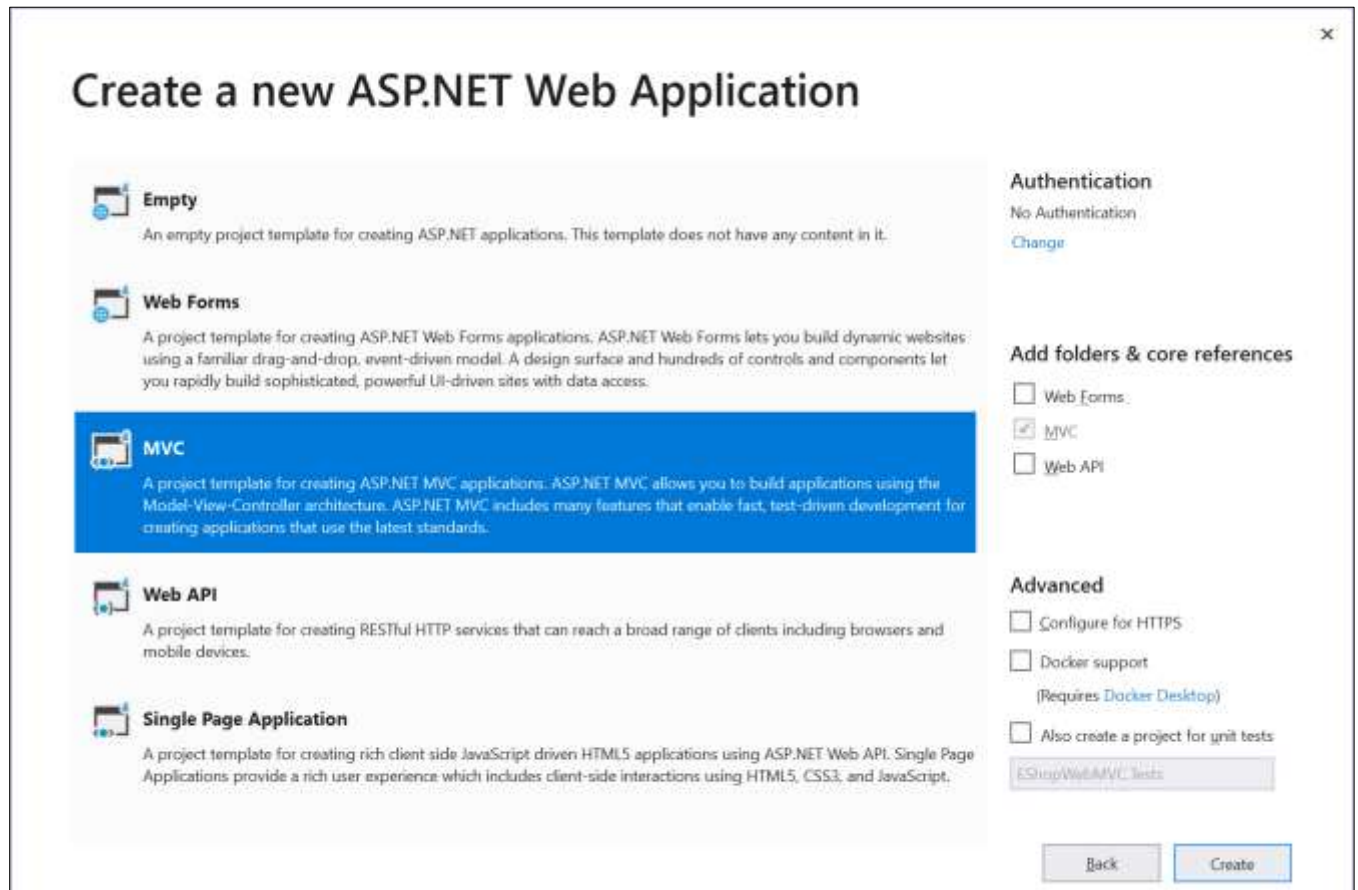
Bước 2: Từ giao diện tạo dự án, gõ “MVC” và tìm đến kiểu Template như trong hình:



Bước 3: Đặt tên dự án EShopWebMVC → Create



Bước 5: Chọn MVC → Create



Các yêu cầu 2, 3, 4 sinh viên làm tương tự ví dụ mẫu trên lớp