Introduction to ASP.NET Core MVC Framework

.

What is MVC?

- The MVC stands for Model View and Controller. It is an architectural software design pattern which is basically used to develop interactive applications. An interactive application is an application where there is user interaction involved and based on the user interaction some event handling occurred.
- The most important point that you need to remember is, it is not only used for developing web-based applications but also we can use this MVC design pattern to develop the Desktop or mobile-based application.
- The MVC (Model-View-Controller) design pattern was introduced in the 1970s which divides an application into 3 major components. They are Model, View, and Controller. The main objective of the MVC design pattern is the separation of concerns. It means the **domain model** and **business logic** are separated from the **user interface (i.e. view)**. As a result, maintaining and testing the application becomes simpler and easier.

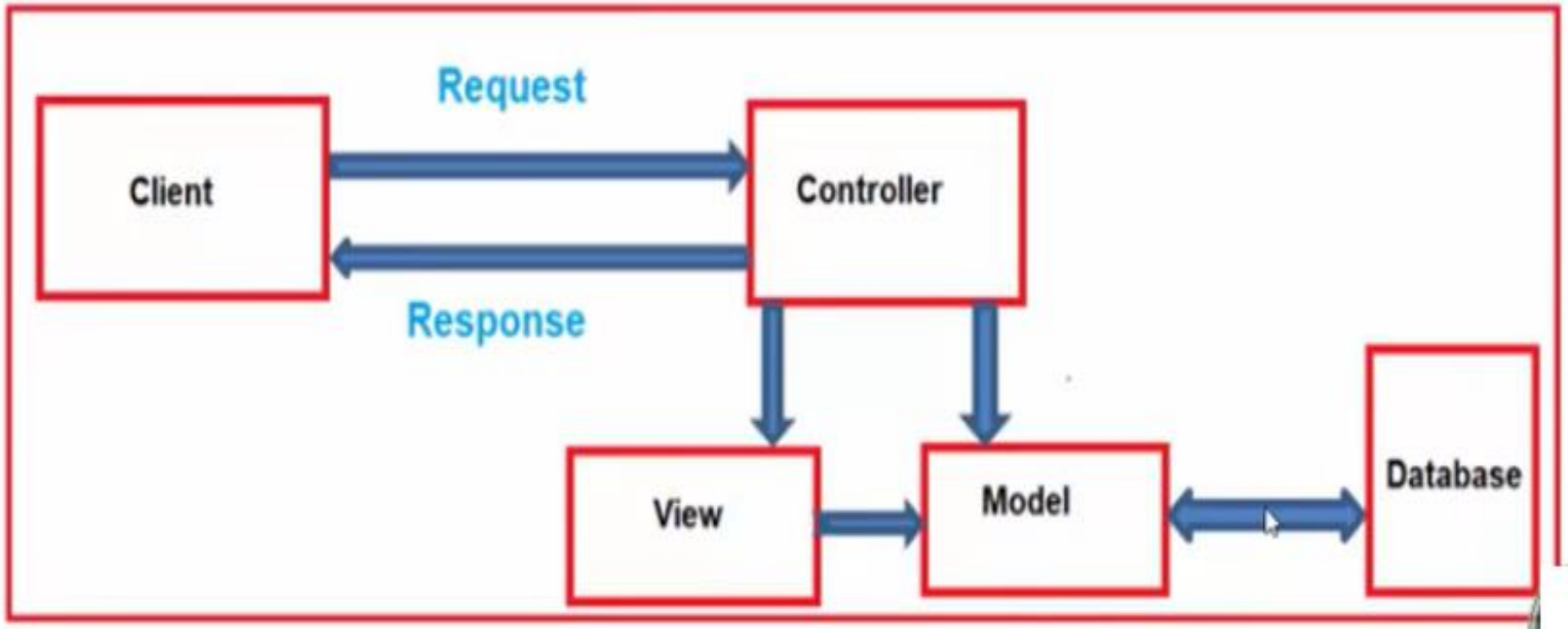
How MVC Design Pattern Works?

- Let us see an example to understand how the MVC pattern works in asp.net core MVC application.
- For example, we want to design an application, where we need to display the student details in a web page as shown below.

Student ID: 2	
Name:	James
Gender:	Male
Branch:	CSE
Section:	A2

- So, when we issue a request something like "**http://techieasad.com/student/details/2**" from a web browser then the following things are happening in order to handle the request.

How MVC Design Pattern Works?

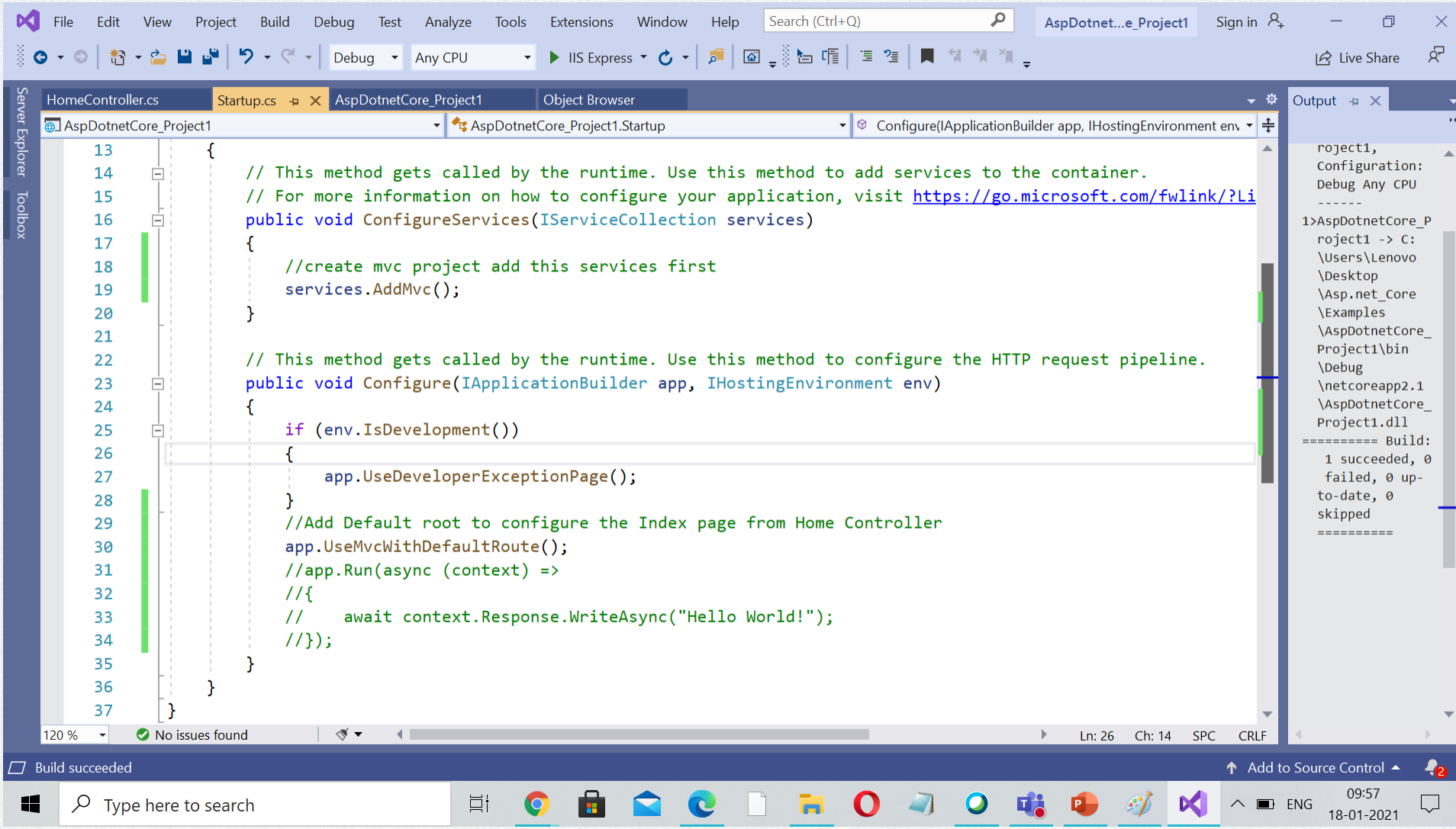


Where MVC is used in the real-time three-layer application?

- In general, a real-time application may consist of the following layers
  - **Presentation Layer:** This layer is responsible for interacting with the user.
  - **Business Layer:** This layer is responsible for implementing the core business logic of the application.
  - **Data Access Layer:** This layer is responsible for interacting with the database to perform the CRUD operations.
- The MVC design pattern is basically used to implement the Presentation Layer of the application. Please have a look at the following diagram

What is ASP.NET Core MVC?

- The **ASP.NET Core MVC** is a **lightweight, open-source, highly testable presentation framework** that is used for building web apps and Web APIs using the Model-View-Controller design pattern.
- The ASP.NET Core MVC Framework provides us with a patterns-based way to develop dynamic websites and web apps with a clean separation of concerns. This **ASP.NET Core MVC** framework provides us the full control over the mark-up. It also supports for Test-Driven Development and also uses the latest web standards.


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
AspDotnet...e_Project1
Sign in
Debug
Any CPU
IIS Express
Live Share
HomeController.cs
Startup.cs
AspDotnetCore_Project1
Object Browser
AspDotnetCore_Project1
AspDotnetCore_Project1.Startup
Configure(IApplicationBuilder app, IHostingEnvironment env
Server Explorer
Toolbox
13 {
14 // This method gets called by the runtime. Use this method to add services to the container.
15 // For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?Li
16 public void ConfigureServices(IServiceCollection services)
17 {
18 //create mvc project add this services first
19 services.AddMvc();
20 }
21
22 // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
23 public void Configure(IApplicationBuilder app, IHostingEnvironment env)
24 {
25 if (env.IsDevelopment())
26 {
27 app.UseDeveloperExceptionPage();
28 }
29 //Add Default root to configure the Index page from Home Controller
30 app.UseMvcWithDefaultRoute();
31 //app.Run(async (context) =>
32 //{
33 // await context.Response.WriteAsync("Hello World!");
34 //});
35 }
36 }
37 }
Output
roject1,
Configuration:
Debug Any CPU

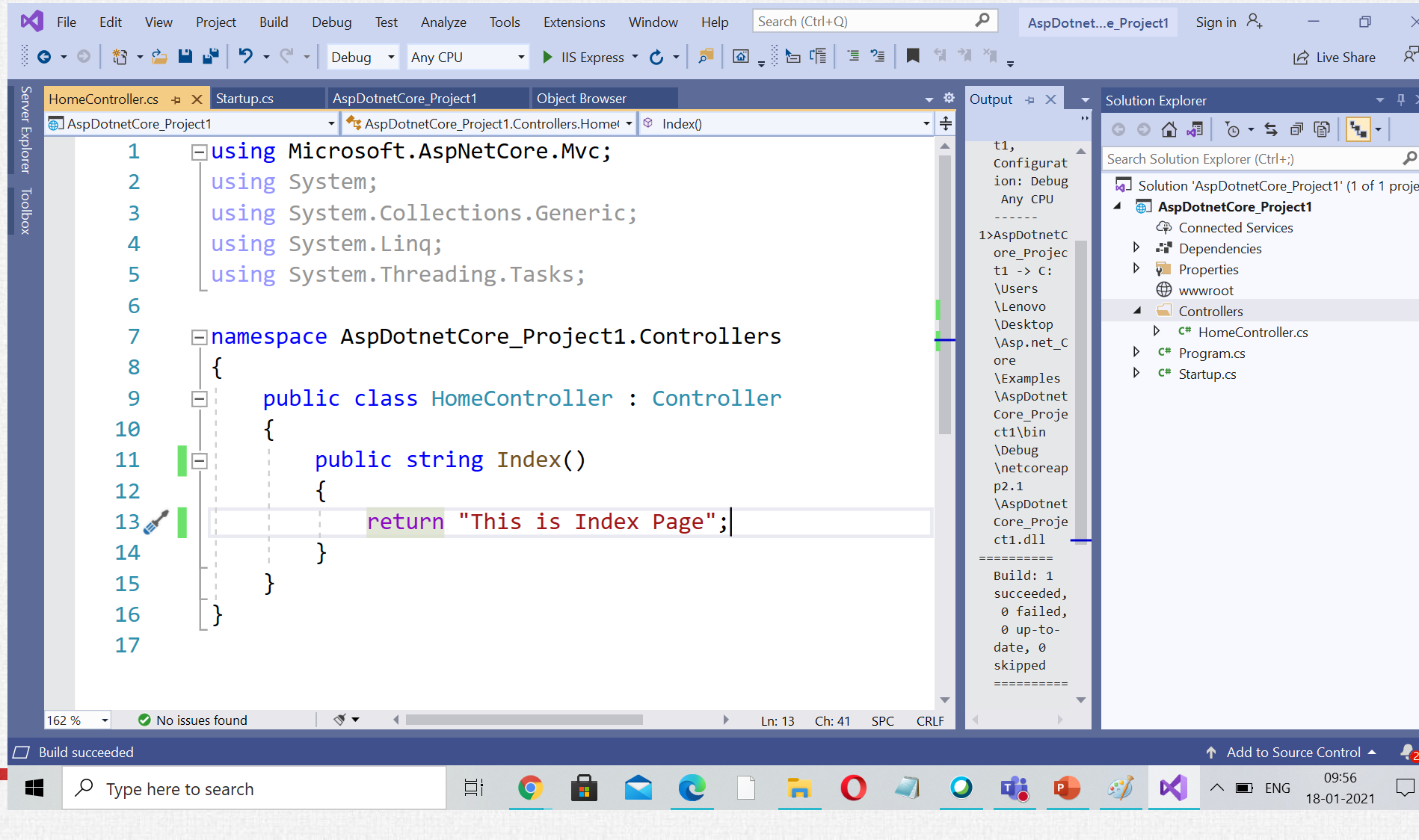
1>AspDotnetCore_P
roject1 -> C:
\Users\Lenovo
\Desktop
\Asp.net_Core
\Examples
\AspDotnetCore_
Project1\bin
\Debug
\netcoreapp2.1
\AspDotnetCore_
Project1.dll
========== Build:
1 succeeded, 0
failed, 0 up-
to-date, 0
skipped
==========
120 %
No issues found
Ln: 26 Ch: 14 SPC CRLF
Build succeeded
Add to Source Control
Type here to search
ENG
09:57
18-01-2021

How to Configure asp.net mvc page.

```
public void ConfigureServices(IServiceCollection services)
    {
       //create mvc project add this services first
       services.AddMvc();
    }

    // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
    public void Configure(IApplicationBuilder app, IHostingEnvironment env)
    {
       if (env.IsDevelopment())
       {
          app.UseDeveloperExceptionPage();
       }
       //Add Default root to configure the Index page from Home Controller
       app.UseMvcWithDefaultRoute();
       //app.Run(async (context) =>
       //{
       //    await context.Response.WriteAsync("Hello World!");
       //});
    }
```

- Create a Home Controller and create index page.



Thanks