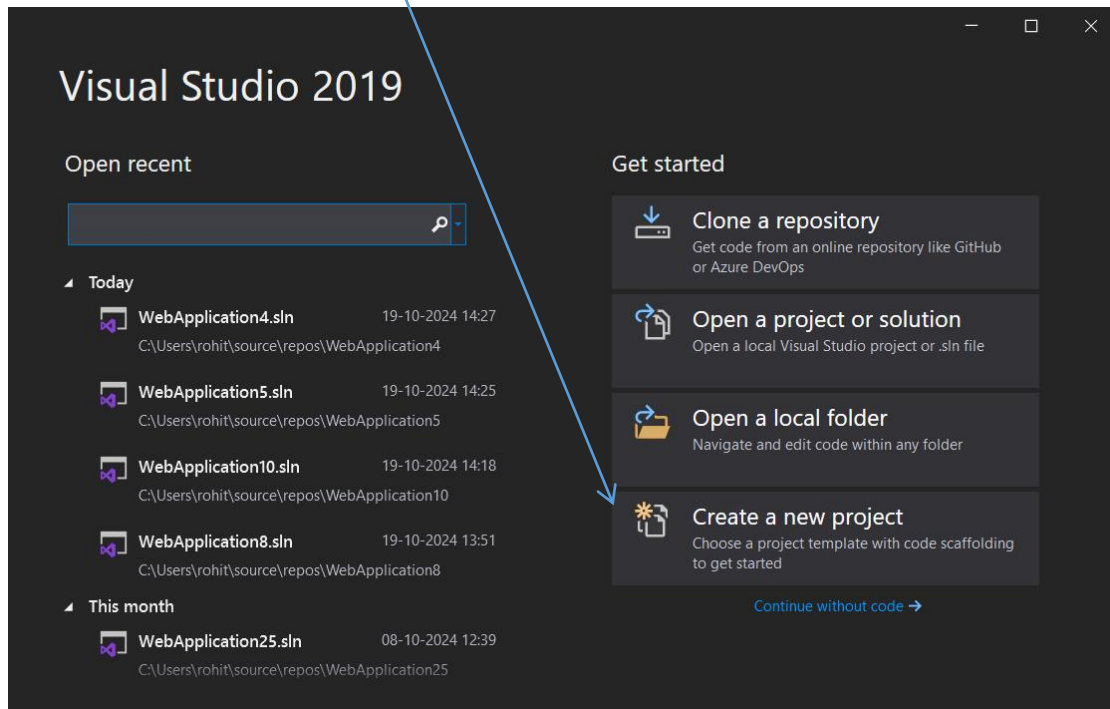
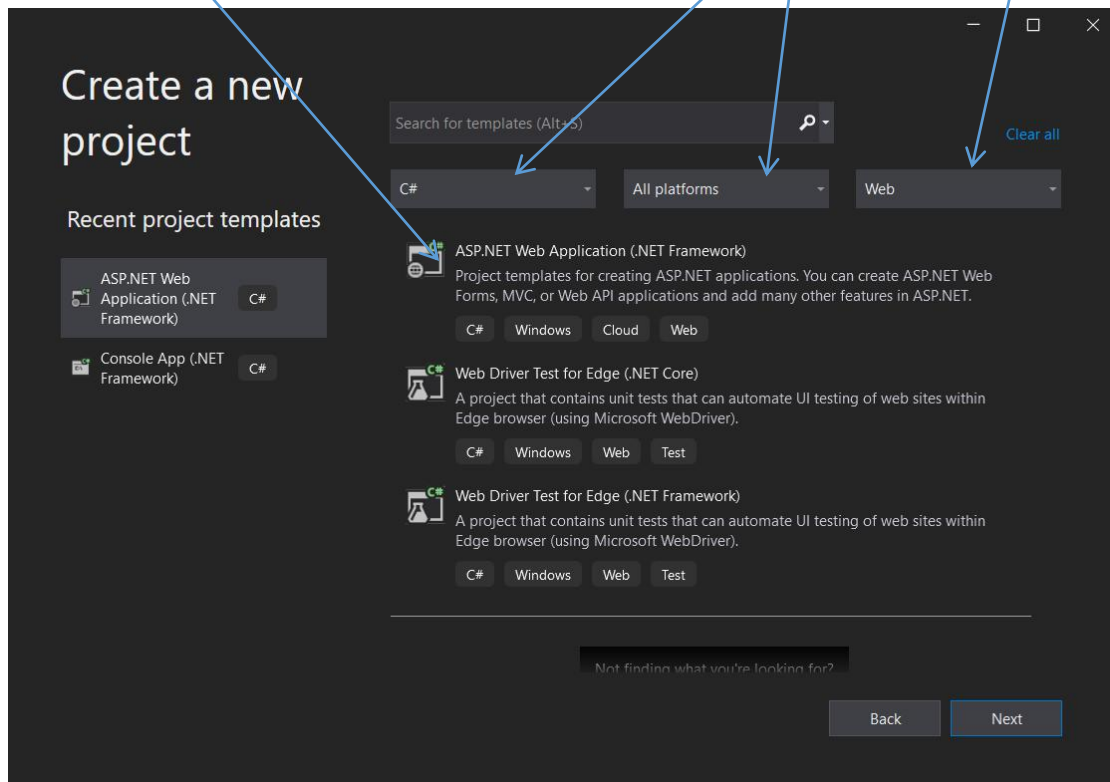


# ASP.NET TUTORIAL

## 1]CREATE A NEW PROJECT



## 2]SELECT ASP.NET WEB APPLICATION(CHOOSE SPECIFIED PLATFORMS(C#,ALL PLATFORM,WEB))



# ASP.NET TUTORIAL

## 3]CLICK NEXT AND NOW CLICK CREATE

Configure your new project

ASP.NET Web Application (.NET Framework) C# Windows Cloud Web

Project name  
WebApplication12

Location  
C:\Users\rohit\source\repos

Solution name ⓘ  
WebApplication12

☒ Place solution and project in the same directory

Framework  
.NET Framework 4.5

Back Create

## 4]CHOOSE EMPTY AND CREATE

Create a new ASP.NET Web Application

**Empty**  
An empty project template for creating ASP.NET applications. This template does not have any content in it.

**Web Forms**  
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

**MVC**  
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

**Web API**  
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

**Single Page Application**  
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

**Authentication**  
No Authentication  
[Change](#)

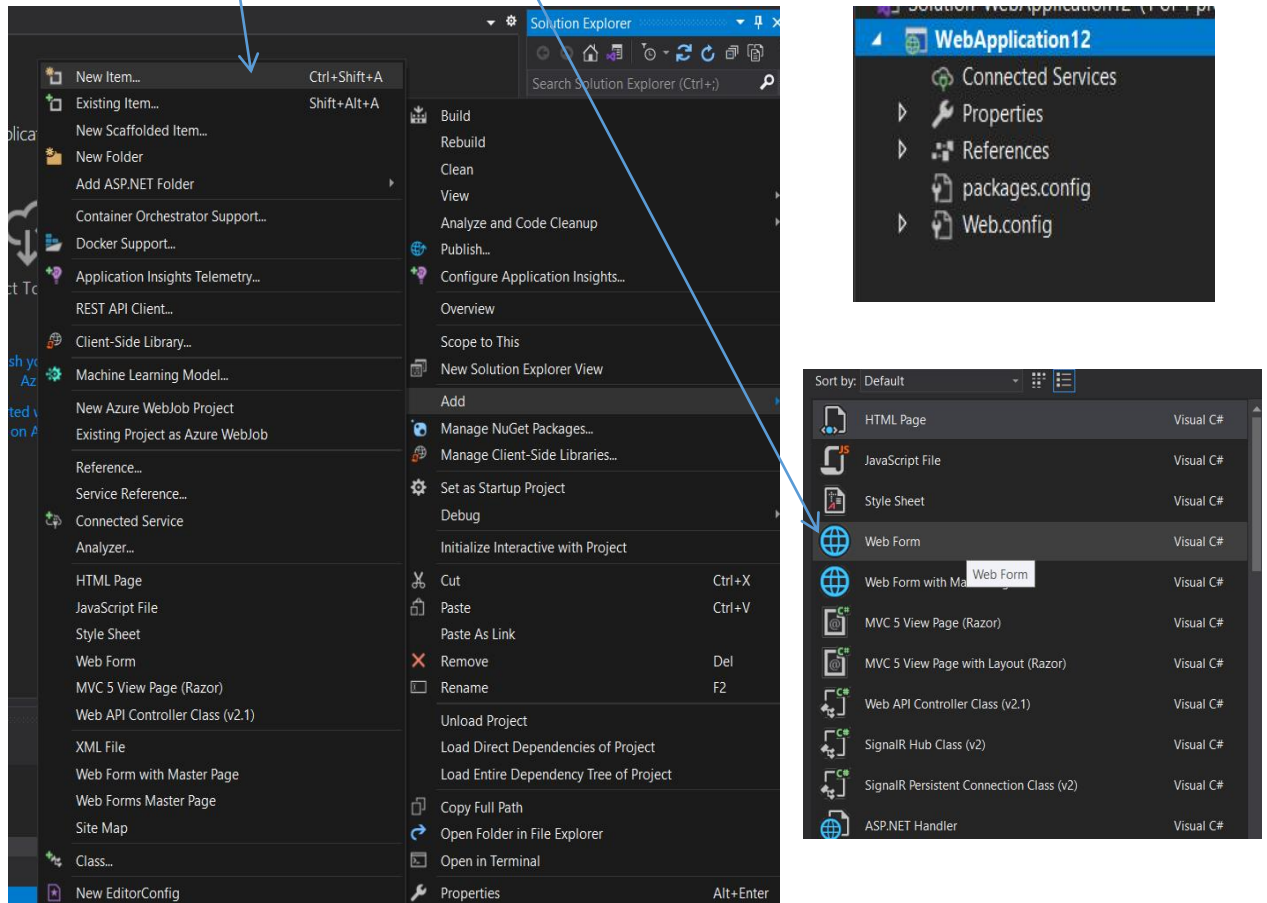
**Add folders & core references**  
☐ Web Forms  
☐ MVC  
☐ Web API

**Advanced**  
☒ Configure for HTTPS  
☐ Docker support  
(Requires [Docker Desktop](#))  
☐ Also create a project for unit tests  
WebApplication12.Tests

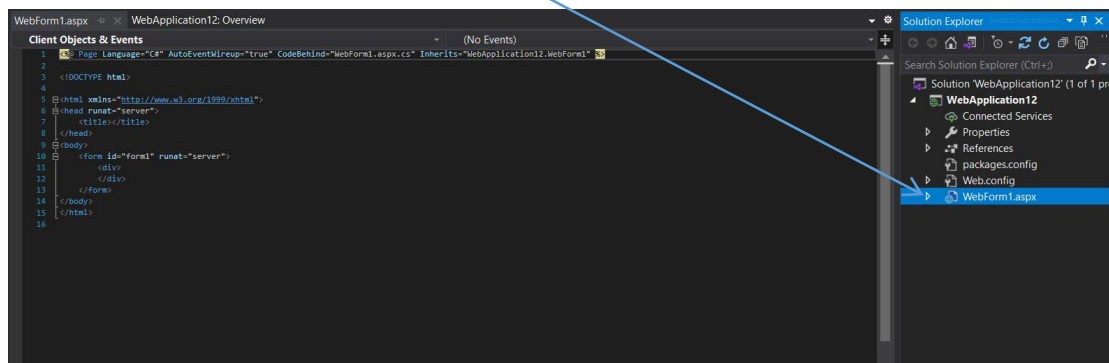
Back Create

## ASP.NET TUTORIAL

5]RIGHT CLICK ON WEB APPLICATION AND PRESS ADD THEN NEW ITEM AND SELECT WEBFORM AND PRESS ENTER



6]GOTO THE SOLUTION EXPLORER AND OPEN THE WebForm1.aspx file



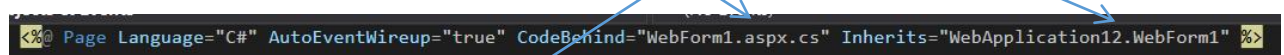
## ASP.NET TUTORIAL

### 7]IMPORTANT-

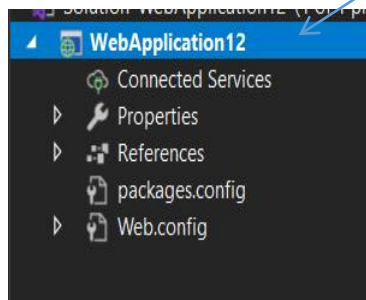
CHANGE THE CODE BEHIND FILE NAME TO YOUR CURRENT FILE IN MY CASE

WebForm1.aspx.cs(Just add .cs to your aspx file )

ALSO CHANGE THE Inherits to YOUR PROJECT NAME .WebForm NAME IN MY CASE MY PROJECT NAME IS WebApplication12 AND CHANGE YOUR INHERIT IN MY CASE GIVEN



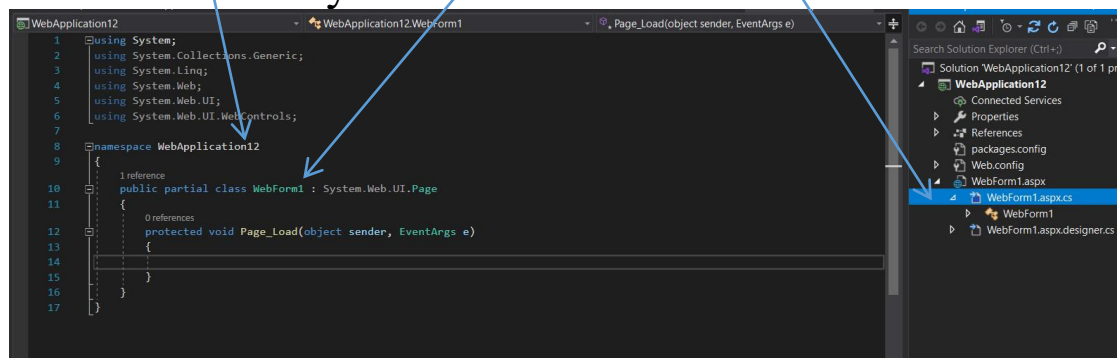
The screenshot shows the Page directive in the code-behind file: `<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication12.WebForm1" %>`. Blue arrows point from the text above to the `CodeBehind` and `Inherits` attributes.



### 8]NOW GO TO YOUR ASPX.CS FILE AND MAKE CHANGES AS FOLLOWS

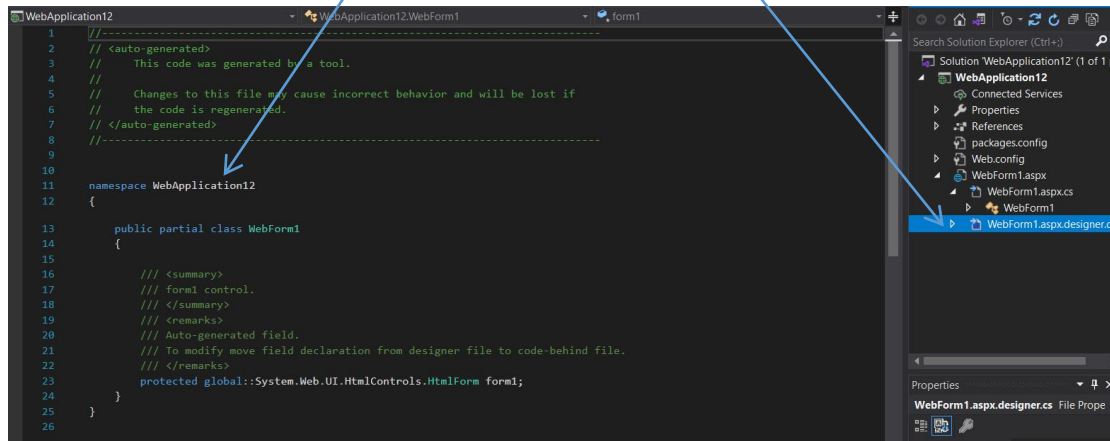
-CHANGE WebApplication Name as Your Project Name

-Select Your WebForm name as public partial class WebForm1 in my case

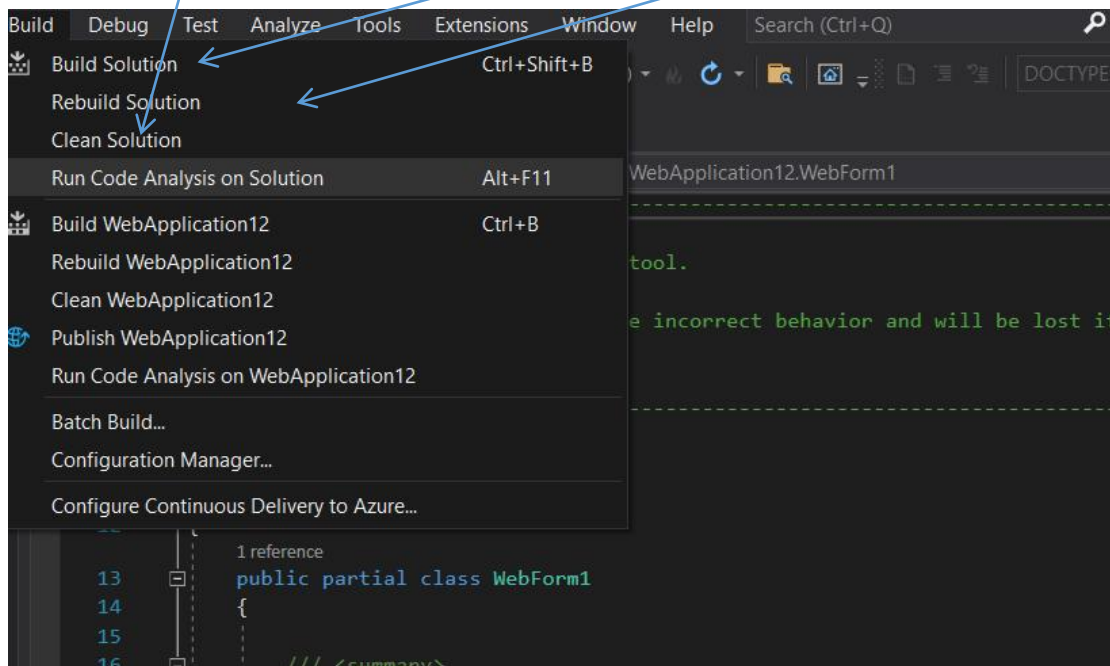


## ASP.NET TUTORIAL

9] NOW GO TO ASPX.DESIGNER FILE CHANGE THE NAMESPACE TO YOUR NAMESPACE IN MY CASE WebApplication12

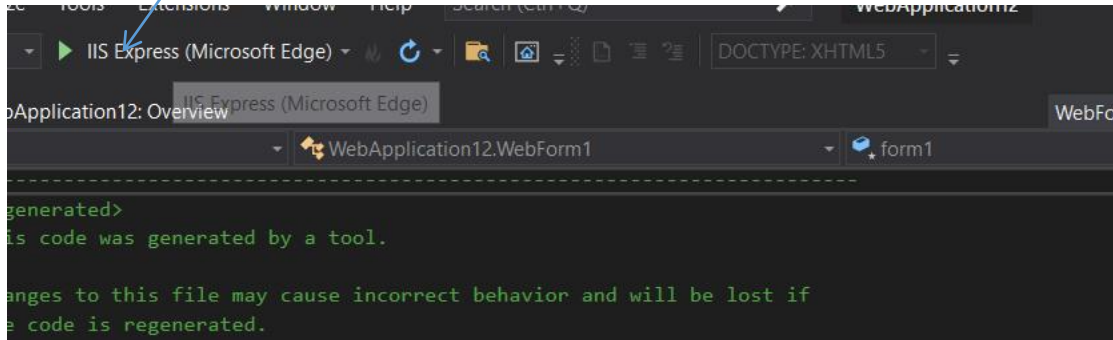


10] NOW COPY PASTE THE GIVEN CODE PLACE ASPX CODE IN WebForm.aspx File and Logical Code in WebForm.aspx.cs AND AT LAST BUILD REBUILD AND CLEAN SOLUTION



## ASP.NET TUTORIAL

11] NOW RUN THE PROGRAM YOU SHOULD RECEIVE THE DESIRED OUTPUT



THIS IS FOR FUN PURPOSE NO MISUSE  
INTENDED

“FROM YOURS SHUBH CHINTAK”

“THANK YOU ”

