# Webforms Lab: Visual Studio Projekt

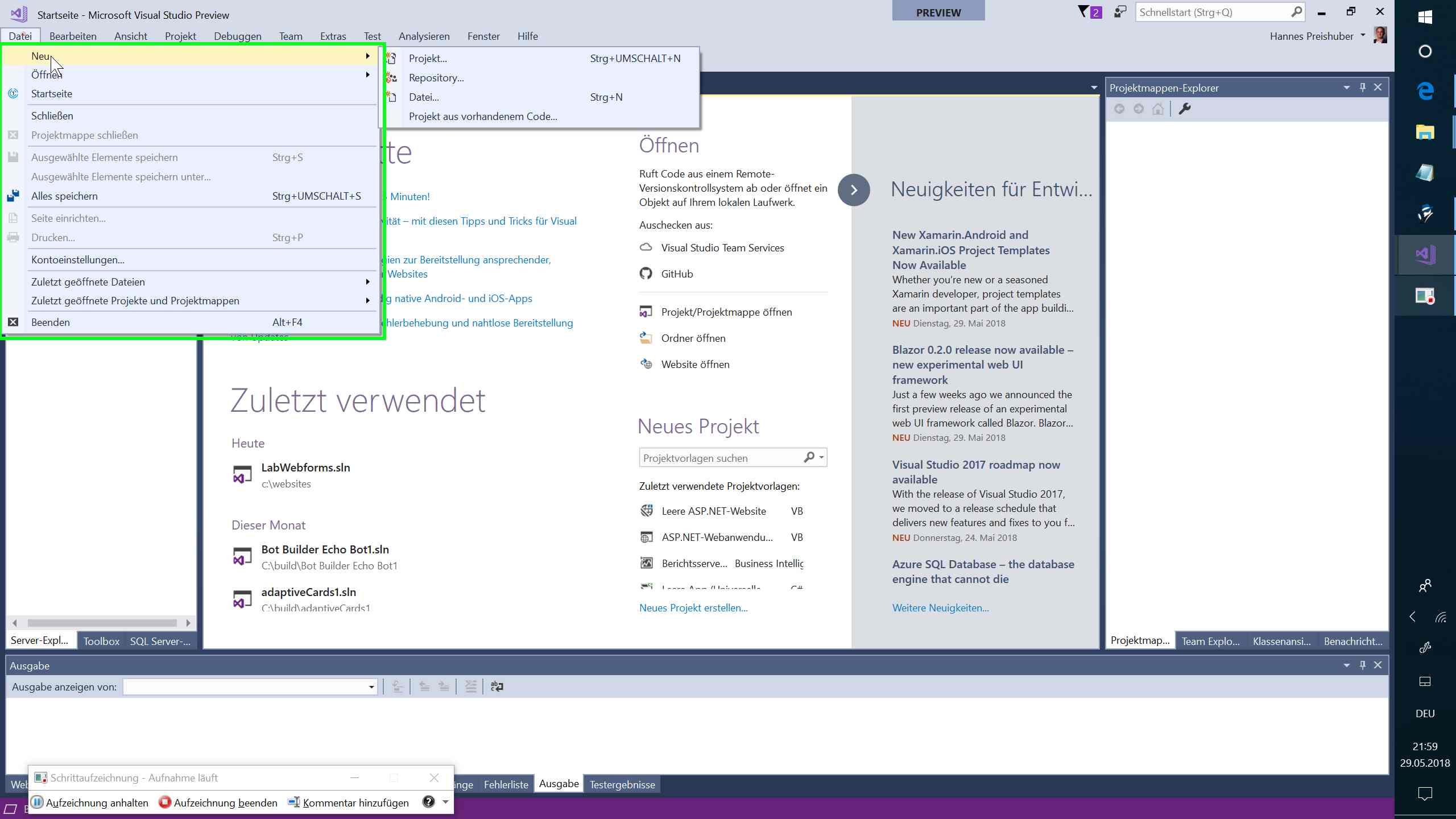
Dieses Lab hilft die Fähigkeiten zu vertiefen in den Bereichen

* Das Visual Studio Web Projekt System
* Paketmanagement mit Nuget
* UI Design mit Bootstrap und CSS
* HMTL JavaScript und Server generierter Code
* Das Webforms Event Model

Starten Sie Visual Studio 2017.

Klicken Sie mit der linken Maustaste auf "Datei (Menüelement)".

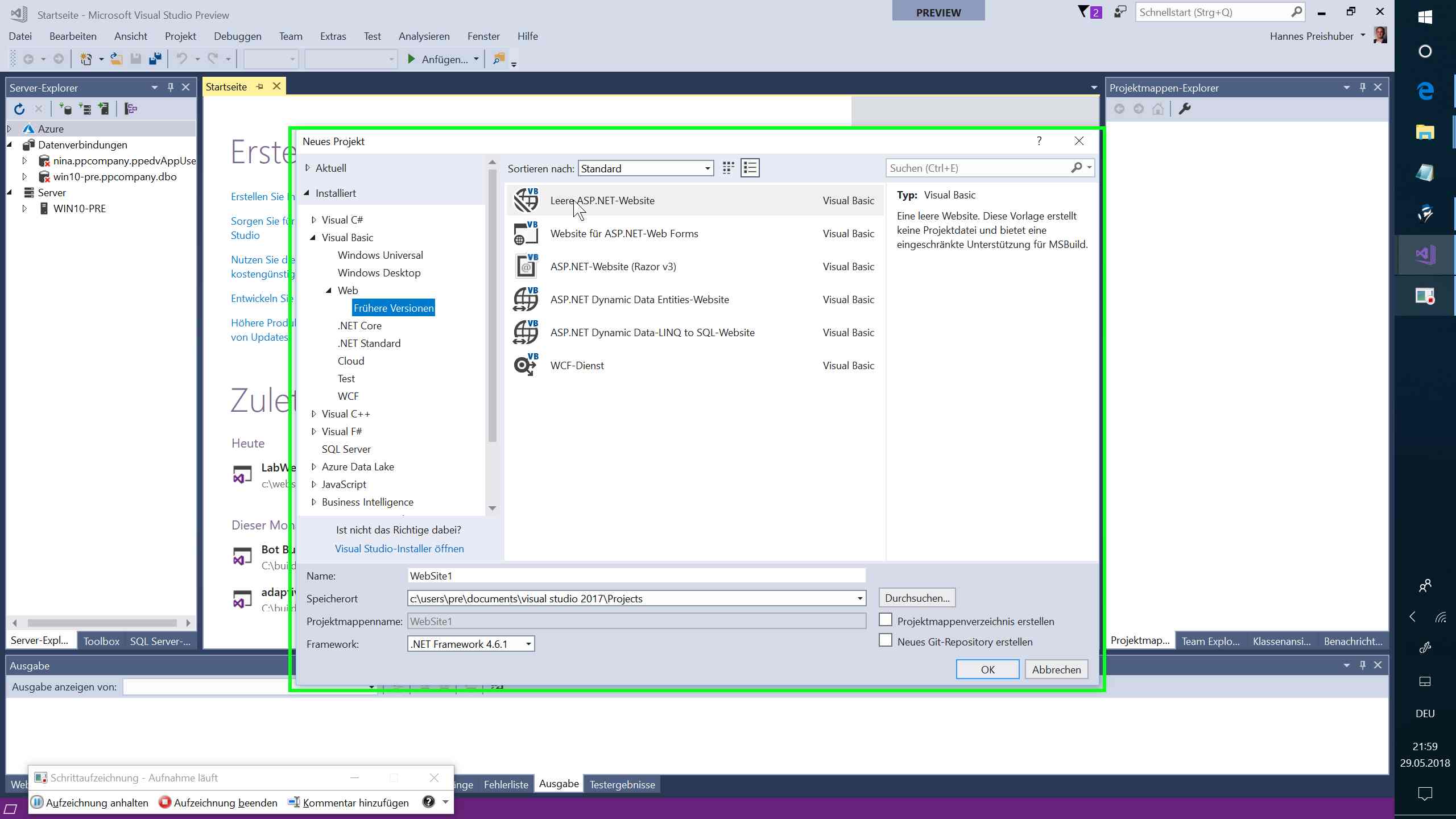
Klicken Sie mit der linken Maustaste auf "Neu (Menüelement)".

Bildschirmfoto Schritt 4

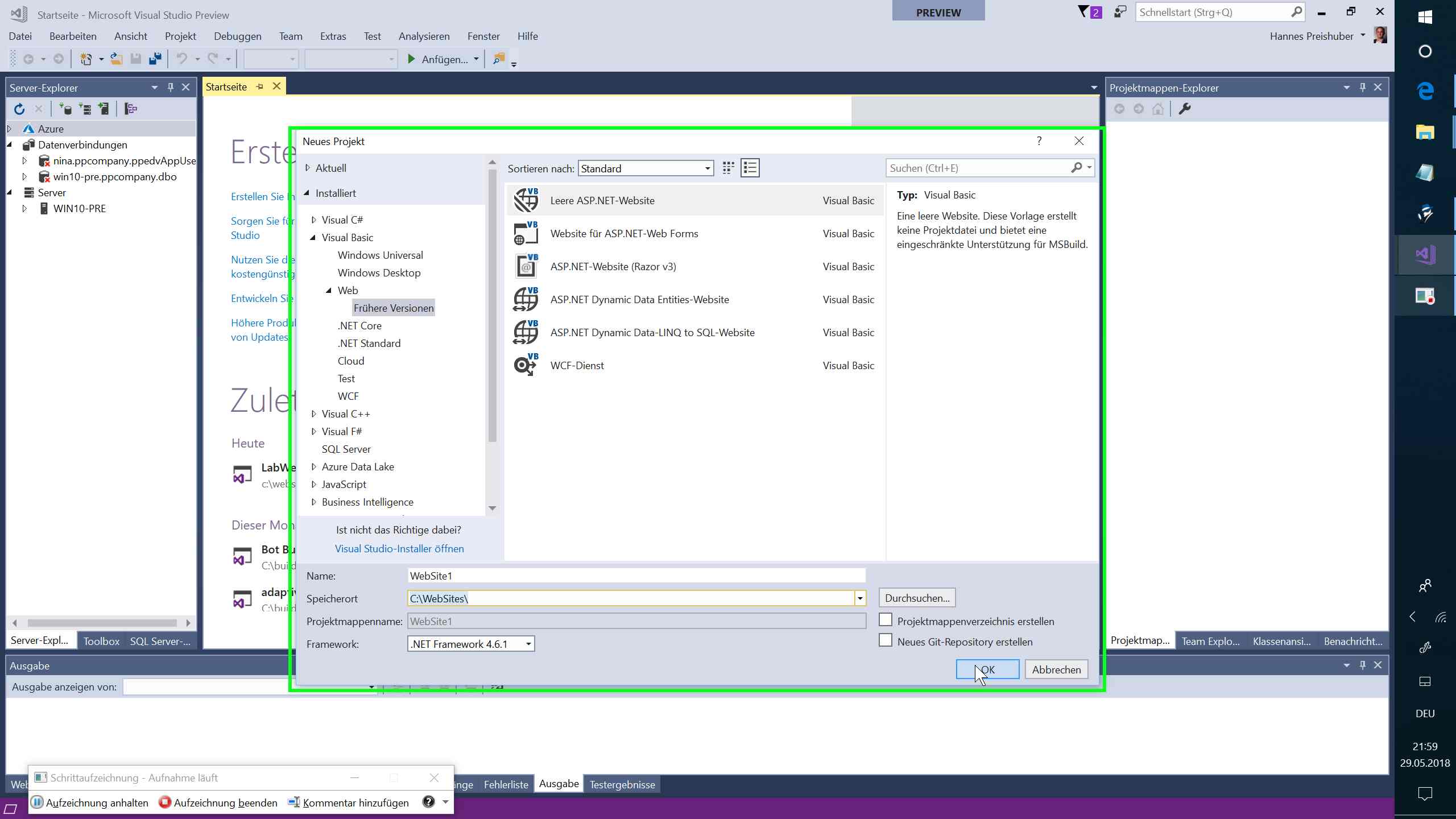
Klicken Sie mit der linken Maustaste auf "Projekt... (Menüelement)".

Klicken Sie mit der linken Maustaste auf "Web" und "Frühere Versionen".

Klicken Sie mit der linken Maustaste auf "Leere ASP.NET-Website (Text)".

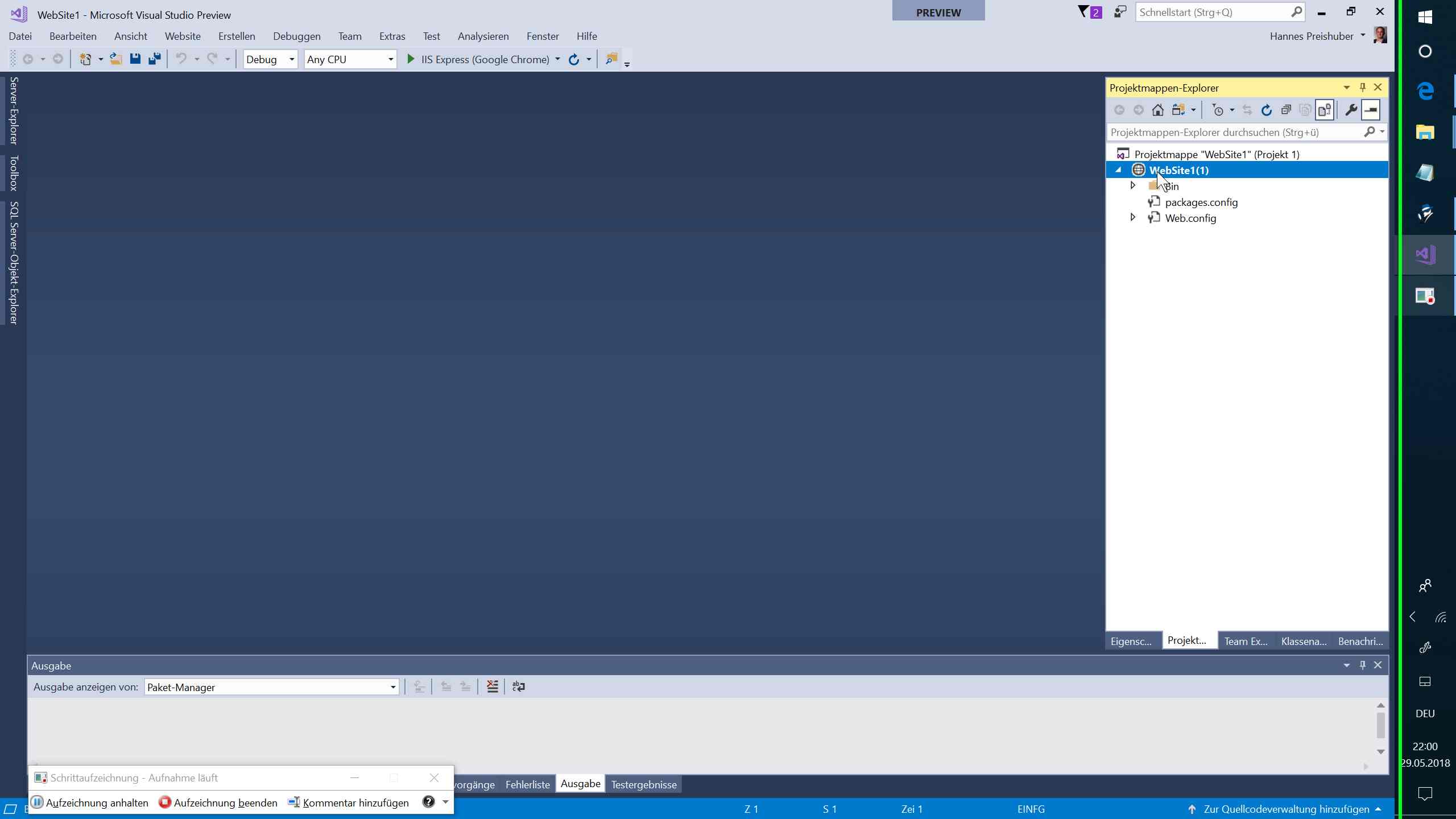
Bildschirmfoto Schritt 6

Tragen Sie im Feld "Speicherort" den Pfad "C:\Websites" ein und klicken Sie auf den OK-Button.

Bildschirmfoto Schritt 9

Die Erstellung der Website dauert mitunter einige Minuten und erfordert eine aktive Internetverbindung um fehlende Pakete aus dem Internet zu laden.

Es wird ein neues Website Projekt in der Verzeichnisstruktur angelegt.

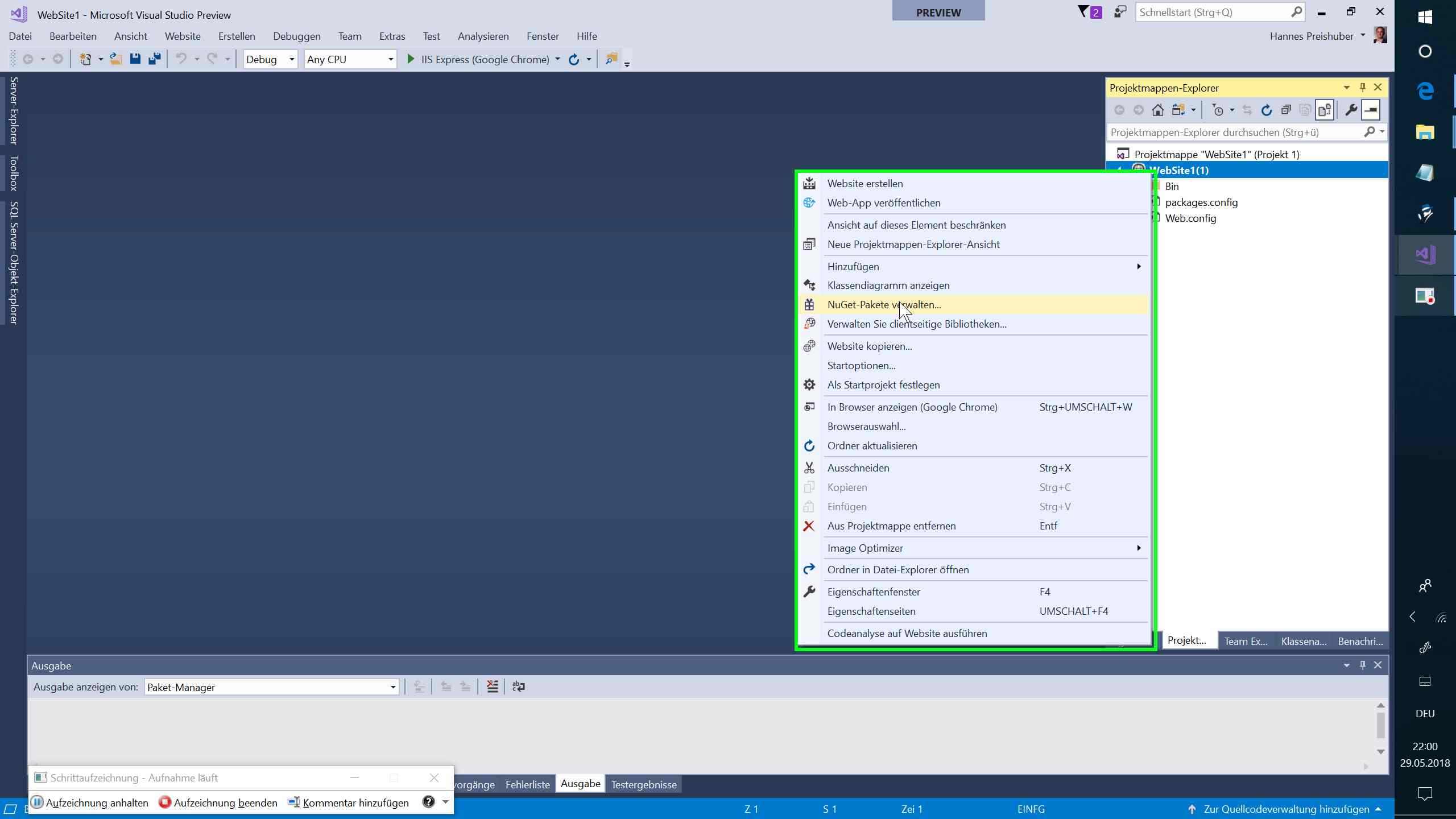


Bildschirmfoto Schritt 11

## Fehlende Pakete ergänzen

Paketmanager helfen Softwareteile wie DLLs dem Projekt hinzuzufügen und aktuell zu halten

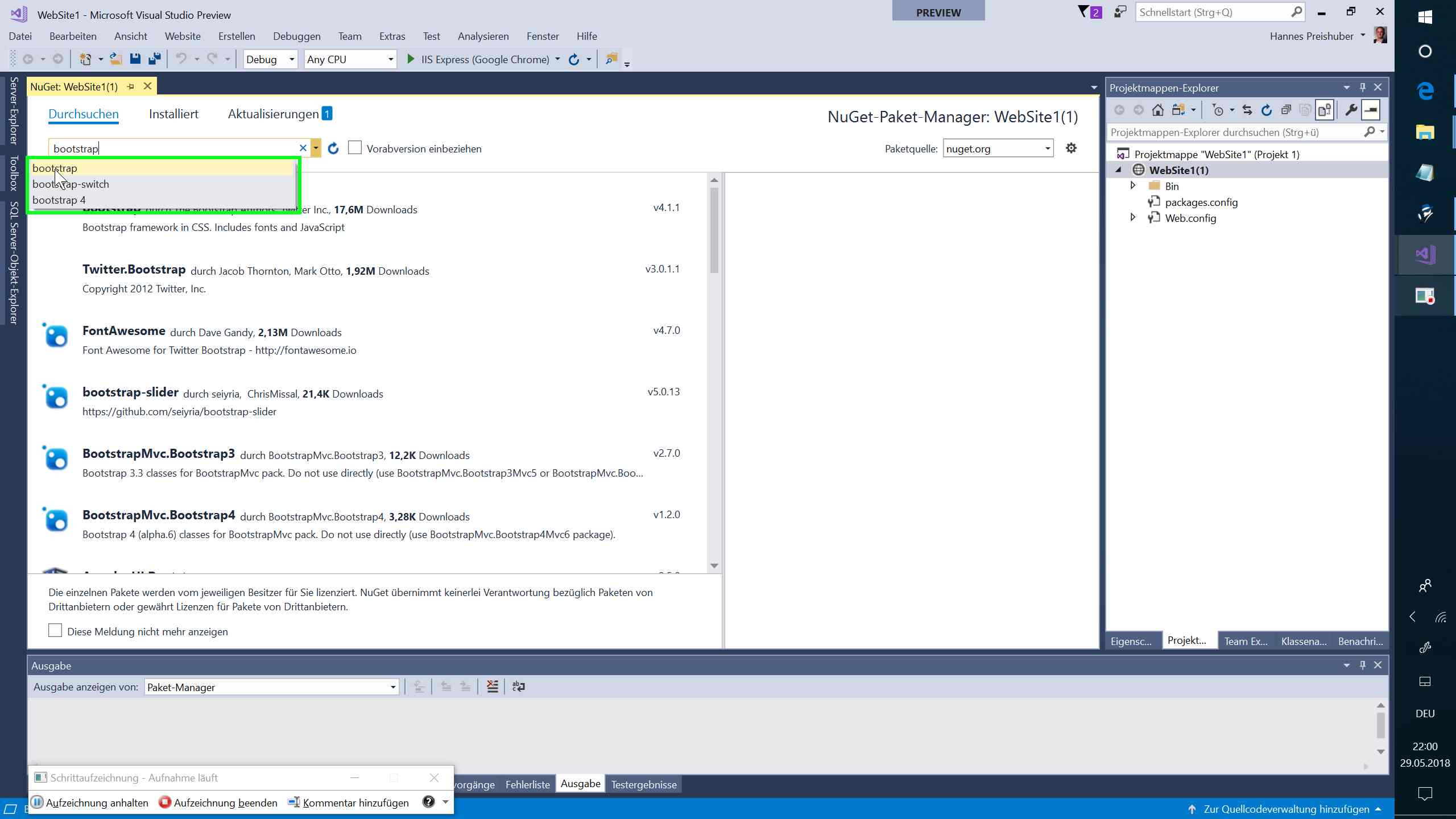
Klicken Sie mit der linken Maustaste auf "NuGet-Pakete verwalten...".

Bildschirmfoto Schritt 12

Klicken Sie mit der linken Maustaste auf "Web"

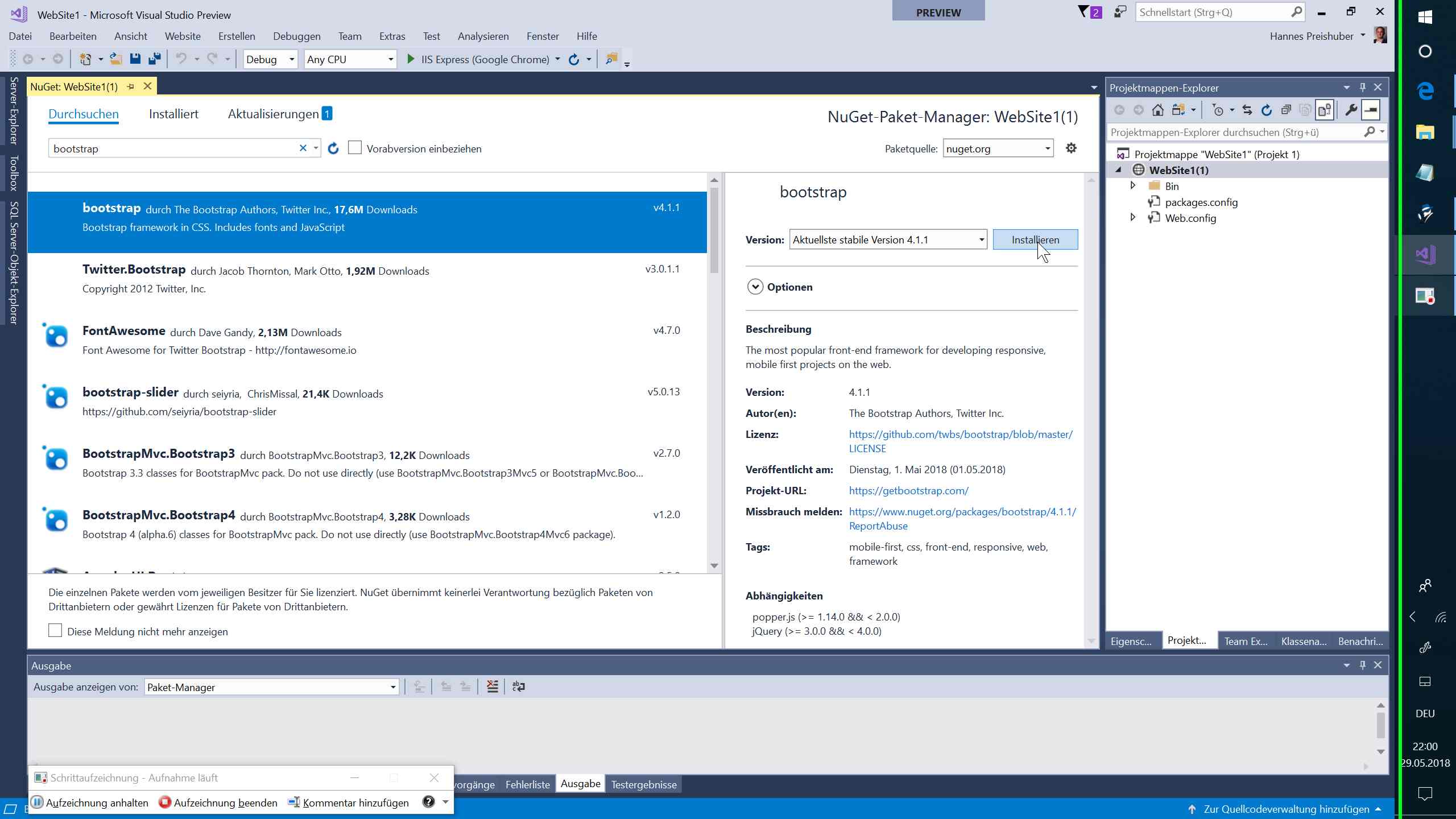
Aktualisierung der Pakete durchführen.

Klicken Sie mit der linken Maustaste auf Durchsuchen und geben Sie die Zeichenfolge Bootstrap ein.



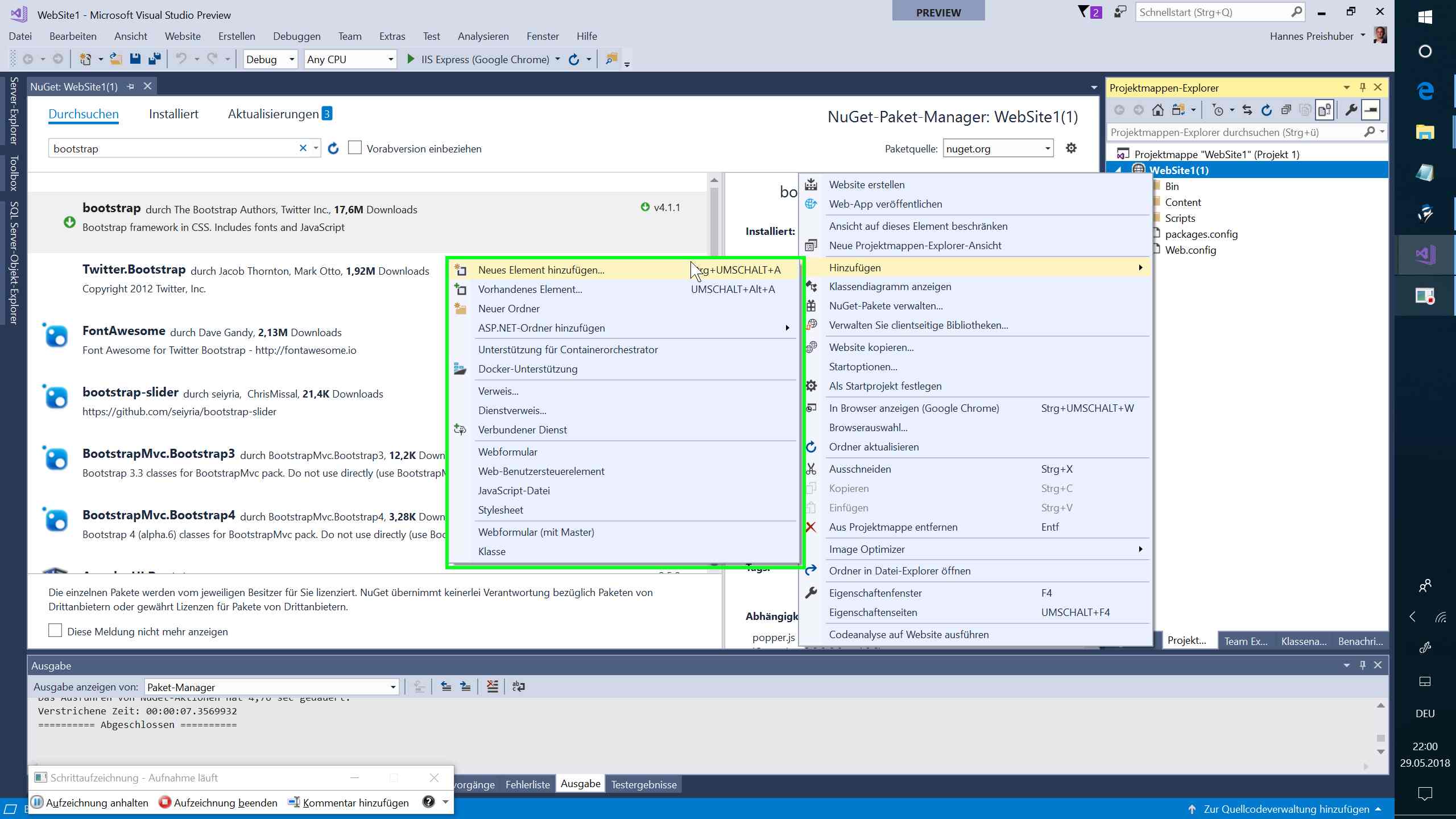
Bildschirmfoto Schritt 18

Bestätigen Sie den Eintrag Bootstrap (Version 4) mit Return

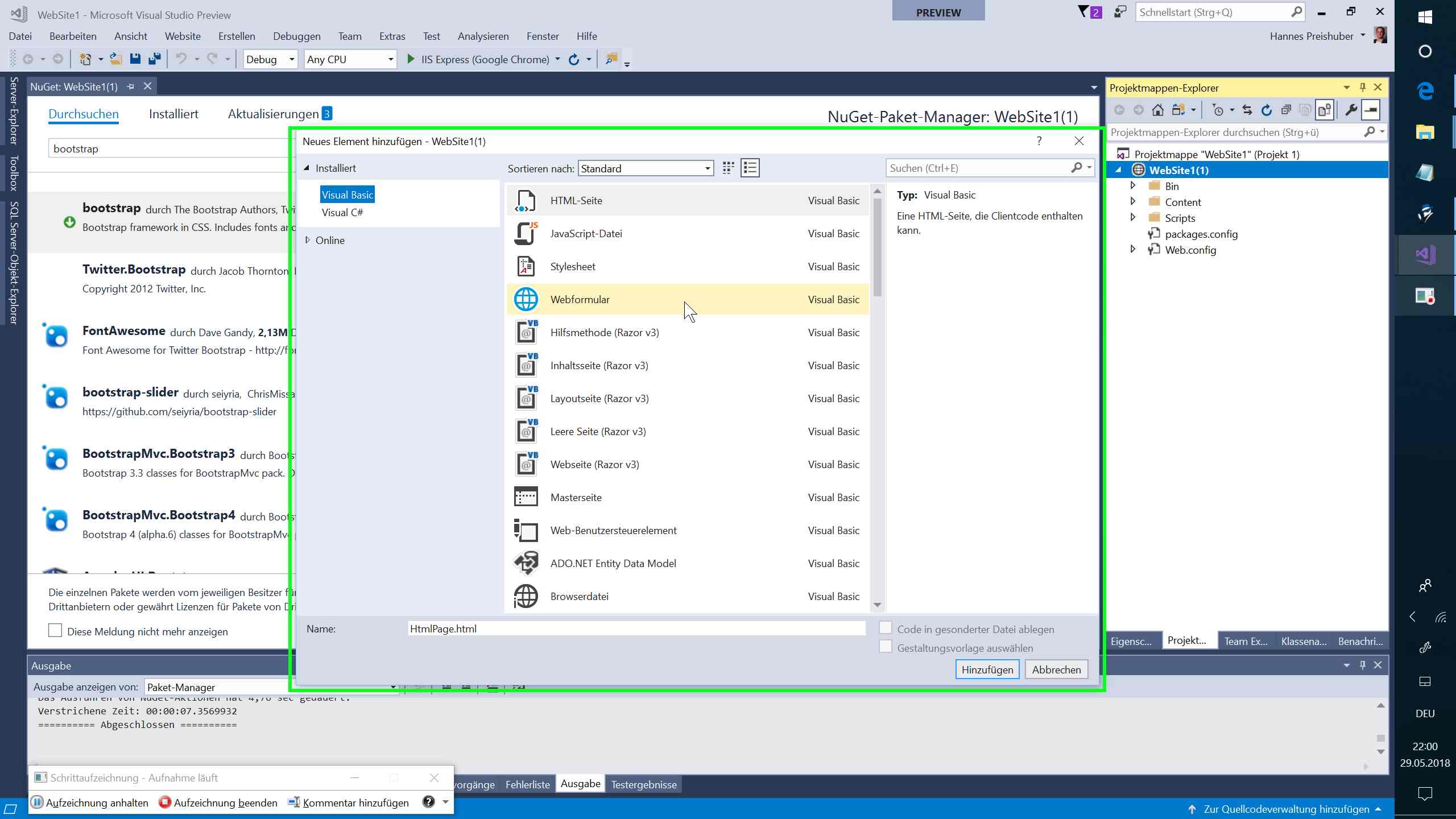
Bildschirmfoto Schritt 19

Bildschirmfoto Schritt 20

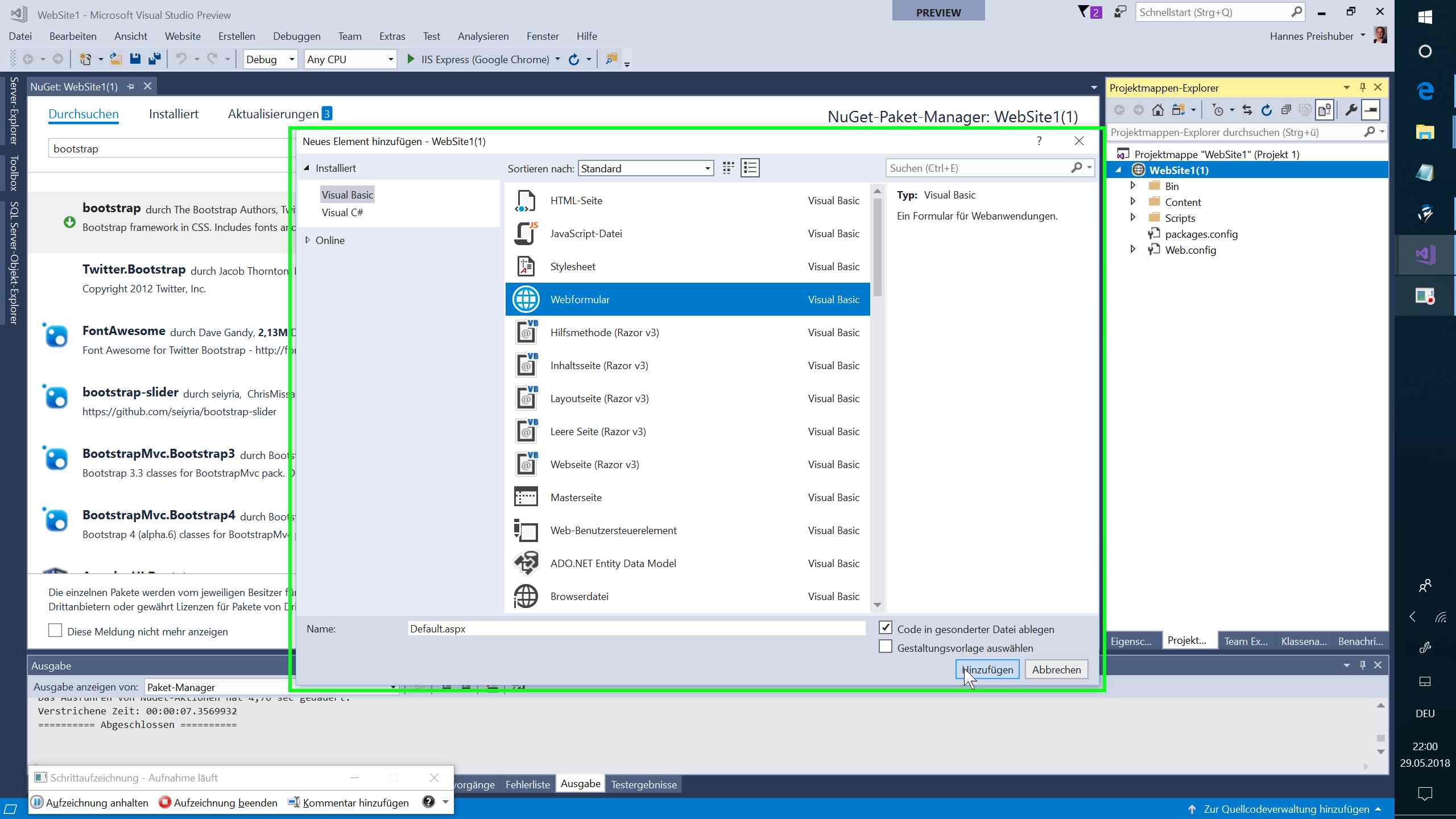
Klicken Sie mit der richten Maustaste auf das Projekt und dann auf "Neues Element hinzufügen...".

Bildschirmfoto Schritt 22

Klicken Sie mit der linken Maustaste auf "Webformular". Achten Sie darauf im linken Teil die richtige Programmiersprache auszuwählen.

Bildschirmfoto Schritt 23

Klicken Sie mit der linken Maustaste auf "\_Hinzufügen". Der Dateiname bleibt in der Voreinstellung default.aspx. Die Option "Code in gesonderter Datei ablegen" ist vorausgewählt. Die Option "Gestaltungsvorlage auswählen" wird später benötigt, wenn mit Master Pages gearbeitet wird und bleibt nicht ausgewählt.

Bildschirmfoto Schritt 24

Es erscheint der Code Editor mit HTML Code. Elementw mit dem Zusatz "runat=server" werden von ASP.NET auf Serverseite als Objekte verwaltet und mit dynamsichen HTML Rendering behandelt.

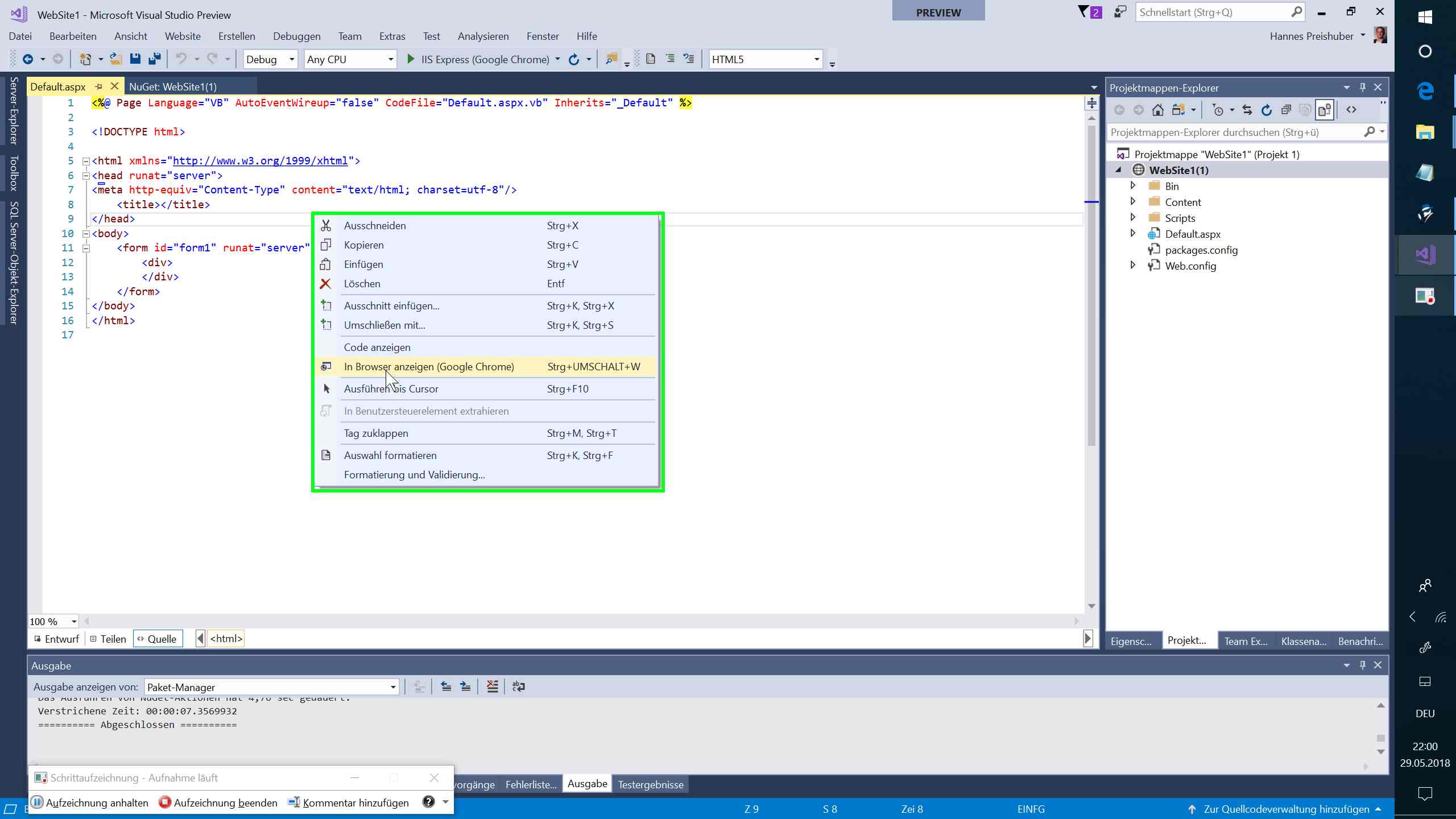
Im unteren Bereich lässt sich die WYSIWYG-Design-Ansicht umschalten. Probieren Sie die Optionen aus.

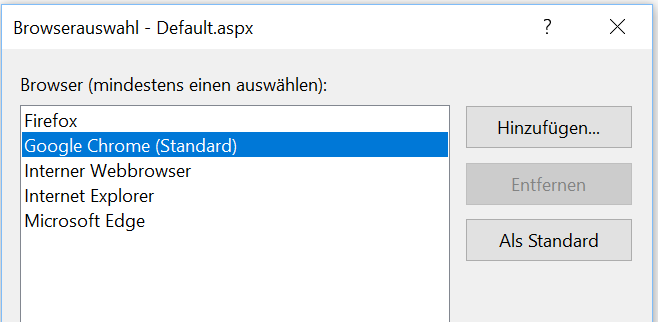
Bildschirmfoto Schritt 25

Die Website kann nun im Browser angezeigt werden.

Klicken Sie mit der rechten Maustaste in den Editor und wählen Sie "In Browser anzeigen".

Chrome muss dafür installiert sein. Dieser Schritt klappt auch mit anderen Browsern. Wenn Sie mehrere Browser installiert haben, lässt sich mit dem kleinem Pfeil oben in der Menüleiste, neben IIS Express, der Browser auswählen.

Bildschirmfoto Schritt 26



Es wird der Browser gestartet und eine Url beginnend mit Localhost aufgerufen. Die Nummer hinter dem Doppelpunkt ist der gewählte Port. Dieser wird bei Ihnen anders lauten. Er wird zufällig gewählt und dem Visual Studio Projekt dauerhaft zugeordnet

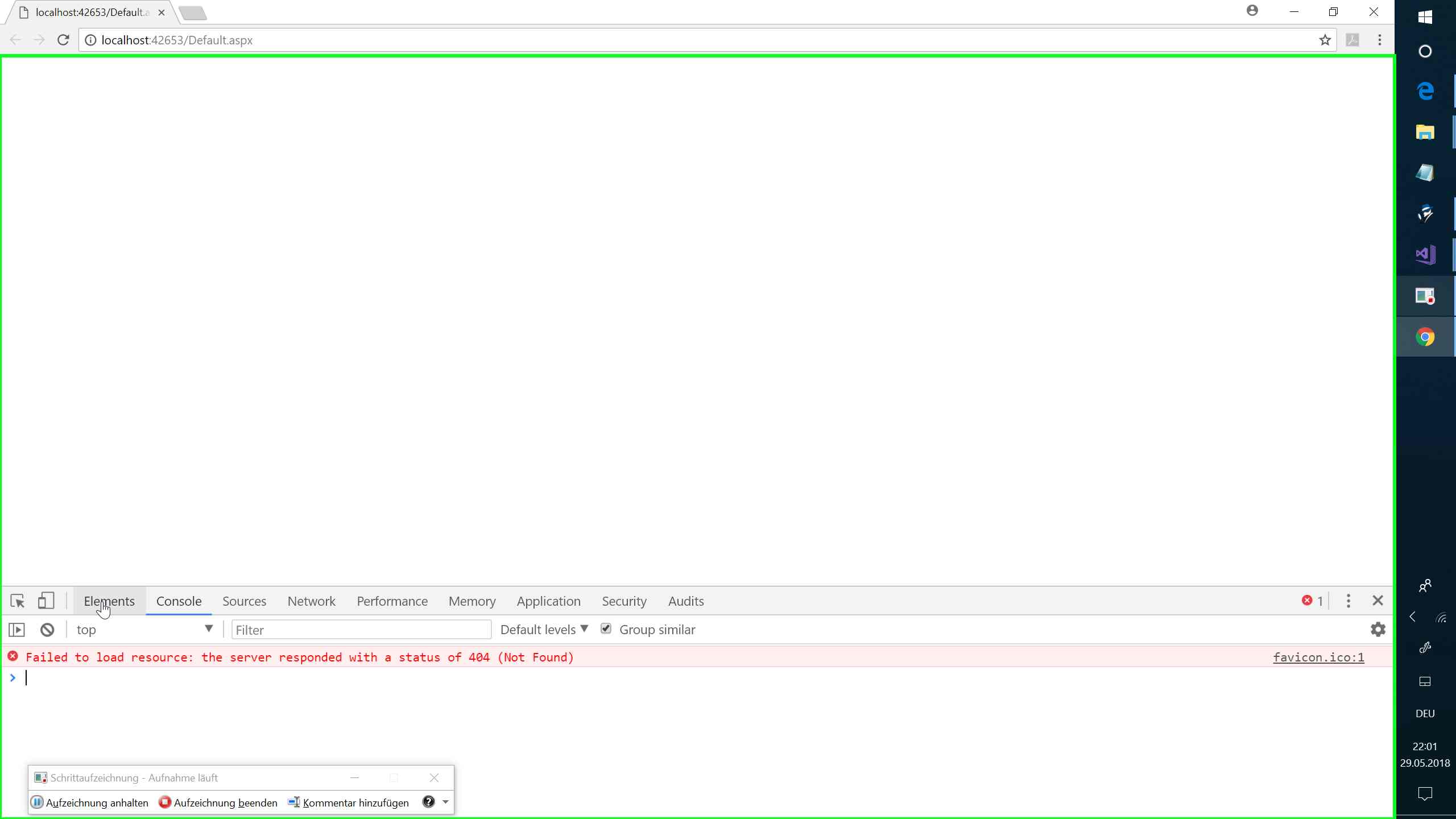
Bildschirmfoto Schritt 27

IIS 10.0 Express ist eine einfache und eigenständige Version von IIS, die für Entwickler optimiert ist. Über die Windows Statusleiste lässt sich der Webserver bei Bedarf beenden. Dies ist praktisch nie nötig, da Visual Studio diesen mit verwaltet.



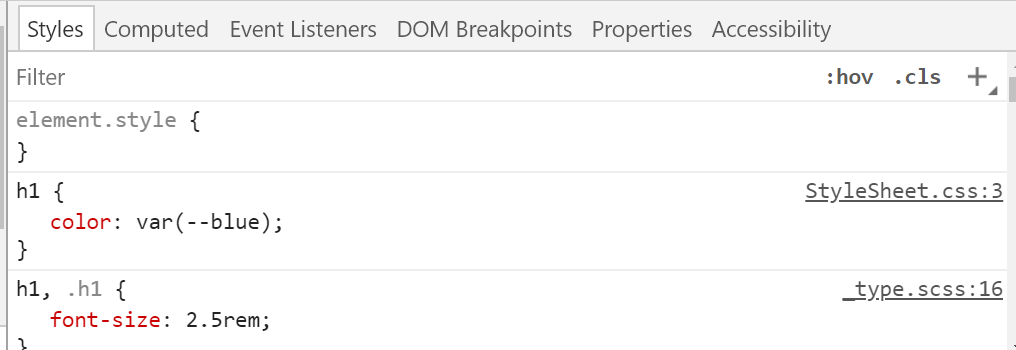
Öffnen Sie die Browser Entwicklerwerkzeuge mit F12

Im Tab Console sehen Sie Ausgaben des Browsers, die auf Fehler hinweisen. Hier fehlt das Favicon, das gentzt wird um in der URL ein kleines Symbol statt des "i" im Kreis darzustellen. Die Fehlermeldung kann für das Lab ignoriert werden.

Bildschirmfoto Schritt 28

Wählen Sie den Bereich Elements um im HTML Dom Elemente selektieren zu können. Hier erkennt man welchen HTML Code ASP.NET genau erzeugt hat. Inspizieren Sie das Form Element und benennen Sie den Unterschied zum Form Element in der ASPX Seite im Visual Studio Projekt.

Im rechten Bereich kann im Reiter Styles das Design des ausgewählten HTML DOM Elements inspiziert werden.



Es lassen sich dort auch direkt Änderungen vornehmen, die aber nicht dauerhaft gespeichert werden. Der Screenshot ist von einem späteren Zeitpunkt des Labs und bei Ihnen wird das noch anders aussehen.

Wechseln Sie zu Microsoft Visual Studio. Es werden nun die nötigen Dateien des Bootstrap UI Frameworks hinzugefügt.

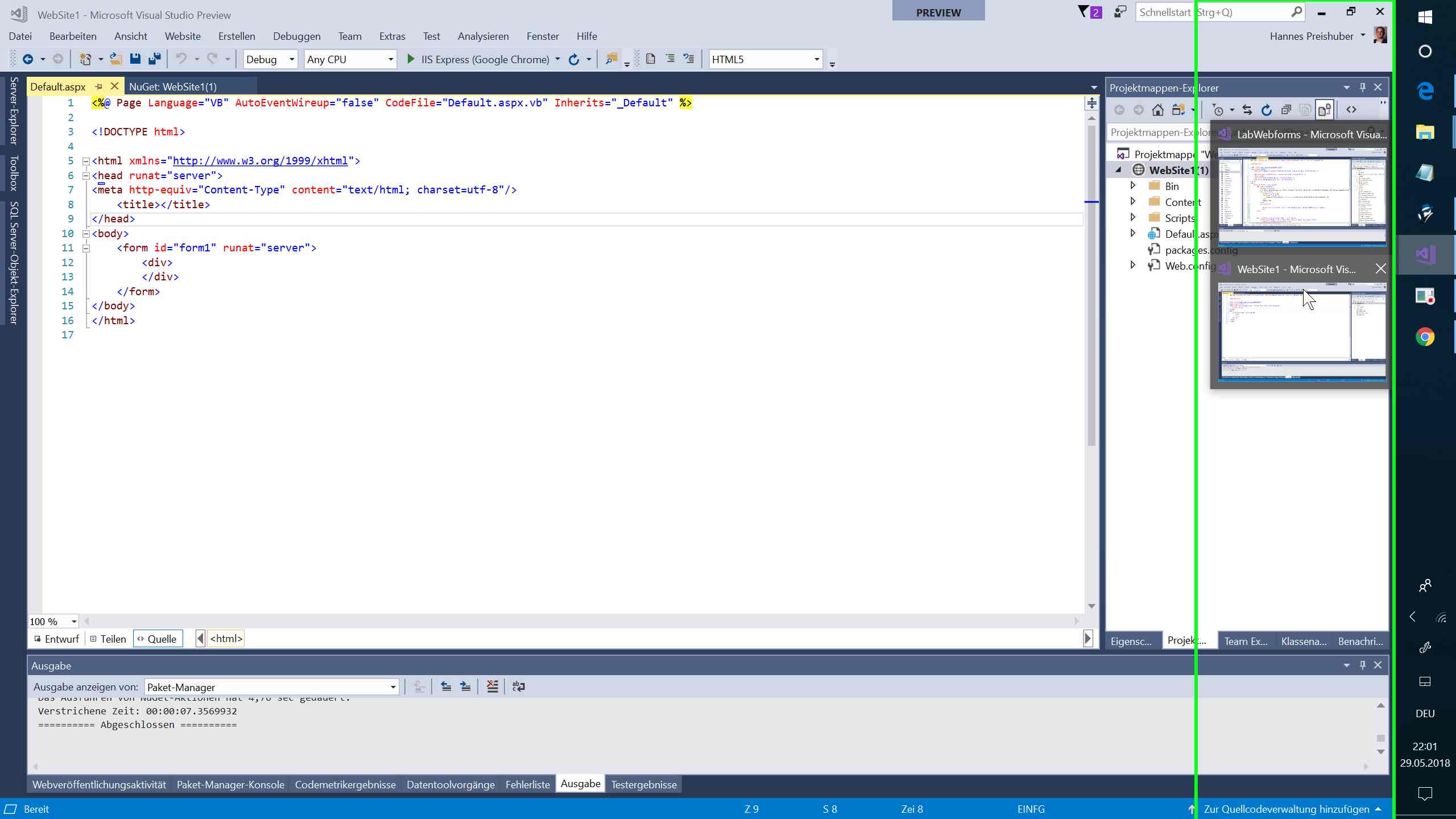
Diese sind:

Jquery.js

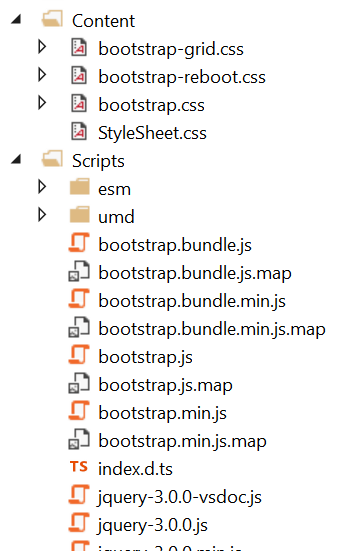
Boostrap.js

Boostrap.css

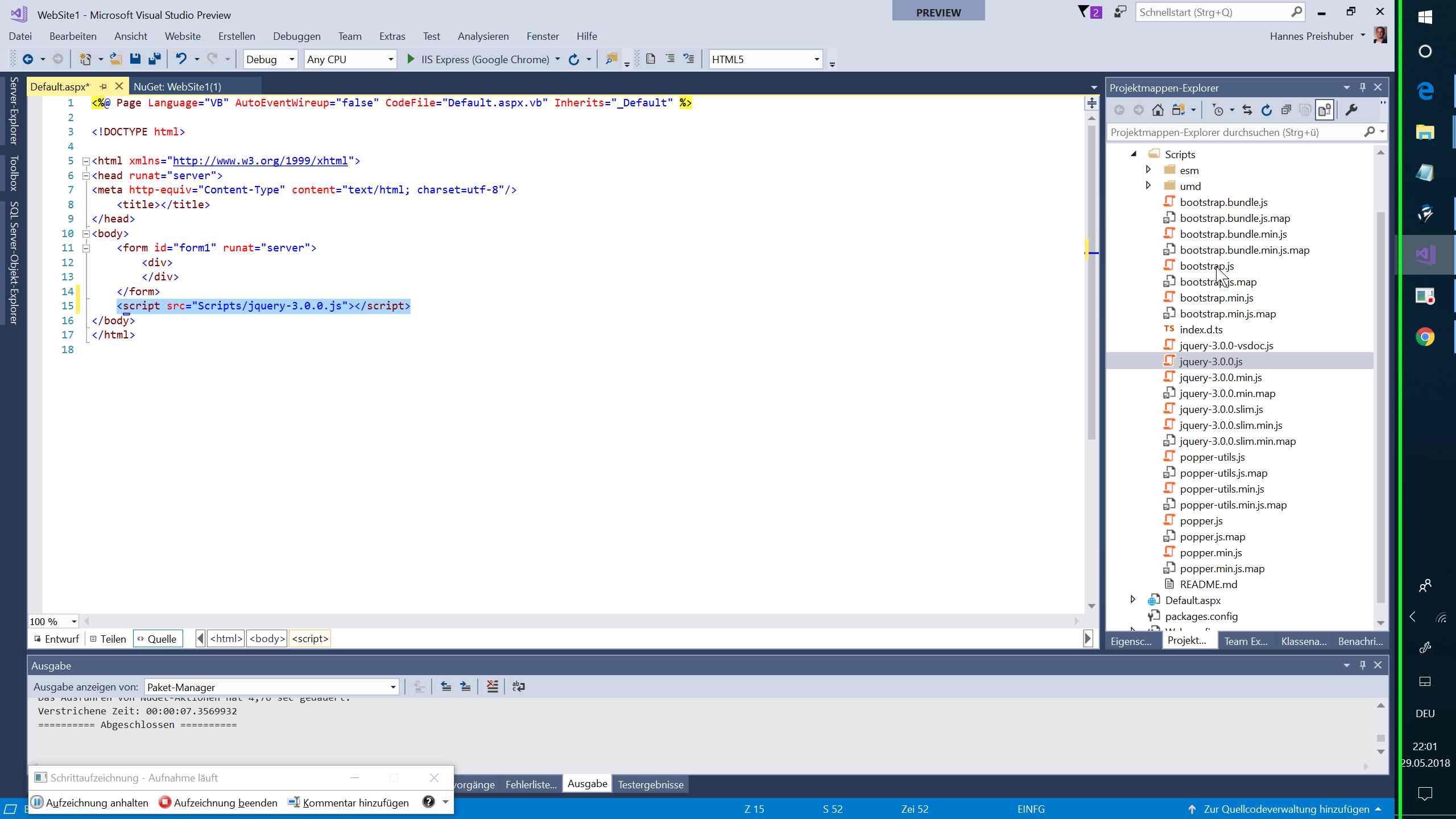
und eine eigene Style Datei

Bildschirmfoto Schritt 34

Dies geschieht per Drag und Drop aus dem rechten Bereich der Projektansicht. Die Verzeichnisse in denen die Dateien sich befinden lauten Scripts und Content

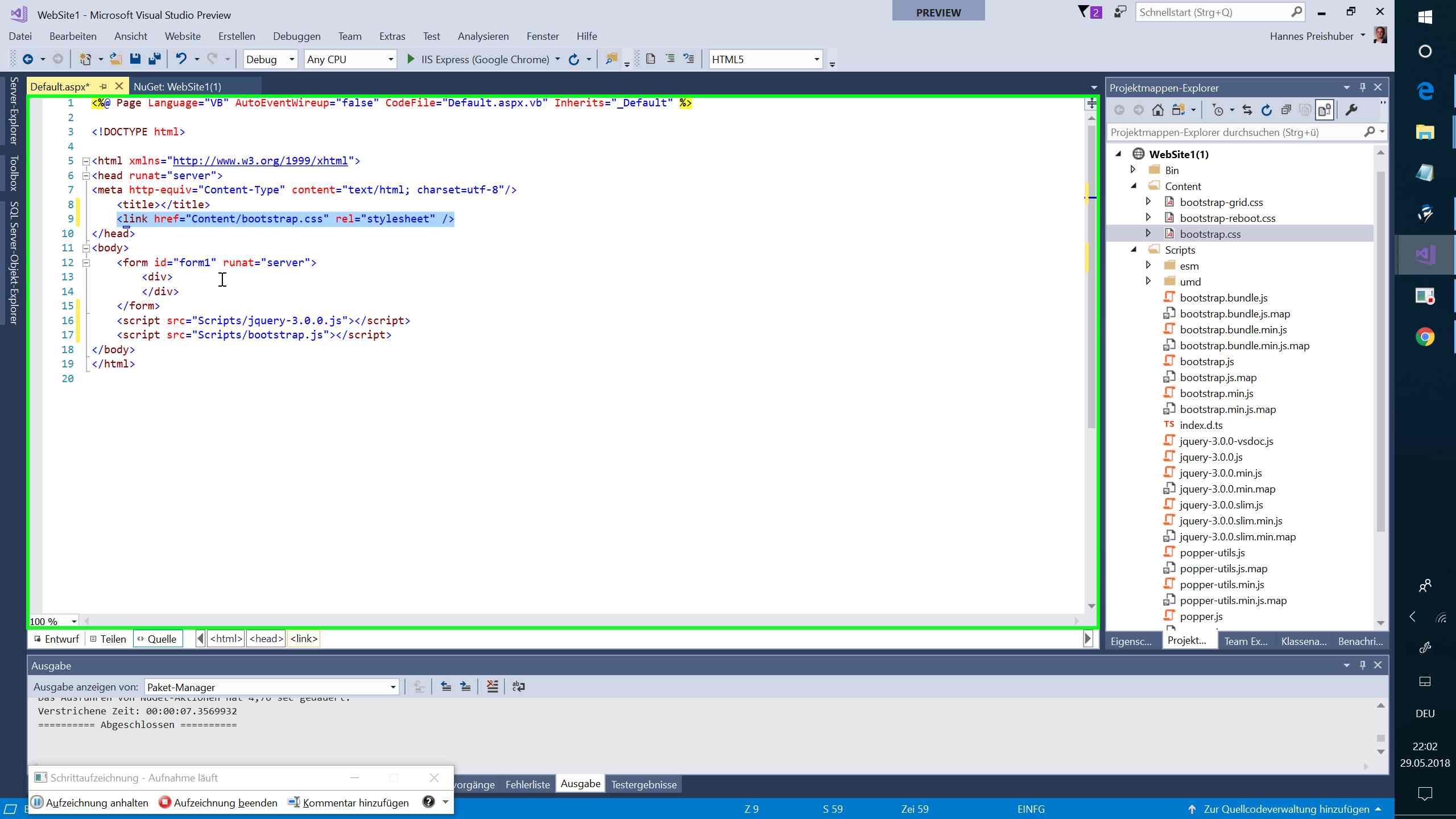


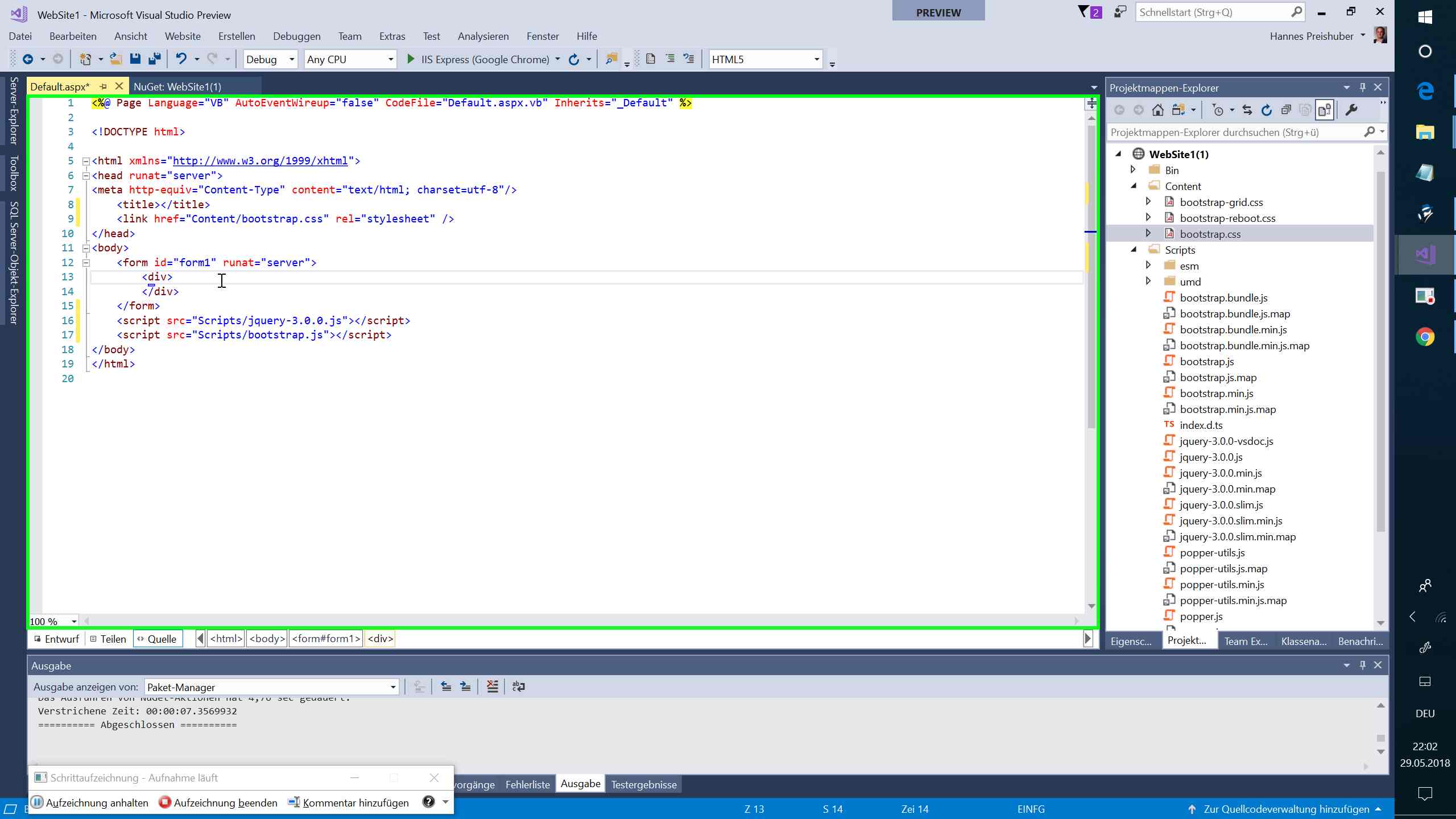
Ziehen Sie die Datei jquery-3.0.0 (abhängig von der aktuell durch Nuget installierten Version) ans Ende der ASPX Seite, nach dem schließenden Form Element.

Bildschirmfoto Schritt 38

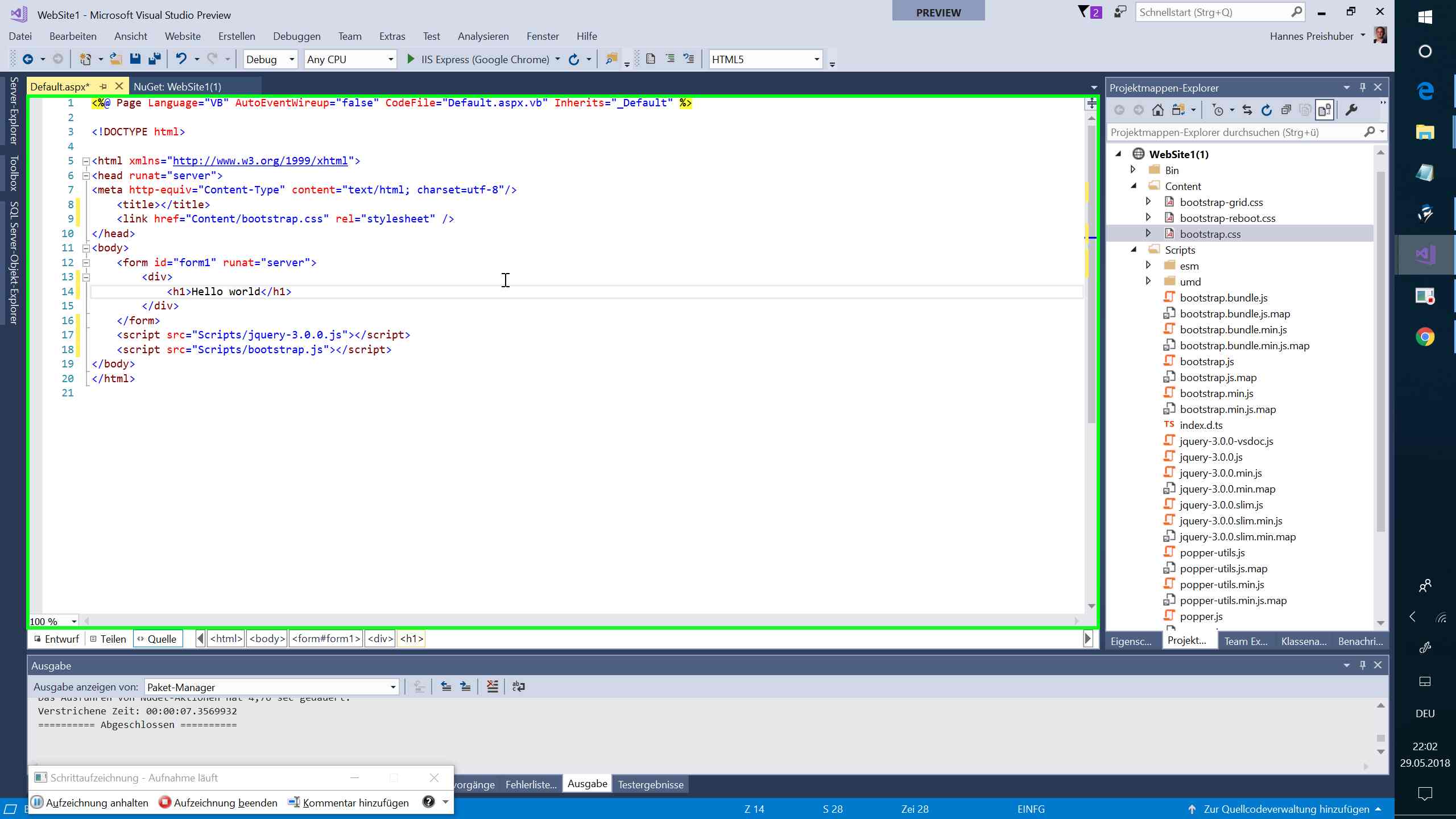
Ziehen Sie die Datei bootstrap.js (abhängig von der aktuell durch Nuget installierten Version) ans Ende der ASPX Seite, nach dem schließenden Form Element und nach der JQuery Referenz.

Ziehen Sie die Datei bootstrap.css (abhängig von der aktuell durch Nuget installierten Version) in den Header Bereich der ASPX Seite.

Bildschirmfoto Schritt 44

Bildschirmfoto Schritt 45

Ergänzen Sie den HTML Code in der Default.aspx Seite um ein H1 Element und Text.

Bildschirmfoto Schritt 46

Klicken Sie mit der linken Maustaste auf "In Browser anzeigen" um die Website im Browser zu starten.

Bildschirmfoto Schritt 47

Das Browserfenster erscheint und Sie sehen Ihre Begrüßung. Öffnen sie die F12 Browser Tools und versuchen Sie den Text blau zu färben.

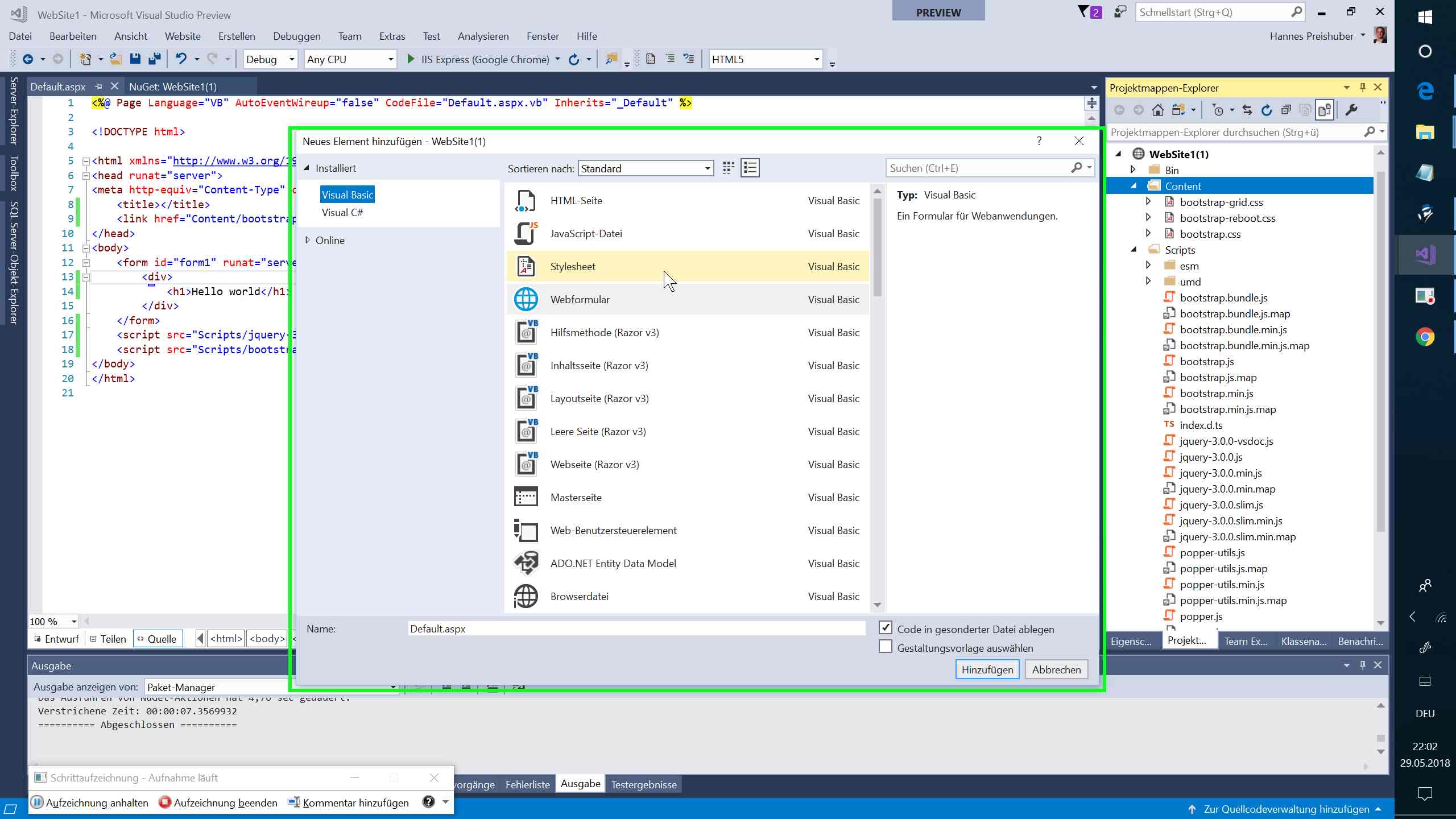
Bildschirmfoto Schritt 48

Dazu wählen Sie im Reiter "Elements" das Element h1 aus. Rechts suchen Sie das CSS Attribut h1 und klicken auf den Wert des color Eintrages. Tippen Sie dort „red“ und beobachten Sie im oberen Bereich die Veränderung der Darstellung.



Um diese Änderung dauerhaft zu speichern wird eine CSS Datei angelegt und dem Projekt hinzugefügt. So werden gegebenenfalls CSS Style Anweisungen aus Bootstrap überschrieben. Die letzte Anweisung gewinnt.

Klicken Sie mit rechter Maustaste auf Projekt und wählen Sie "Neues Element hinzufügen..." aus.  
Klicken Sie im Dialogfenster auf "Stylesheet".

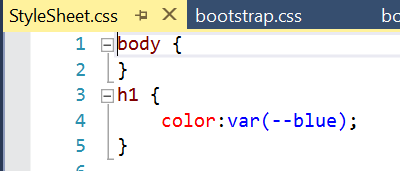
Bildschirmfoto Schritt 53

Es öffnet sich das Editor Fenster für die CSS Datei.

Ziehen Sie Datei in den Ordner Content.

Man kann Farben als Hex Code der RGB Kanäle #ff0000 oder als Name (red) angeben. Ein neues CSS Feature erlaubt es Variablen zu definieren. So kann eine Farbe an zentraler Stelle definiert und mehrfach verwendet werden. Suchen Sie in der Datei Bootstrap.css die entsprechende Stelle an denen die Farben definiert sind und benennen Sie die Regeln.

Ergänzen Sie in der neu erzeugten Datei Stylesheet.css die CSS Regel H1 und definieren Sie die Schriftfarbe (color) mit dem vordefinierten Farbschema blue.



Bildschirmfoto Schritt 57

Ziehen Sie mit der Maustaste die Datei StyleSheet.css in den Header Bereich der Datei Default.aspx unterhalb der Referenz auf Bootssrap.css

Speichern Sie die Datei.

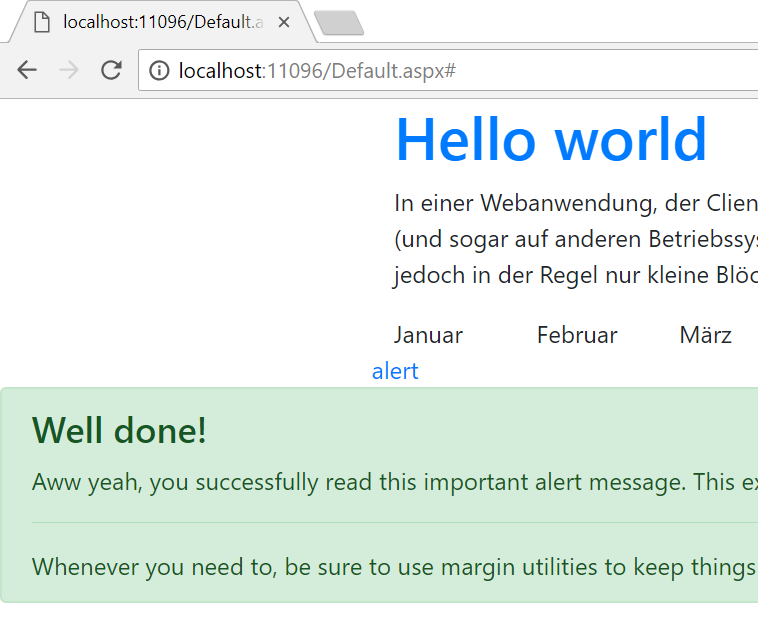
Bildschirmfoto Schritt 61

Klicken Sie auf das ASPX Dokument und wählen Sie "in Browser anzeigen".

Aufruf erfolgt über "localhost:42653/Default.aspx" in Google Chrome.

Die Schrift ist nun blau.

Als nächstes soll eine Website mit folgendem Layout erstellt werden. Achten Sie auf die Ränder links und rechts.



Kapseln sie den bestehenden HTML H1 und P in ein DIV mit dem Klassenattribut container und refreshen Sie die Website im Browser. Falls geschlossen wieder mit rechter Maustaste die Website im Browser öffnen.

<div class="container">

<h1>Hello world</h1>

<p>In einer Webanwendung, der Client (Browser) und Server können Sie verschiedene Programme, die häufig ausgeführt wird, auf unterschiedlichen Computern (und sogar auf anderen Betriebssystemen). Die beiden Hälften der Anwendung freigeben daher sehr wenig Informationen; Sie können die Kommunikation jedoch in der Regel nur kleine Blöcke von einfachen Informationen austauschen.</p>

</div>

Um den Überblick über die vielen geschachtelten DIV Elemente zu behalten, und damit auch korrekten HTML Code zu garantieren, gibt es drei Möglichkeiten.

Achten Sie auf grün unterwellte HTML Element im Visual Studio Editor

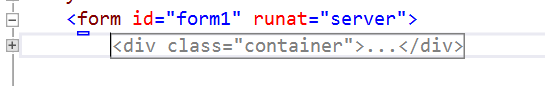


Tippen Sie ein einzelnes <DIV> und bewegen Sie nun den Mauszeiger auf das Element. Es wird ein Tooltip angezeigt. Löschen Sie das DIV wieder.

Zur Formatierung und automatischen Einrückung entsprechend der Hierarchien drücken Sie regelmäßig die Tastenkombination STRG K+D

Halten sie Die STRG (oder CTRL) Taste gedrückt und drücken Sie erst die K und dann die D Taste. Lassen Sie alle Tasten los. Das Dokument wird dann formatiert.

Bei korrektem HTML erscheinen im Editor Minus Symbole die es erlauben komplette Bereiche einzuklappen. Probieren Sie das aus!



Ergänzen Sie den HTML Code mit vom Server dynamisch generierten HTML. Es werden die Monatsnamen in zwölf Spalten ausgegeben.

<div class="container">

<h1>Hello world</h1>

<p>In einer Webanwendung, der Client (Browser) und Server können Sie verschiedene …</p>

<div class="row">

<% For i = 1 To 12 %>

<div class="col-md-1">

<%=System.Globalization.CultureInfo.CurrentCulture.DateTimeFormat.GetMonthName(i) %>

</div>

<%Next i %>

</div>

</div>

Starten Sie die Website im Browser. Verändern Sie Breite des Browserfensters. Wenn es schmal genug ist, werden die Monatsnamen untereinander dargestellt. Dies demonstriert die Responsive Design Fähigkeiten von Bootstrap.

## Bootstrap Alert

Am Ende des HTML Codes der Page, vor dem schliessenden </FORM> Element, wird der folgende HTML Code eingefügt. Die Texte in den P Elementen dürfen Sie natürlich ändern.

<div class="alert alert-success " role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

<h4 class="alert-heading">Well done!</h4>

<p>Aww yeah, you successfully read this important alert message. This example text is going to run a bit longer so that you can see how spacing within an alert works with this kind of content.</p>

<hr>

<p class="mb-0">Whenever you need to, be sure to use margin utilities to keep things nice and tidy.</p>

</div>

Starten Sie nun die Default.aspx Datei wieder im Browser per Rechtsklick. Es erscheint eine grüne Box im Browser, die die komplette Breite einnimmt. Der Benutzer kann diese schließen in dem er auf das x rechts klickt. Das Bild dazu findet sich vorher.

Ergänzen Sie nun den HTML Code in der ASPX Seite. Das DIV des Alert Templates wird um ein Style Attribut fade ergänzt. Damit ist das Alert nun beim Laden der Seite im Browser nicht mehr sichtbar.

<div class="alert alert-success fade " role="alert">

Als nächstes wird ein Hyperlink eingefügt der per Javascript das Alert animiert einblenden kann, wenn der Benutzer klickt.

Folgender Code-Schnipsel wird in das bestehende HTML Dokument nach der Monatsauflistung eingefügt. Die DIV Class Row Elemente müssen sich auf der gleichen hierarchischen Ebene befinden. Da es kein eingebettetes DIV mit der Klasse col- gibt, erscheint in der Browseransicht der Hyperlink „alert“ nicht genau unterhalb der Monatsansicht. Dies kann man im Bild erkennen, bzw. müssen Sie selbst beobachten, wenn Sie die Seite im Browser aus Visual Studio öffnen.

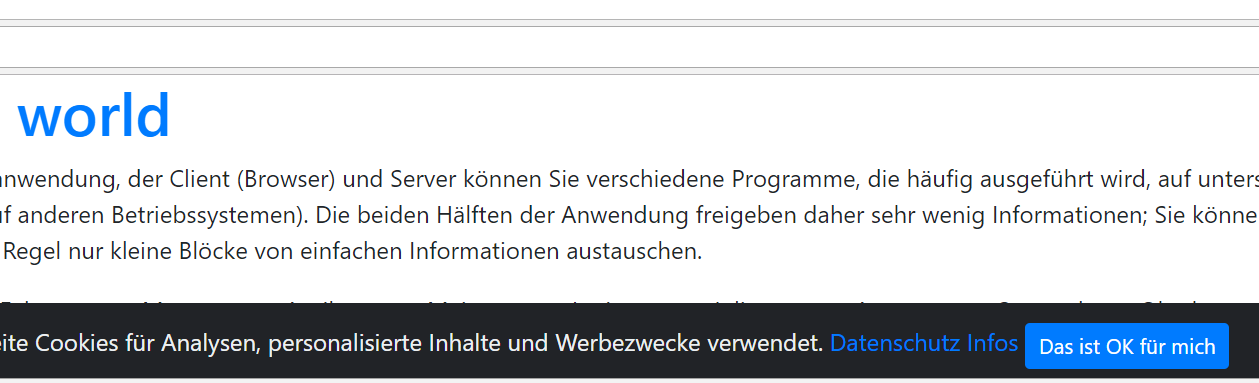
<div class="row">

<a href="#" onclick="$('.alert').removeClass('fade').hide().fadeIn('slow')">alert</a>

</div>

## Cookie Warnung nach DSGVO

Im nächsten Abschnitt wird eine Cookie Meldung angezeigt. Wenn der Benutzer diese bestätigt wird diese nicht mehr eingeblendet.



Ergänzen sie das Stylesheet.CSS um folgende CSS Klasse und speichern Sie die Datei.

.cookiealert {

position: fixed;

bottom: 50px;

left: 0px;

width: 100%;

margin: 0;

z-index: 999;

border-radius: 0;

background: #212327;

transform: translateY(100%);

transition: all 500ms ease-out;

color: #ecf0f1;

}

Als nächstes fügen Sie in der Default.aspx Seite unten vor dem schließenden Form Element ein ASP.NET Steuerelement ein. Am effizientesten geschieht dies über drag und drop aus der Werkzeugleiste von Visual Studio. In das Placeholder Element fügen Sie ein Button Steuerelement ein.

<asp:PlaceHolder ID="CookieMeldung" runat="server">

<div class="alert cookiealert " role="alert">

<b>Cookie Hinweis-</b>

Ich stimme zu, dass diese Seite Cookies für Analysen, personalisierte Inhalte und Werbezwecke verwendet.

<a href="https://ppedv.de/microsoftexperte/datenschutz.aspx" target="\_blank">Datenschutz Infos</a>

<asp:Button ID="CookieBtn" runat="server" Text="Das ist OK für mich"

CssClass="btn btn-primary btn-sm" OnClick="CookieBtn\_Click" />

</div>

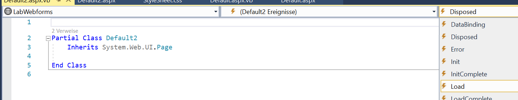
</asp:PlaceHolder>

Um im Code Behind der ASPX Seite die Events für das Button-Click und das Page-Load Event einzufügen, können Sie die Wörter eintippen oder sich von der IDE helfen lassen.

In der ASPX Seite wird das Attribut onclick getippt.. Darauf erscheint ein Popup, in dem ein neues Ereignis (Event) erstellt werden kann.



Wechseln Sie die Ansicht durch Rechtsklick in der ASPX Seite und Klick auf "Code Ansicht". In der oberen Leiste erlauben 3 Drop Down Listen, Objekt und Event auszuwählen



In der Code Behind Datei erscheinen die beiden Events, mit je zwei Parametern. Ergänzen Sie den Code wie im folgenden Listing. Damit wird ein Cookie mit dem Namen AC erzeugt, an dem erkannt wird ob der Benutzer zugestimmt hat, oder nicht. Der Platzhalter, samt beinhalteten HTML Code für das Alert Popup, wird so ausgeblendet.

Private Sub \_Default\_Load(sender As Object, e As EventArgs) Handles Me.Load

CookieMeldung.Visible = IsNothing(Request.Cookies("AC"))

End Sub

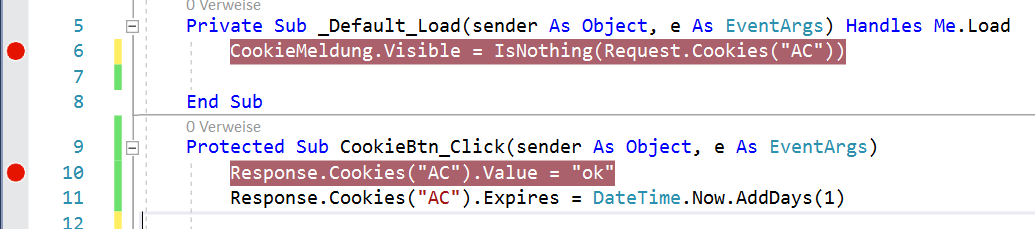
Protected Sub CookieBtn\_Click(sender As Object, e As EventArgs)

Response.Cookies("AC").Value = "ok"

Response.Cookies("AC").Expires = DateTime.Now.AddDays(1)

End Sub

Starten Sie das Beispiel. Im Projekt Explorer klicken Sie auf die ASPX Seite und wählen Sie „im Browser anzeigen“. Das Beispiel zeigt dauerhaft die Alert Nachricht an. Wir haben einen Fehler. Um diesen zu finden, kommt der Visual Studio Debugger zum Einsatz. Setzen Sie zwei sogenannte Breakpoints, indem sie auf der linken Seite in der grauen Leiste in den betreffenden Zeilen klicken.



Drücken Sie F5 und starten Sie die Website im Debugger. Die Ausführung stoppt an Zeile 6 , diese wird gelb markiert. Drücken Sie noch einmal F5 erreichen Sie den nächsten Breakpoint in Zeile 10. Hier wird deutlich, dass die Sichtbarkeit des Placeholder Steuerelements gesetzt wird, bevor der Cookie gesetzt wurde. Das Page Load Event wird zuerst ausgelöst.

Ergänzen Sie nach Zeile 11

CookieMeldung.Visible = False

Nun muss das Beispiel funktionieren und die Alert Nachricht dauerhaft verschwinden.

Die F12 Browser Tools helfen Cookies zu sichten, zu verändern und zu löschen um Szenarien zu testen. Wählen Sie den Reiter Application in Chrome und wählen Sie in Storage den Punkt Cookies. Entfernen Sie das Cookie AC aus der Liste rechts und probieren Sie die Website noch einmal per refresh aus.

