


February 8, 2022

ASP.NET MVC Interview Assignment



AstreaX Inc.
4400 N Scottsdale Rd, Ste. 9-316
Scottsdale, Arizona 85251
www.AstreaX.com

Table of Contents

Table of Contents.....	2
 Summary	3
Part 1.....	3
Part 2.....	4

Summary

Below is a basic ASP.Net MVC assignment that we have our candidates try. When you are complete, you can zip up the project directory and send it to me for our Architects to review. Please do not hesitate to contact me with any questions that you may have.

Part 1

Create a simple controller and view that displays a page with a title, a header section for customer data and a hardcoded html table that lists 3 rows with fake hard-coded data.

1. Create a [ASP.NET](#) Core Web App (Model-View-Controller) application with no authentication.
2. Add a Customer controller
This controller should have an action called Orders. This should be accessible at <http://localhost:portnumber/customer/orders>. If you click the Play/Run button in visual studio this will start the web server and the project and will launch your default browser going to the site.
3. Add a view page (.cshtml) for the Orders action method created in step 2. It should utilize bootstrap for styling and look like the following.

AstreaX.MVCInterviewAssignment [Home](#) [Privacy](#)

Orders

Name	Customer #
John Smith	111111

Order Id	Order Date	Description	Total
Test	Test	Test	Test
Test	Test	Test	Test
Test	Test	Test	Test

© 2021 - AstreaX.MVCInterviewAssignment - [Privacy](#)

NOTE: Bootstrap will already be added to your project when you create it and will be available for you to use.

- The contents of the view should be inside a bootstrap container.
- It should have the title of Orders

- Use the grid system (<http://getbootstrap.com/css/#grid>) for spacing in the header portion, i.e., Name and Customer#
- And use the bootstrap table classes (<http://getbootstrap.com/css/#tables>) to style the table. For now, the data in the table will be hardcoded.

Part 2

Let's continue with the customer orders sample application ...

1. In the /models folder, create 2 classes. CustomerOrder and CustomerViewModel.
2. CustomerOrder will have the following properties:
 - a. OrderId : int
 - b. OrderDate : DateTime
 - c. Description : string
 - d. Total : decimal
3. CustomerViewModel will have the following properties:
 - a. CustomerNumber : int
 - b. FirstName : string
 - c. LastName : string
 - d. FullName : string, get only property that is a concatenation first name a space and then last name
 - e. Orders : list of CustomerOrder.
4. In the CustomerController's Orders action method, instantiate the view model set the properties and then add 3 orders to the customer.
5. Pass the model into the call to View(). This will send the model to the view.
6. At the top of the view page, set the model for the view to be of type CustomerViewModel.
7. Use razor syntax to replace the hard-coded john smith and customer number to be the values from the model.
8. Replace the data rows in the table and use razor syntax with a foreach to loop through the customer view model's list of Orders and create rows in the table for each Order.
9. For the Total field use a format string to display the value as currency using the .ToString() method.

It should look like the following:

Orders

Name

John Smith

Customer #

111111

Order Id	Order Date	Description	Total
1	5/21/2021 8:50:37 AM	Part 1	\$10.00
2	5/21/2021 8:50:37 AM	Part 2	\$20.00
3	5/21/2021 8:50:37 AM	Part 3	\$30.00