

Assignment 3

Due Date: Thursday, April 22nd, 11:59 PM

Weight: 15%

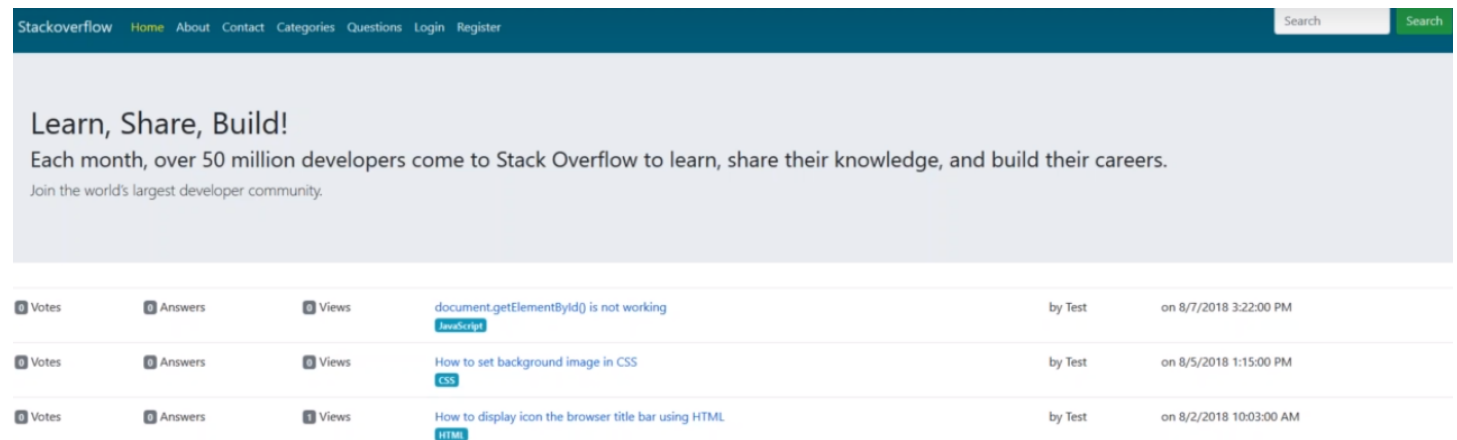
Objective Build a question/answer discussion board similar to stackoverflow using MVC ASP.NET

In this assignment you will need to build an ASP.NET MVC Application to mimic a multi-user question and answer forum; similar to **stackoverflow** with the ability to answer questions

Note: For this assignment, **it is expected that security/user registration and authentication are implemented**

Main page

Below is a sample mock up of the home page. **Note how for each question, you are keeping track of the views as well as the number of votes**



At the very top, the following links should be available for the user:

- **Home:** Takes you to the home page (i.e. The page displayed below) of the application
- **About :** provides static content to the user about this web site
- **Contact:** provide static content to the user on how they can contact you
- **Categories** navigates you to the categories page where you will see a list of the following categories dynamically retrieved from the database



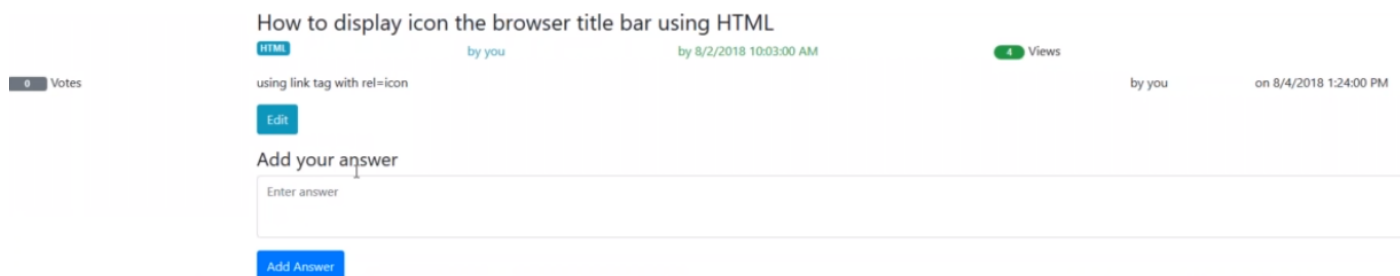
- **Question** Navigates the user to a list of questions that have already been asked dynamically retrieved from the database. In addition to the columns displayed below, the date the question was asked must be displayed as well:

0 Votes	0 Answers	0 Views	document.getElementById() is not working JavaScript
0 Votes	0 Answers	0 Views	How to set background image in CSS CSS
0 Votes	0 Answers	2 Views	How to display icon the browser title bar using HTML HTML

Clicking on each question takes you to the page for that question where you can add your answer.

In addition, your implementation should support:

- The ability to edit an answer
- Displaying who the answer was added by
- Displaying the Views count



Controllers:

For this lab, you will have the following controllers:

HomeController: should contain at least the following controller methods

- **Index:** displays list of top 10 questions
- **About** displays the About page
- **Contact** displays the Contact page

- **Categories** should display the top 10 categories
- **Questions** should display the top 10 questions

Question (Some of the methods - but not necessarily all. You may find that you will need others depending on your implementation)

- InsertQuestion(Question q);
- GetQuestions();
- GetQuestionByQuestionID(int QuestionID)
- AddAnswer(): **Pass the necessary parameter(s)**

Models

You will need the following models:

- Question which contains:
 - Question Id
 - QuestionName
 - QuestionDateAndTime { get; set; }
 - Category Id
 - Category object reference
- Category
 - Category Id
 - Category Name
- Answer
 - Answer Id
 - Question Id
 - Answer Text
 - Answer date and time

And any other attributes that you may deem necessary

Database

In order to support your application, you will need to have a database behind it.

Your database will contain **at least** the following tables

- **Categories**
- **Questions:**
- **Answers**

Additional Notes

- Some implementation details have been intentionally left out to provoke your thinking and leave the door open for creativity on your part.
- All pages share the header shown below.



Submission:

Submission of must consist of :

- a zip file of your source code :
- A video recording with audio demonstrating your application - going through the various functionalities.

Your zip file must be named: <YOUR_LASTNAME>_A3.zip

Your video file must follow the same naming convention.

Failure to comply with the naming convention will result in mark deduction.