University of Canberra

Faculty of Science and Technology

**Web Design and Programming (7175 & 6691)**

**Assignment 2**

**Submission date**: 23:59 Sunday 01/11/2020 (Week 13)

**Type**:Individual assignment

**Total mark**:20

**Minimum mark to pass this assignment:** 4

**Submission**:Submit via Canvas the following

1. A .zip file that contains all files in your **web application project** and **database files**.
2. **Usernames** and **passwords** for web manager and registered user.

**Late submission**: 5% of the total mark (i.e., 1 mark) per day.

**Tasks**: Design and implement a responsive website for Data Science and Machine Learning that is similar to the website in Assignment 1. Online users can rate and post feedback after they are logged in, and a website manager can log in and edit or delete any current posts.

**Requirements**:

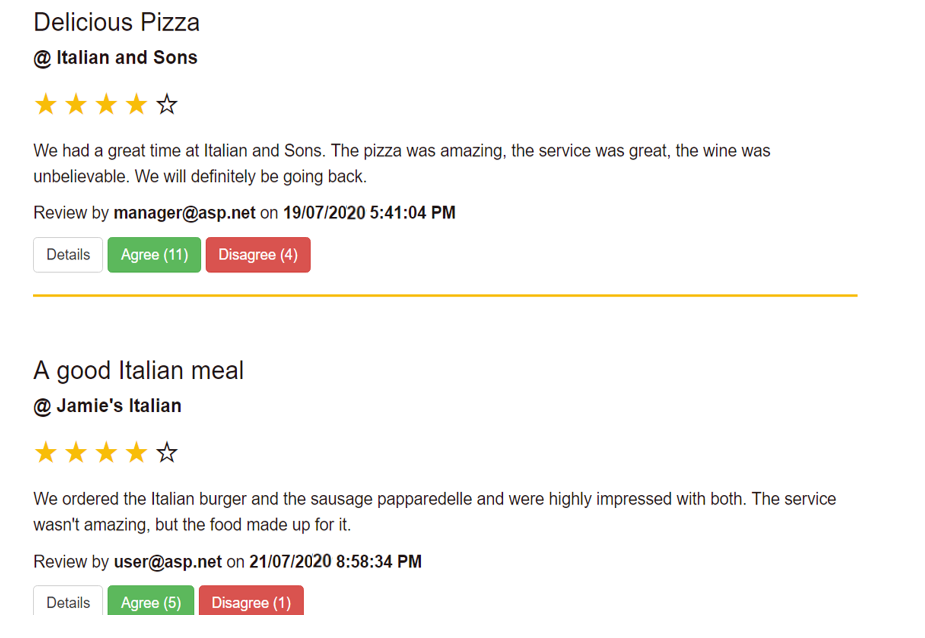
1. **Template**: **ASP.NET Core Web Application** in Visual Studio 2019. Language: **C#** (-20 marks if these template and language are not used).
2. **Website**: full-width layout, responsive design, and having the following pages: **Home** page, **Restaurants** page, **Cuisines** page, **Dishes** page, **Prices** page, **Locations** page, **Reviews** page and **Contact & About** page for contact and about this website. Links to those pages are available as menu items on the main menu of the website.
3. The new web pages for reviews (**Customer Reviews** page (**Index.cshtml**) and pages for creating, editing, details, deleting review posts in folder for customer reviews pages) are created using **MVC Entity Framework** (-20 marks if MVC Entity Framework is not used) and have the design as seen in the screenshots below.
4. User can read all review details posted on **Restaurants** page without login.
5. User is required to log in before posting review on **Restaurants** page.
6. User cannot edit or delete his/her review post after it is posted.
7. Website manager can edit or delete any review post on **Restaurants** page or **Customer Reviews** page after he/she is logged in (you can manually assign manager role to a registered user in the current database of this website).
8. A review post contains the following: date (when it is posted), name (who posted it, use the current username from login), heading, star rating, and review content. The user *does not need to enter date and name*, instead you implement controller methods to display and save date and name to the current database when the post is submitted.
9. There are 2 additional buttons (Agree and Disagree) for each review post and the user can increase the current number for agree or disagree after he/she is logged in.
10. All review details (date, name, heading, star rating, review content, and current numbers for agree and disagree) of all posts have to be stored in the database created using MVC Entity Framework (-10 marks if all details are not saved in the database).
11. Handle all exceptions that may occur at runtime (– 1 mark for each exception found).

No report is required. More details and help will be provided in tutorials and lectures.

Marks for each item are given below. The full marking guideline is on the last page.

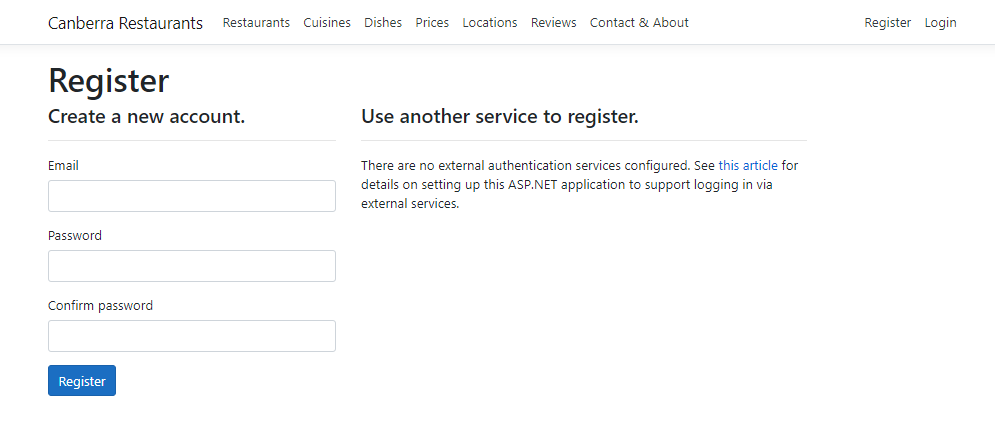
1. [6 marks] If the user clicks **Reviews** on the navigation bar, the **Customer Reviews** page will appear with the design below. Note details of those customer reviews below will be added later from other pages. **Note**: There is no customer review posts when you start creating this page.



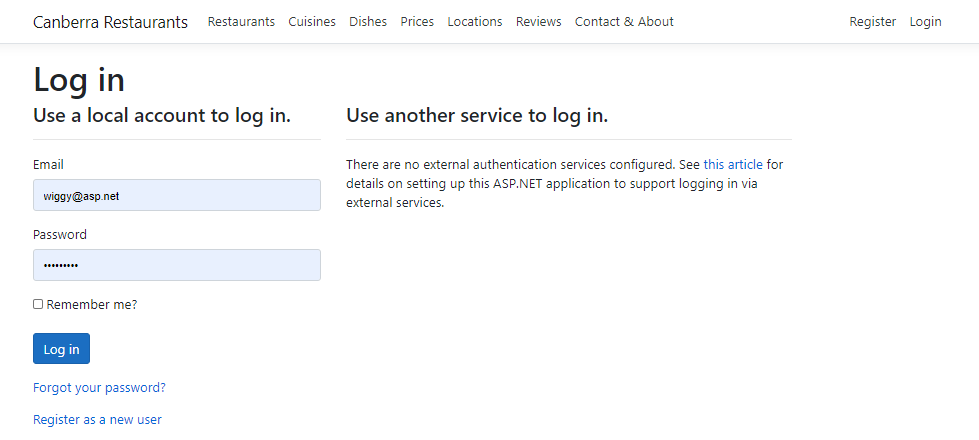


These customer reviews will be added automatically from the Create page (details how to create are listed below)

1. [0.25 mark] On this **Customer Reviews** page, if the user clicks on **create an account**, the following page will appear, and the user can register. The register data has to be saved to the database.



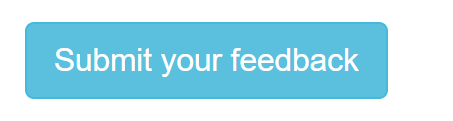
1. [0.25 mark] On this **Customer Reviews** page, if the user clicks on **log in**, the following page will appear, and the user can enter username and password to log in (the user has to create an account first):



1. [0.5 mark] After the user is logged in (either website manager or registered user), the user will be redirected to this **Customer Reviews** page, and will not see



Instead the user will see a blue button

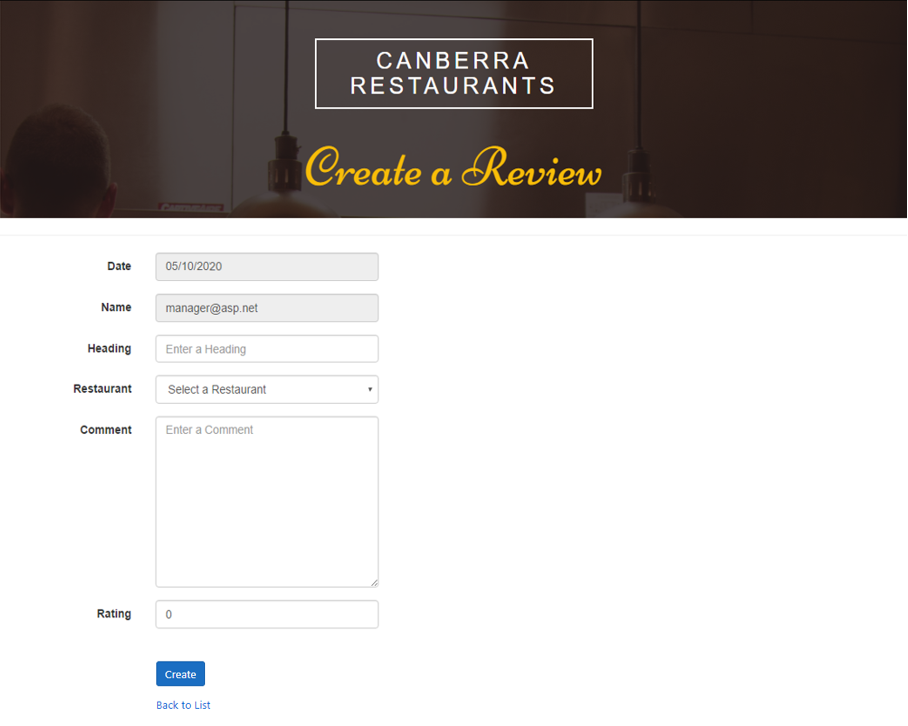


on this **Customer Reviews** page as follows (other content and layout on this page remain the same)

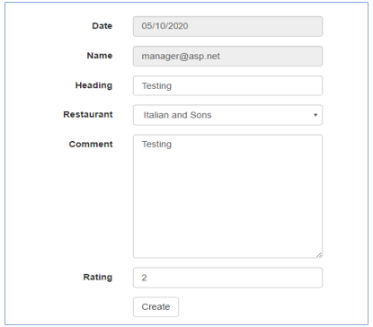
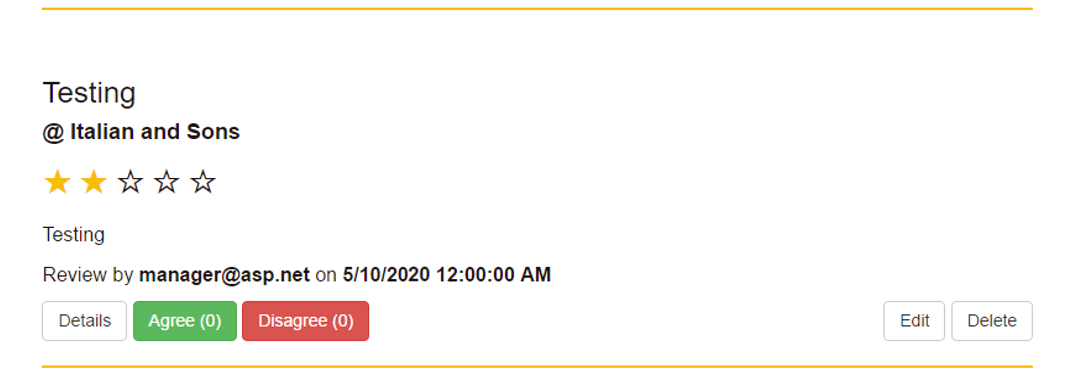


1. [1 mark] If the user clicks on this **Submit a Review** button, the user will see the **Create a Review** page below. Note that
   1. **Date**: the current date (when the user opens this page)
   2. **Name**: the username (for login),
   3. **Other details**: as seen in the screenshot below

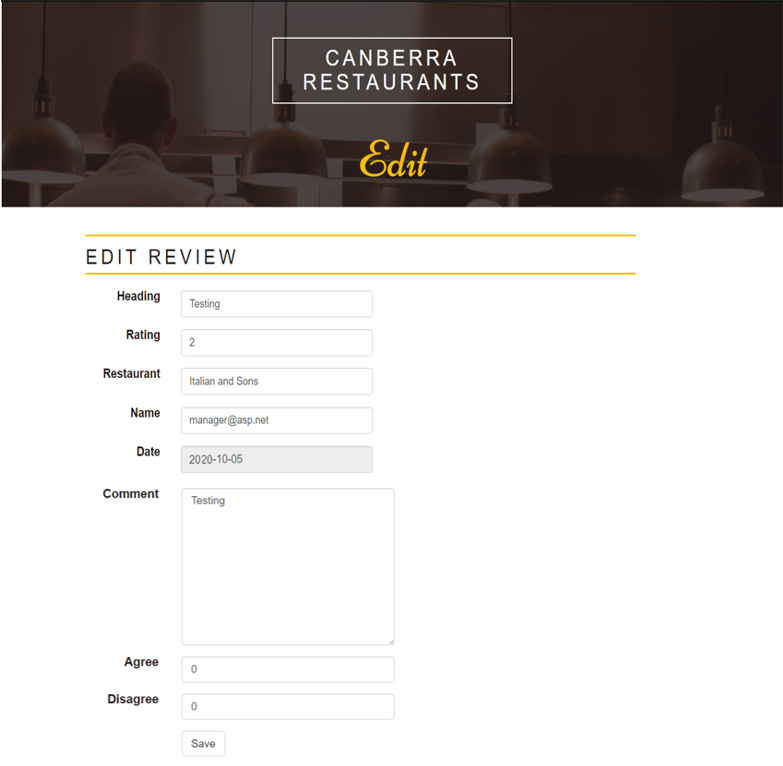
The user will use this page to create a new customer review as follows: the user selects a restaurant from the dropdown list (there are at least 6 restaurants you have implemented in Assignment 1), enter a heading, enter a comment and change the number for rating from 0 to 1, 2, 3, 4 or 5 (your website needs to handle exceptions if the user does not enter a number between 0 and 5). After the user clicks the **Create** button, the user will be redirected to the **Customer Reviews** page and will see the new customer review posted in the list. The number entered for star rating will be the number of yellow stars seen on the page.



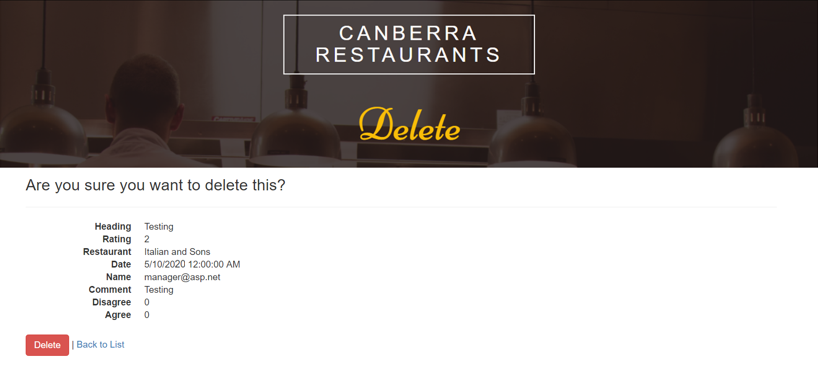
On the **Customer Reviews** page:

1. [0.5 mark] If the user is **website manager**, after logged in the user will see two extra buttons (**Edit** and **Delete**) appeared on every customer review (see the screenshot above). However, if the user is not website manager (i.e. just a **registered user**), the user will not see these two buttons.
2. [1 mark] If the website manager clicks on the **Edit** button, the manager will see the **Edit** page as below. The manager edits all details except the date (greyed out, not editable) then clicks the **Save** button and be back to the **Customer Reviews** page. The manager will see the updated review on this page. Note: the date appeared on this Edit page is the current date at which the Edit form is opened.



1. [1 mark] If the website manager clicks on the **Delete** button of a review, the **Delete** page will appear as seen below with all details of that review



After clicking the **Delete** button, the manager will be redirected to the **Customer Reviews** page and will not see that review (it has been removed from the list).

Note: For items 9 and 10 below, the registered user can only change the number for **either Agree or Disagree (cannot change both)** and **only once in the current session** (the registered user needs to logoff to stop the current session then login again to start a new session if the user wants to change the number for either Agree or Disagree again).

1. [0.75 mark] If the logged in user clicks on **Agree**, it will add 1 to the next number. For example **Agree(11)** becomes **Agree(12)** after clicked.
2. [0.75 mark] If the logged in user clicks on **Disagree**, it will add 1 to the next number. For example **Disagree(4)** becomes **Disagree(5)** after clicked.
3. [1 mark] If the logged in user clicks on **Details** (for example in the first review for **Delicious Pizza**), the following page will appear





If the logged user clicks on **Back to List** link, the logged in user will be redirected to the **Customer Reviews** page.

[1.5 mark] Adding the **list of customer reviews** to **Restaurants page**

The area for big picture on this page in Assignment 1 will be removed and the list of customer reviews will be added. **Note: the list of customer reviews needs to be sorted by dates, newest to oldest one.**



**Marking Guideline (max 20 marks)**

* [4 marks] Adding Home page, Restaurants page, Cuisines page, Dishes page, Prices page, Locations page, and Contact & About page from Assignment 1. HTML, CSS, JavaScript and jQuery need to be in the right files.
* [6 marks] Customer Reviews page
  + [1 mark] Background, headers, Be the Critic text and 2 square images
  + [0.5 mark] Design of the list of reviews (full width layout)
  + [3.5 marks] Design and implementation of a review (must be the same design with that seen on page 2), Controller methods and Model class
  + [0.5 mark] Register and login links (seen on page 3)
  + [0.5 mark] Change ‘Please login …’ to Submit a Review button
* [1 mark] Design of **Create a Review** page (seen on pages 4 & 5)
* [0.5 mark] Show/hide **Edit** and **Delete** button on **Customer Reviews** page depending on user role
* [1 mark] Design of **Edit** page
* [1 mark] Design of **Delete** page
* [1.5 marks] Implementation of **Agree** and **Disagree** buttons and methods
* [1 mark] Design of **Details** page
* [1.5 marks] Adding and sorting (by date) list of reviews to **Restaurants** page
* [2 marks] Manager role and Registered User roles are correctly implemented
* [0.5 mark] Main menu and Footer
* [1 mark] Overall
* [– 20 marks] Template for ASP.NET Core Web Application not used
* [– 20 marks] MVC Entity Framework not used
* [– 10 marks] If all details of customer review posts not saved in the database
* [– 1 mark] For each exception not handled

Lectures on Thursdays and Tutorials/Labs from Week 2 to Week 7 will help you design and implement this website.

Your assignment will be marked by your tutor. Please contact your tutor for further information on your mark.

Visit Virtual Room and Discussion forum on the Canvas site of this unit for more information posted by lecturer, tutors and other students.

Assignment extension: details in Unit Outline on Canvas.

-- END --