300583 Web Systems Development

PRACTICAL SET 2 DUE: 9PM 25 SEP 2020

In Practical Set 2, you will continue to work on the 'Gourmet Pizza' shop based on what you have done in Prac1. Before proceeding with this practical set, you should back up your work for Prac1. Your backup of Prac1 may be needed in case you have marking disputes with us.

Steps for completing this practical set

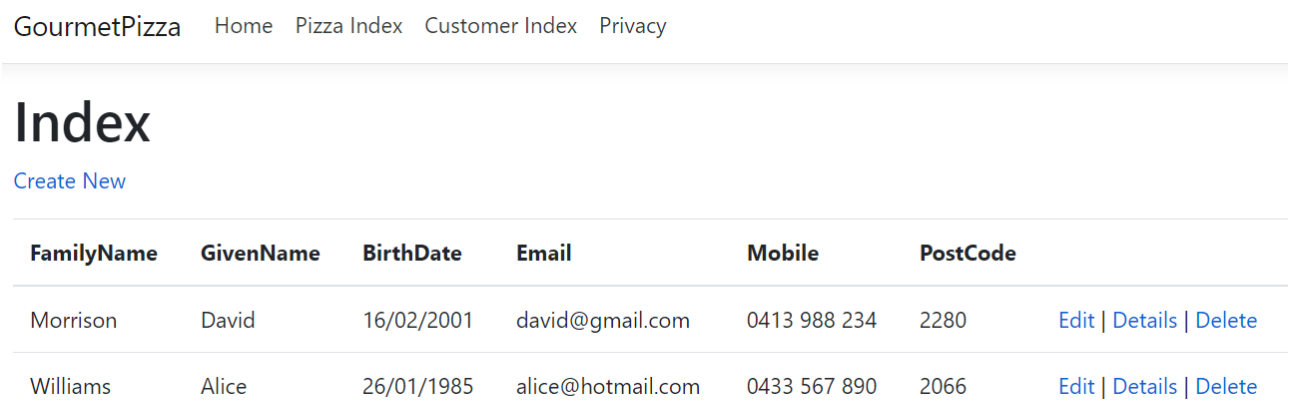
1. Define a new Model class called 'Customer' for describing the customers of this pizza shop. Suppose the shop needs to record the following pieces of information about a customer: family name, given name, date of birth, email address, mobile number and postal code.
    - The class you define must have the properties for all pieces of info above and an ID property to uniquely identify a customer.
    - For the mobile number and the postal code, you should use string type, which allows '0' to appear in the beginning and also makes the later validation easier.
    - All property names are of your choice.
2. Perform scaffolding to generate Razor pages for CRUD operations on 'Customer' table in the database. You should use the existing database context class for database connection (NB: this one was created when you scaffold the Pizza class in Prac1).

    Hint: refer to both Lecture 4 and Lecture 5 slides.
3. Migrate the Customer class to database.
4. Modify the "_Layout.cshtml" file to add a "Customer Index" hyperlink to the navigation bar. This hyperlink should point to the Index page for the customers. (hint: refer to our sample project accompanying Lecture 4).
5. Run the GourmetPizza project and Click the "Customer Index" link in the navigation bar. Use the web interface obtained to add the following data into the Customer table in the database.

Family Name	Given Name	Date of Birth	Email Address	Mobile Number	Postal Code
Morrison	David	16/08/1955	david@gmail.com	0413992233	2171
Williams	Alice	03/05/1997	alice@hotmail.com	0442889921	2150

Table 1: Initial Customers Details

At the end of this step, after clicking the "Customer Index" link, the following output should be roughly displayed in the browser:



6. Add displaying annotations to the Pizza class to achieve the following:
    a. The heading for the price column should be displayed as 'Price Each'.
    b. The price field should show a '$' sign before the price in the Index page and Details page, but not in Create page and Edit page.

       Hint: Consider DataType.Currency.
7. Add validation annotations to the Pizza class to achieve the following:
    a. The pizza name field is required, and can only consists of English letters, digits, spaces and underscore, and has a length between 3 characters and 20 characters inclusive.
    b. The pizza price should be in the range from $5.00 to $20.00 inclusive.
8. Add displaying annotations to the Customer class to achieve the following:
    a. The heading for the Date of Birth column should be displayed as 'Date of Birth'.
    b. The Date of Birth field should only display the date part (i.e., exclude the time part).
9. Add validation annotations to the Customer class to achieve the following:
    a. The family name field is required, and can only consists of English letters, hyphen and apostrophe, and has a length between 2 characters and 20 characters inclusive.
    b. The given name: (same as above).
    c. The Email Address field is required and in a valid email format.
    d. The Mobile Number field is required and in the format of 04xx xxx xxx (i.e., there should be a space in the fifth character and in the ninth character). The error message for this field must fully advise users of the mobile number format required by this field, and be of your own words.

       Hint: a space can simply be represented by a space in Regex.
    e. The Postal Code field is required and consists of exactly 4 digits; and its first digit cannot be "9" for residential use (see https://en.wikipedia.org/wiki/Postcodes_in_Australia). The error message for this field must fully advise users of the postal code format required by this field, and be of your own words.
10. Follow the basic steps in Lecture 7 slides to add a search box in the Customers/Index page. This search box should allow us to search customers by checking whether their family names or given names contain the input string.

    Hint: The lambda expression inside the Where() method can use the "||" operator to indicate the 'or' relationship.

11. Follow the basic steps in Lecture 7 slides to implement a pizza order page. Specifically, this page should function as follows:
    a. There is a link named "Order Pizza" in the navigation bar.
    b. After clicking this link, a pizza ordering web form should be presented to users. This web form should contain the following input devices:
        i. A dropdown list of pizza names. This list of pizzas should be retrieved from the database.
        ii. An input box for the number of pizzas with the above name to order. The input for this box is required, and the number entered must be between 1 and 10.
        iii. An input box for a credit card number. The input for this box is required, and must exactly contain 16 digits. In the Model definition, it is the best to declare that the property for the credit card number has a data type of 'string'.
    c. Upon the successful submission of this form, an acknowledgement web page should be displayed with the following information.
        i. The name of the pizza to order.
        ii. The numbers of the pizza to order.
        iii. The total price for the above order. In calculating this one, the price of the pizza with that name must be dynamically retrieved from the database.
        iv. The credit card number to charge.

Checkpoint (2 marks)

The Prac2 Checkpoint will occur in your registered practical class in week 7. At this checkpoint you should complete steps 1-5.

Marking criteria:

- Use Sqlite DB browser to open the 'PizzaShop.db' file. You should see the Customer table is correctly created and populated with data in Table 1. (1 mark)
- Click the "Customer Index" link in the navigation bar. You should see the Customer info in Table 1 is correctly displayed. (NB: No need to satisfy the displaying and validation annotations mentioned in later steps; 1 mark.)

NB: Any mistake in achieving a bullet point above will result in a zero mark for that bullet point. No decimal mark is available.

Final Submission (16 marks)

Please follow the steps below to submit your completed Prac2 to vUWS:

- Zip up the entire GourmetPizza folder and its sub-folders, which contain all your files for your project.
- Name the ZIP file *Surname_studentid*_Prac2.zip
- Submit the zip file:
    - In VUWS, go to 'Practicals'
    - Click on 'Prac2 Submission'; this link will be made available closer to the deadline.
    - Attach your zip file and submit

Marking Criteria: Please see the rubric for Prac2 in the **My Grades** page on vUWS.

Notes:

- You can submit the zip file multiple times until the due time.
- Work submitted after the due time will incur a 10% per day late penalty as per university policy **https://policies.westernsydney.edu.au/document/view.current.php?id=227**
- **Based on the policy, the 10% above means the 10% of the worth of the assessment, not your final score. So if the worth of the assessment is 10 marks and you scored 5 marks with a 5-day lateness, then you'll receive 0 mark.**