**Team Project Requirement – Submission 3**

Note: The screen samples illustrated in this document are to help you understand the requirements. You are free to design your interface.

**Part 1**

In this assignment, your team will continue building the OpenTable web app on top of your project submission 2 by applying the skills ‘session and cookie’ you learned in Chapter 9.

**Function requirement:**

1. **Reservation** - Building the function for regular users (clients) to make a reservation.

**Restaurant List Interface** – is the 1st interface for the user to make reservations. It consists of two parts: a filter and a table of restaurants. Filter includes Metropolises, Cuisine Styles, and Price Ranges. The filter also includes two buttons: Find Table and Clear. Initially, the ‘restaurant’ part displays all restaurants in the database. When the user clicks ‘Find Table’, the restaurant table matches the filtering criteria. This interface looks like what you’ve built for the administrator to manage the restaurants.

**Restaurant Detail interface** - For each restaurant listed in the table, provide a ‘Detail’ button. Upon clicking, another interface is rendered to display the details of the selected restaurant, including region, name, address, phone number, cuisine style, price range, and available time slots for reservation. Let’s assume each time slot is one hour, such as 11 am, 12 pm, 1 pm, and etc. The time slot options should be extracted from the ‘Open Hour’ set for the restaurant, such as 11 am – 21 pm, and divided by hour. Note that you should only show available time slots. This means that you need to query the ‘Reservation’ table to delete the time slots that have been allocated. You should also allow the user to input the number of people for the reservation and the reservation date. After the user finishes the input, the user can either click the buttons ‘Hold Table’ or ‘Cancel’. Clicking the ‘Hold Table’ button will place the pre-reservation into a shopping cart (feel free to use other font-awesome icons). Clicking the ‘Cancel’ button will return the user to the restaurant list page which retains the user’s initial filtering criteria.

|  |
| --- |
| Happy Feet Restaurant at 1299 Ogden Rd. Wheaton, IL 60006  American  $$ |
| Reservation Date: (an input box, or a drop-down showing a week starting from today) |
| Time: (a drop-down list or a list of buttons showing the available time options, such as [11 am] [12 am] [1 pm] [2 pm] |
| [Hold Table] [Cancel] |

**Shopping-Cart Interface** - Once a pre-reservation is placed, the user is able to click the shopping cart to check all pre-reservations. The user can delete a single reservation, or click‘Confirm Reservation’ which will persist all pre-reservations into a Reservation table (so become reservations), or go back ‘Home’ which returns to the restaurant list page with the user’s initial criteria retained.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Restaurant | Date | Time | People | Action |
| Happy Feet at 1299 Ogden Road in Chicago | 4/13/2025 | 12 pm | 6 | [Remove] |
| India Garden at 111 Monroe in Chicago | 4/13/2025 | 12 pm | 2 | [Remove] |
| [Confirm Reserve] [Home] | | | | |

**Retain filtering criteria** - Note that when the user navigates in the app, the user’s filtering criteria should be retained unless the user ‘clears’ the filter. The pre-reservation should also be returned until the user confirms the reservation or deletes it.

**Retain Pre-reservations** - The pre-reservation shall be retained in the shopping cart during the user’s navigation.

**Retain Pre-reservation after the session (Extra 5 bonus points)** – The pre-reservation shall stay for one week even after the user logs out.

All the actions should be confirmed with an acknowledgment. Form re-submission are properly handled.

**Technical Requirement:**

1. Use the **session** to store the user’s filtering criteria and pre-reservations.
2. You can place the ‘**Cart’ icon in \_UserLayout**. This link should show the badge with the number of pre-reservations in the cart if the cart is not empty. Or, you can reference the link for ‘favorites’ illustrated in the Chapter 9 example ‘NFLTeams’.
3. Create a model entity **Reservation** to store reservation-relevant information, such as date, time, number of guests, and reference to the restaurant object.
4. Use **TempData** to store all one-time status messages along with **PRG**. This message should also be placed in **\_UserLayout** (or \_Layout) so the message format and position stay consistent for all actions.
5. [Bonus points] Use Cookie to retain pre-reservation for the out-of-session time period (7 days). Ensure you just keep the minimum information in the cookie.
6. To handle the available time slots, you query the restaurant table for ‘OpenHour’ and query the reservation table for all reservations booked in a restaurant. You then divide the OpenHour to hour-based slots and remove those that have been booked. It is likely that the pre-reserved time slot is booked out by someone else when the user is trying to confirm the reservation, that is OK, and we don’t handle this scenario in our school project.

**Part 2**

This part is to practice your learned technique from Chapter 11 to create custom validations.

**Function requirement:**

1. In the main area (for regular users), you add a ‘register’ link in the \_UserLayout navigation pane. Upon clicking, display an “Account Registration” form for a user to create an account with OpenTable.

* UserName: lower- and upper-case letters and numbers 0 - 9. Use property-level validation and **regex**
* Date of Birth (DOB) : must be a past date and no more than 100 years old Use **server-side custom validation attribute.** If you add client-side validation for this feature, you get another **extra 5-point bonus**.
* Email: must not exist in the database. Use **remote validation**
* Phone Number digit and delimiter of “-“ or “.” only, such as 123.456.7890 or 123-456-7890. Use property-level validation and **regex**

1. Upon successful registration, add the ‘User’ to the table. In addition to the above required information, add a timestamp default to the time when the user account is created and add a UserType using an **enum** (with choices of Client and Admin). After the user info is added to the table, display a one-time confirmation message stating, “Your registration is completed. Welcome to Open Table!”.

If there is any validation error, please show a **model-level summary message** stating, “Please fix the error,” and also show the **property-level error messages** for the offending elements.

**Technical Requirement**

1. You must code the validation as required in Orange.
2. For remote validation, please ensure server-side validation is always present. In case the client-side validation is compromised, your application can always count on server-side validation to make your app robust. Make sure you use TempData to coordinate the validation on the client-end and server-end so the validation is not duplicated, as illustrated in the Chapter 11 project.
3. To test your client-side validation function as designed, ensure you see the error message with offending elements BEFORE you click the ‘submit’ button (this is where the students lose the points easily). To ensure your validation works on the server side, you can turn off the JavaScript in your browser, as I’ve shown in the class, to exam the validation error presence after clicking the button.
4. For the client-side validation on DOB [**Bonus Points - 5**], ensure you place the custom javascript in the wwwroot folder and include the jQuery libraries and custom javascript function in the inner view requiring them.

**Submission Requirement**

* Your screenshot should capture the major features you build in this version.
* Zipped source code (please zip at the project root level. Don’t contain multiple levels of OpenTable folders. If you don’t know the project’s root level, please email me.)
* No peer review this time (the ‘torture’ ends, but you have had the experience to boast about in your resume and story to tell in the interview!)