**Reservation System**

Anish Man Shrestha

[anish.shrestha237@gmail.com](mailto:anish.shrestha237@gmail.com)

**GitHub Link:** <https://github.com/Anish-Shrestha/Reservation-System-ASP.NET-MVC>

**To Run Project**

* Download project form above github link.
* Build the project using visual studio.
* Restore the database. (project done in code first entity framework, the tables are automatically generated even if there are no tables in newly created database)
* Change the connection string accordingly in **web.config** file.
* Exceptions are logged in log file in location “C:\ReservationSystem\_Logs\Logs.txt" which is also configurable in **web.config** file.
* Run the project. (**ReservationSystem.Web** is start up project)

**Environment**

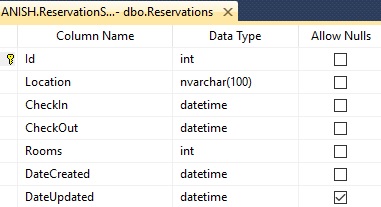
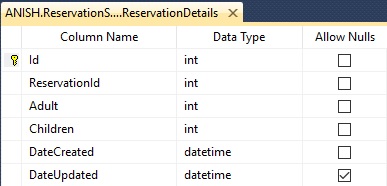
Visual Studio Community 2015, SQL Server 2012

**Summary**

1. This project is developed using **Entity Framework Code First development**, **Generic Repository Pattern**, Dependency Injection using **Ninject** and **Automapper** for binding view model with the domain model and vice versa.
2. The logging of the error, info is done using **Log4Net**.
3. In the **Model** section there is all of our domain objects. Entity Framework will count on them in order to build the database.
4. **Fluent API** is used to configure domain classes
5. There are two types of validation for Search-site form, **client side and server side validation**.
6. The libraries used in front-end part are

* **Bootstrap** – It is a responsive front-end framework used for layout design of view page and for various UI components.
* **TypeheadJs** – It is used for autocomplete of location using API provided (https://api.teleport.org/api/cities/?search=%QUERY).
* **Bootstrap-datepicker** – It is used for date-picker at Check-in and Check-out textbox.
* **ValidationJs** – It is used for client side validation of the search-site form.

**Database Tables**

** **

1. **Reservations**

It stores the information like Location, Check-in date, Check-out date and Number of rooms. Detail information of room is stored in **RerservationDetails** table.

1. **ReservationDetails**

It stores the information of number of adults, and children in the particular room. It contains the column **ReservationId** which is responsible for binding Reservations table with the **ReservationDetails** table.

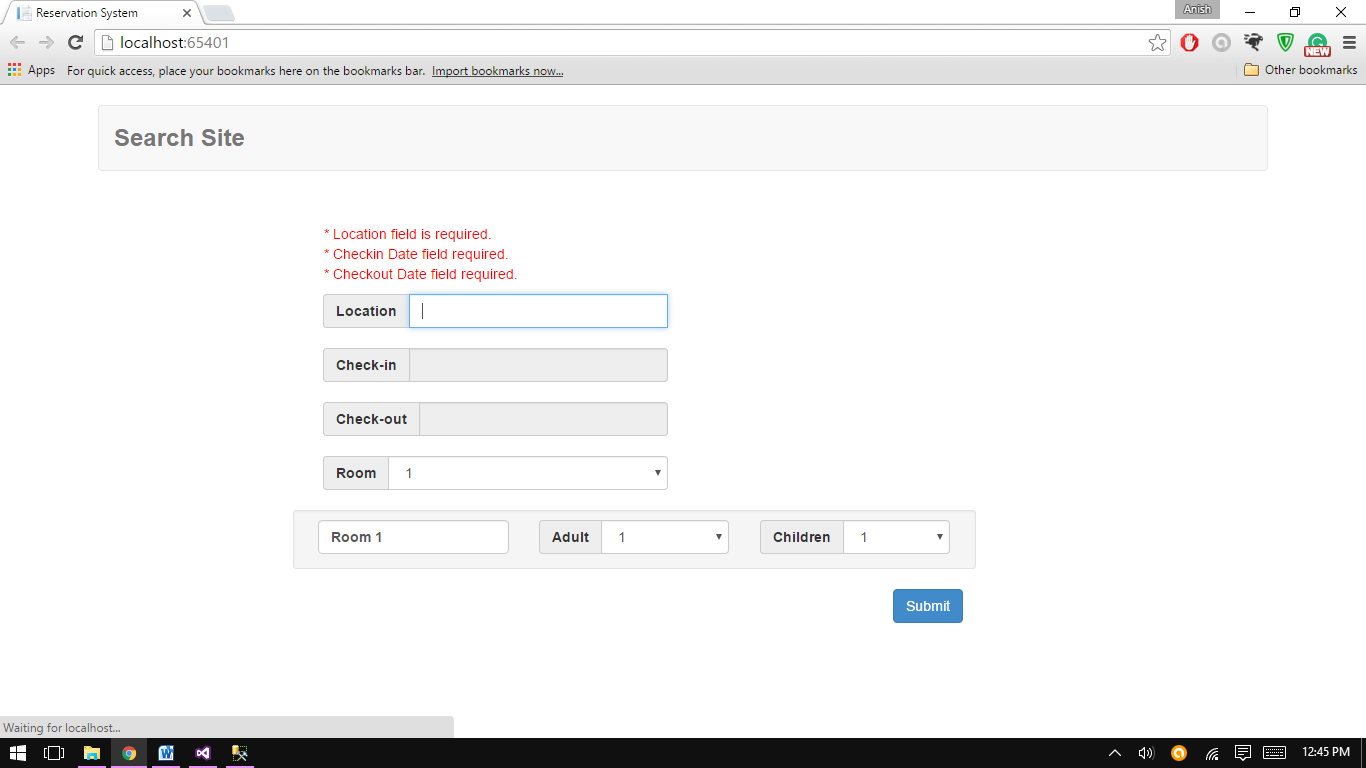
Reservations table and **ReservationDetails** table have one to many relations. For example, reservation with 5 rooms will have 1 record in Reservation table, 5 different records in **ReservationDetails** table.

**Future Enhancement**

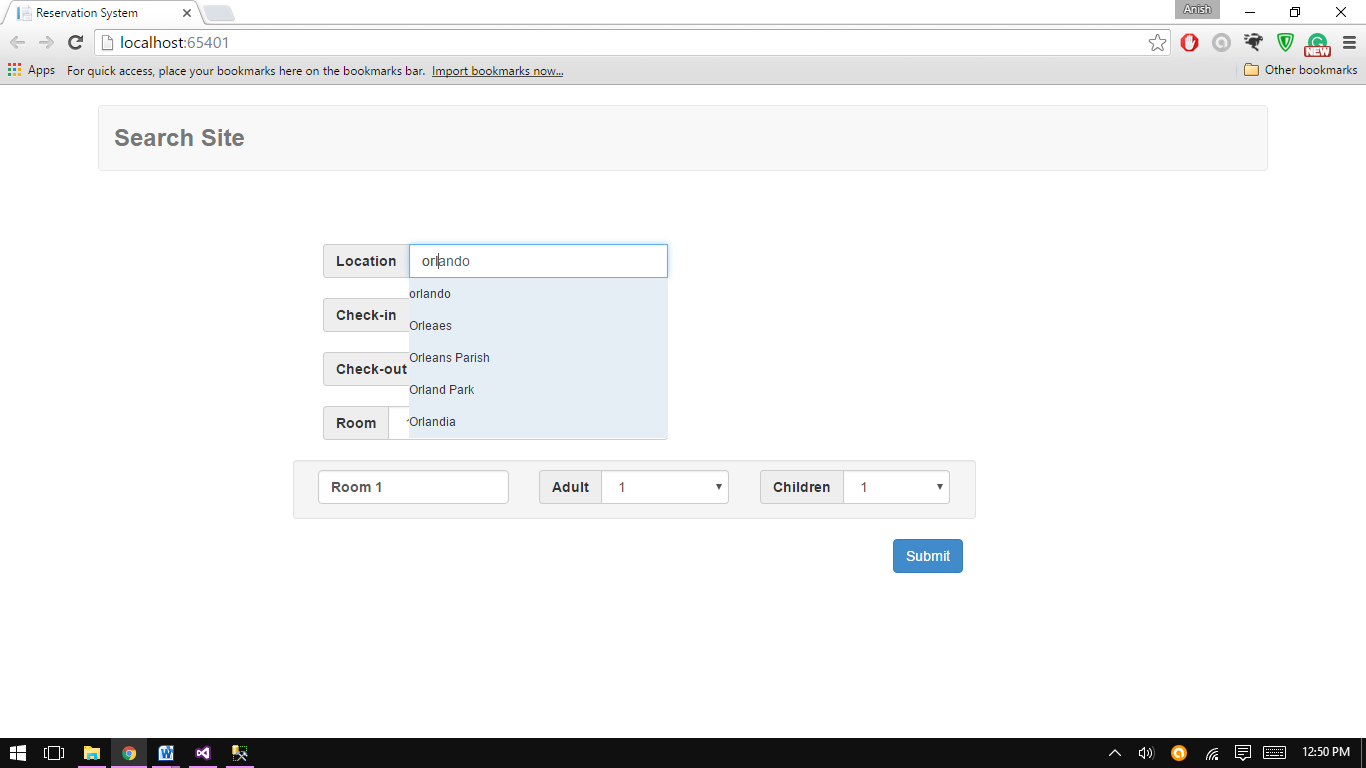
* Unit testing
* Integrating Unit of Work, Transactional Scope.
* Using DTO models in service layer which will segregate the domain model with the view model.

**Screen Shots**

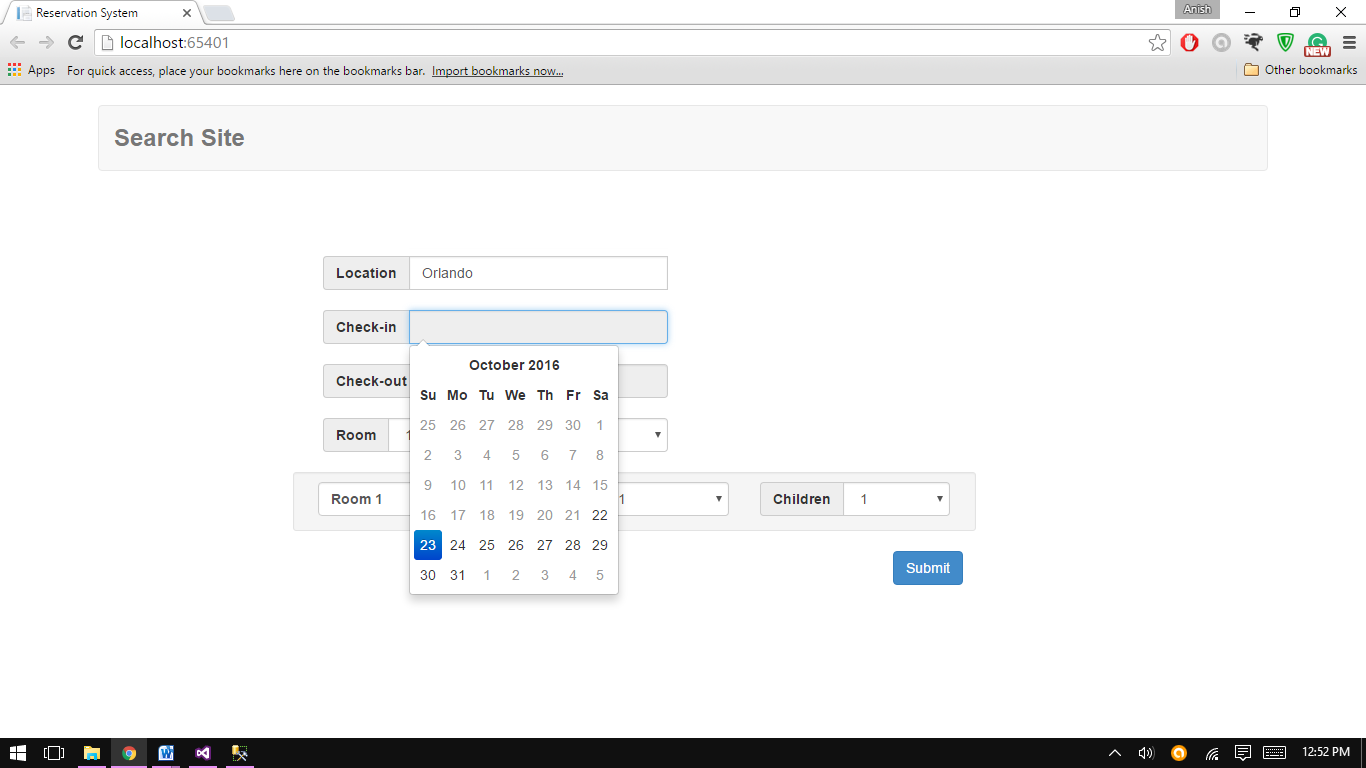
**Validation**



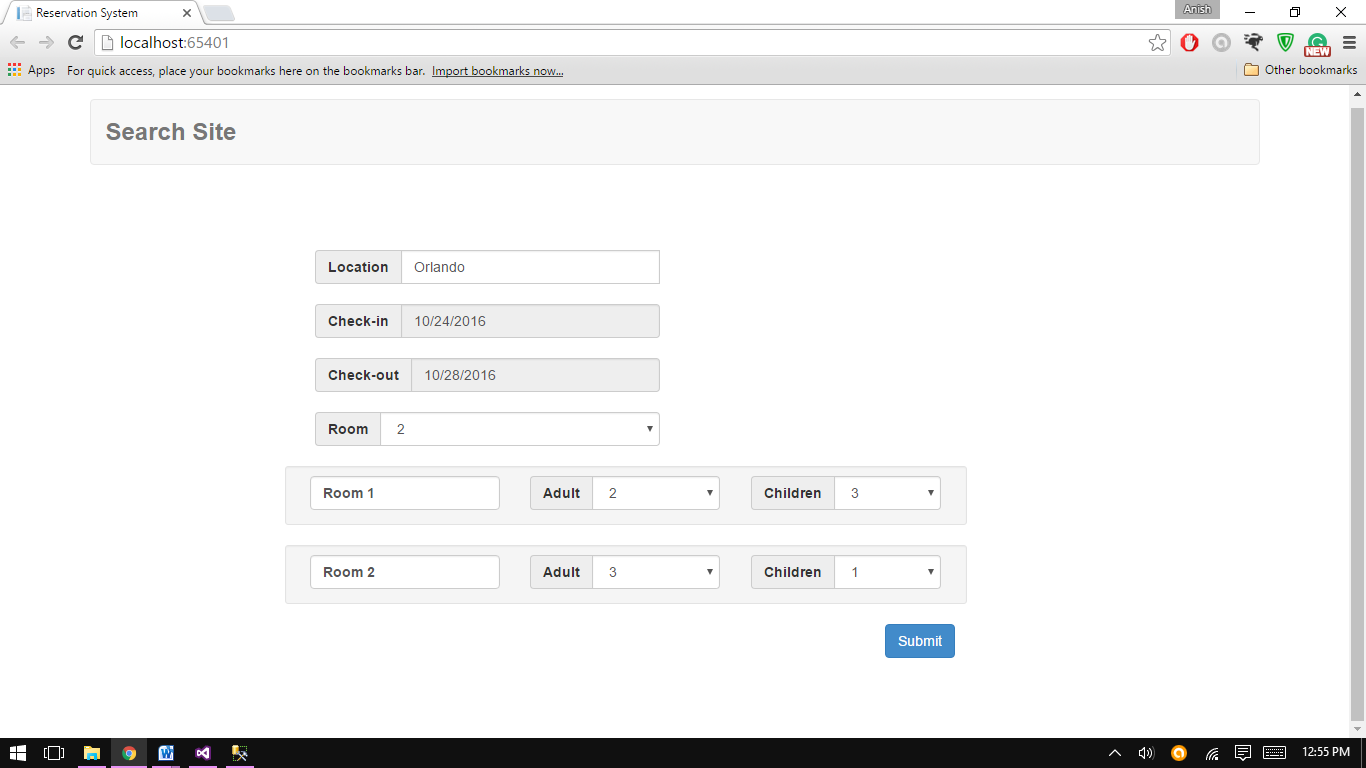
1. **Autocomplete in location**

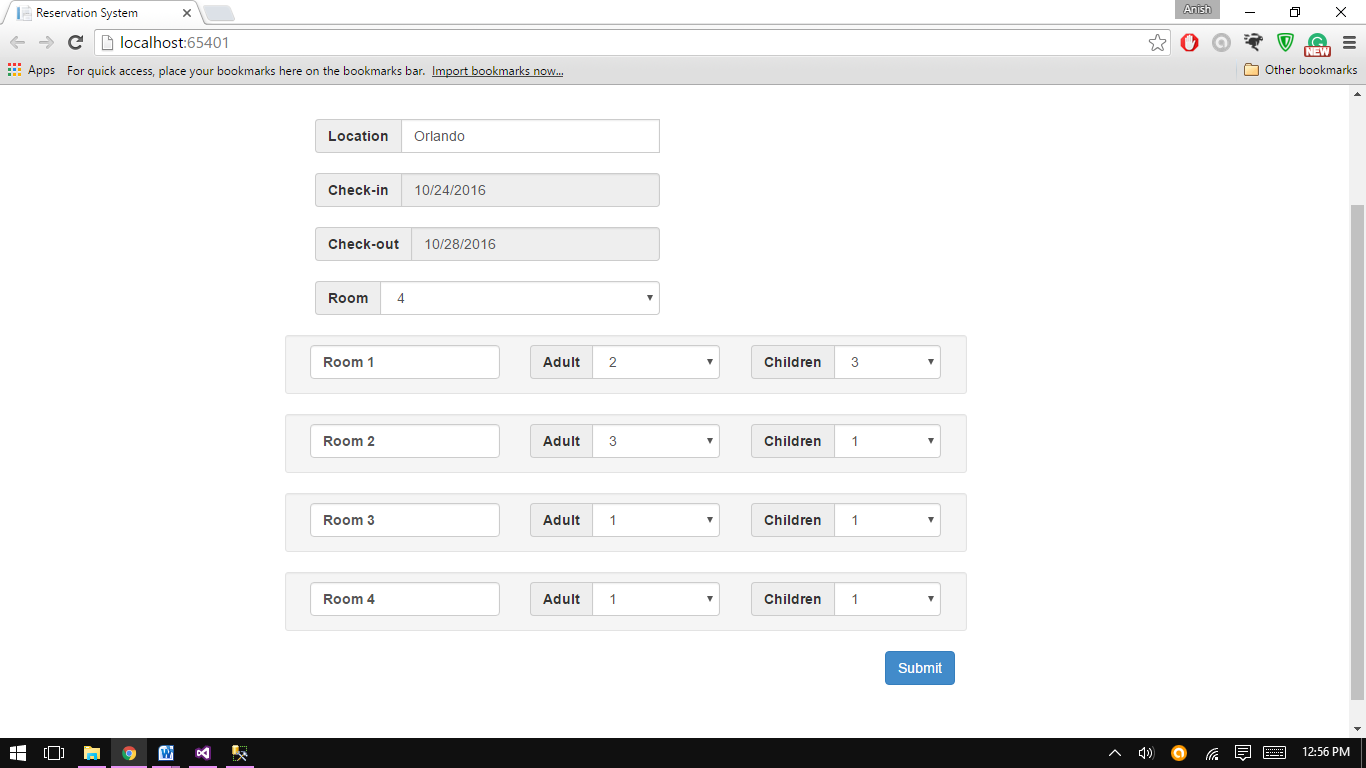


1. **Bootstrap – datepicker**



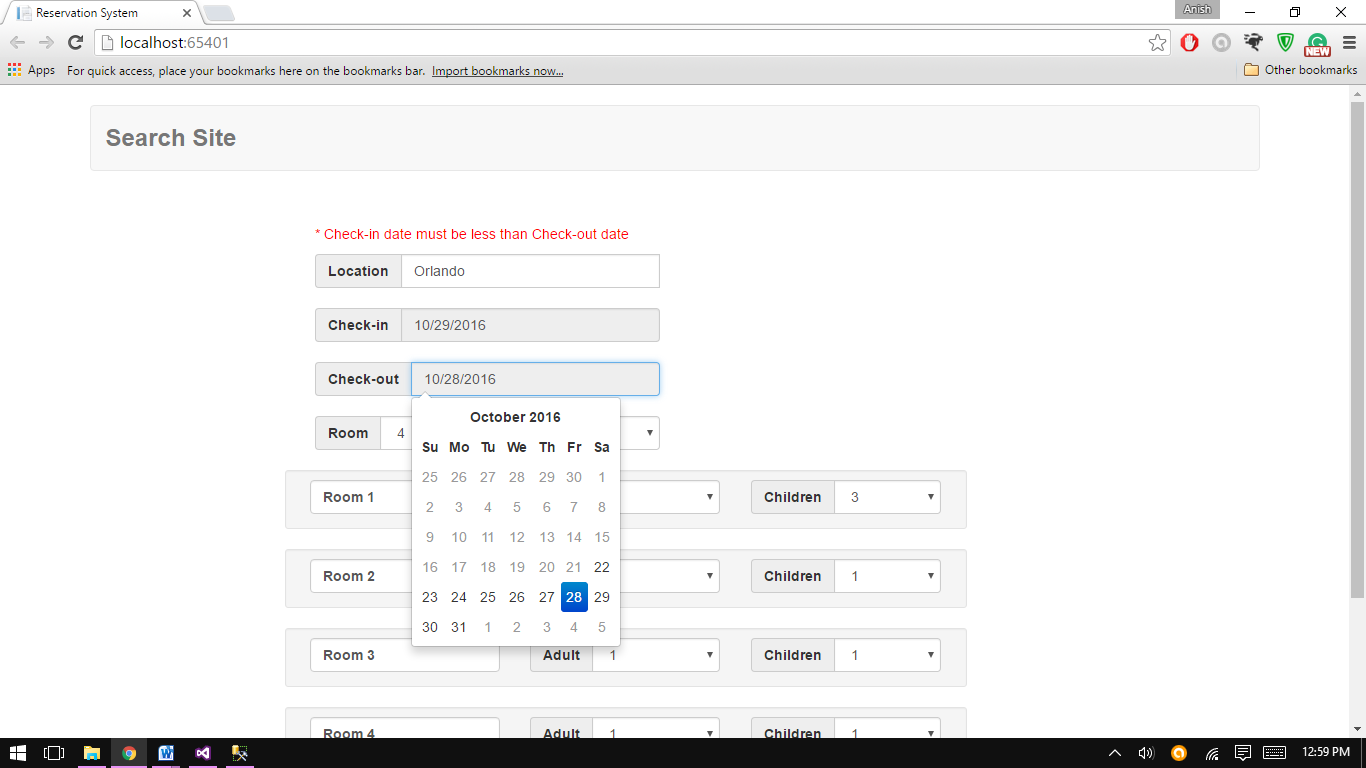
1. **Room detail list depending upon room dropdown**



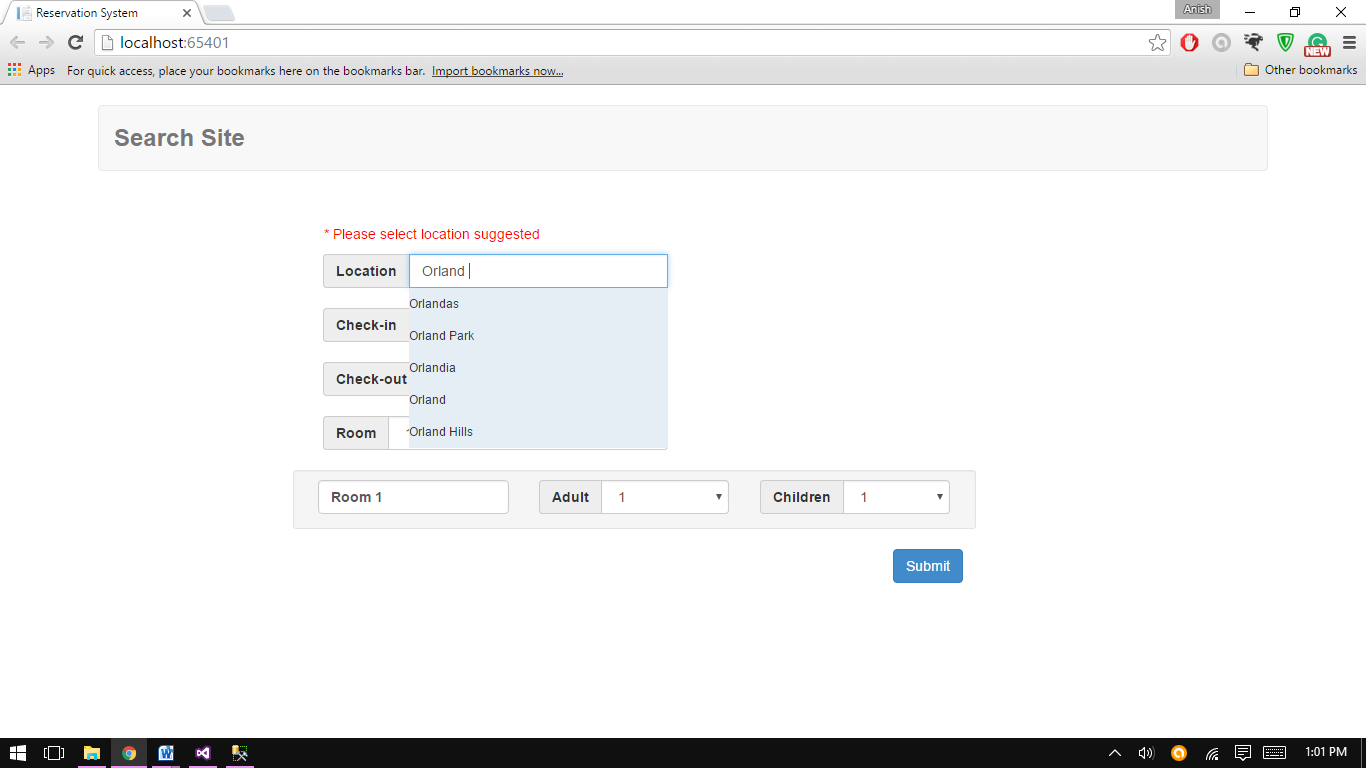


1. **Other validations**

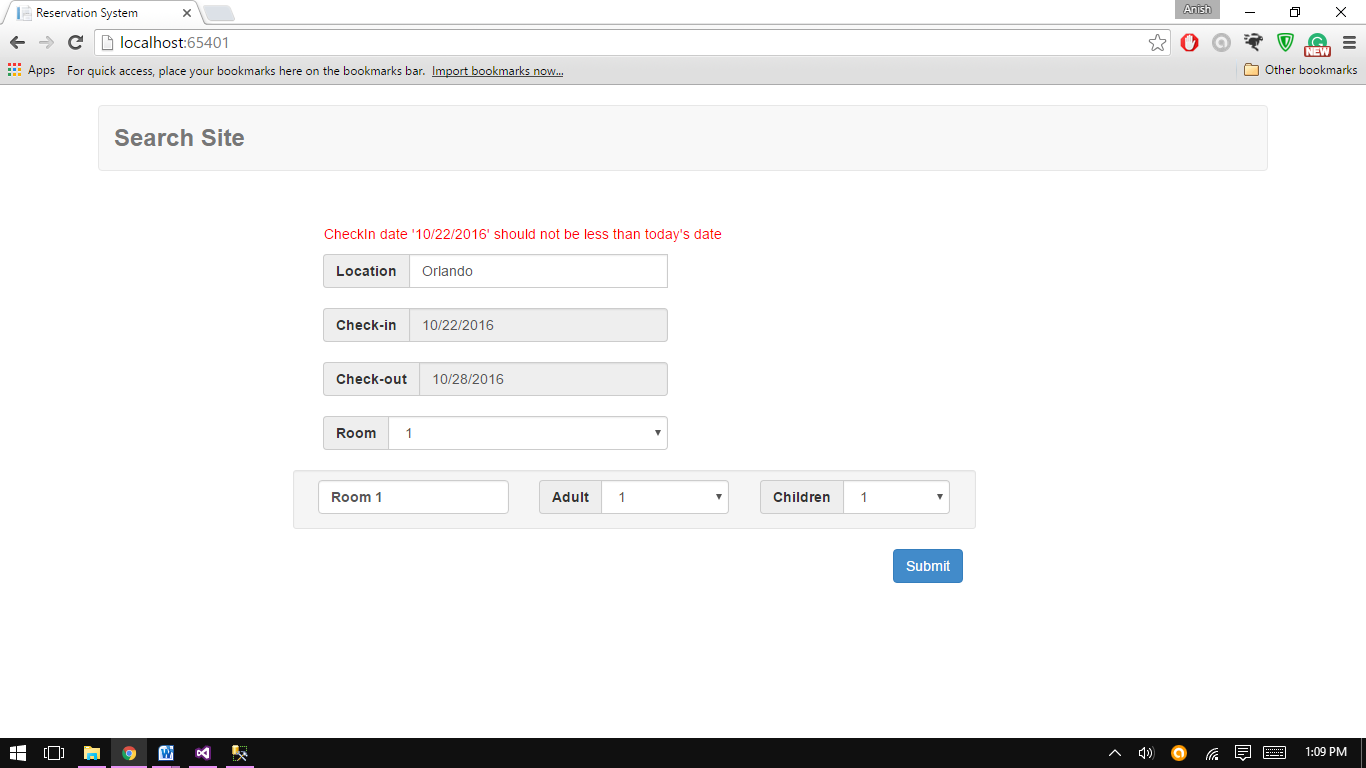
* Check-in date must be less than check-out date



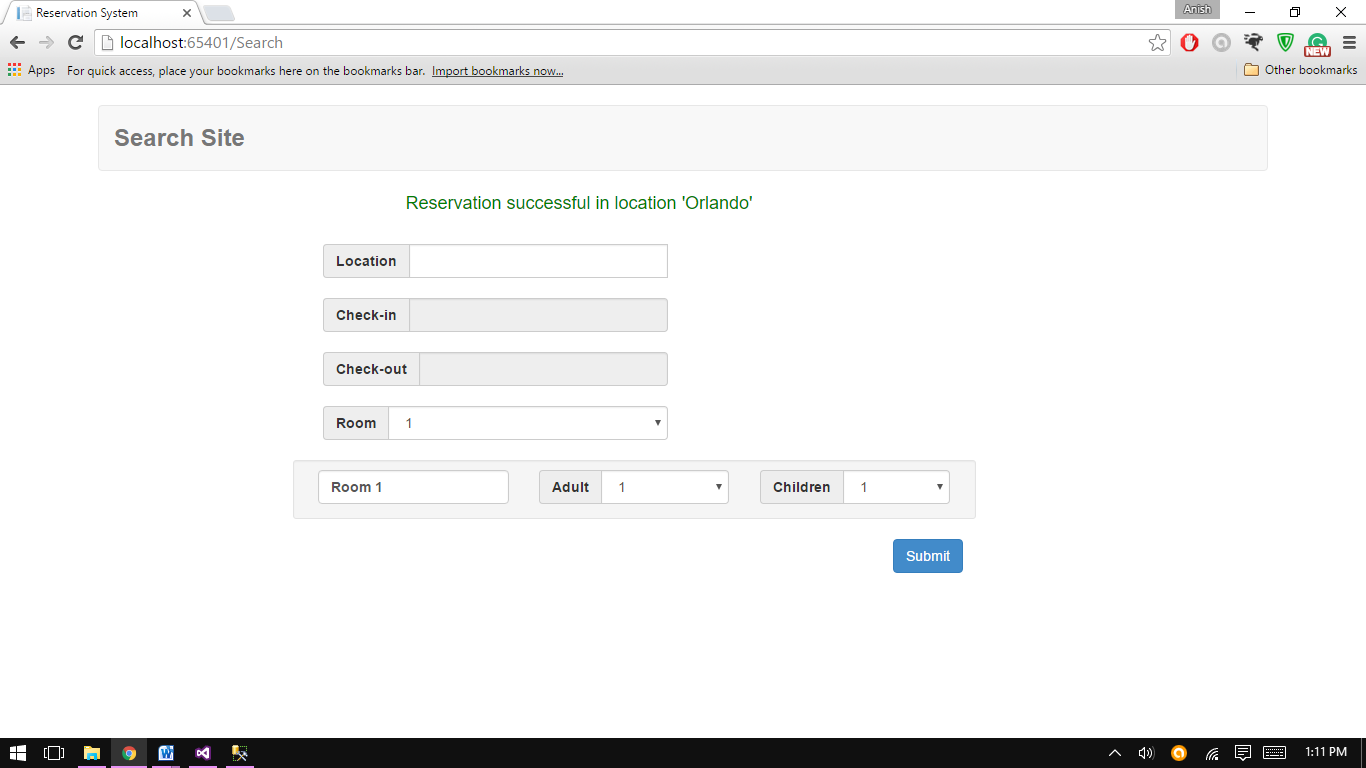
* Value selected in location is also validated using the same API.

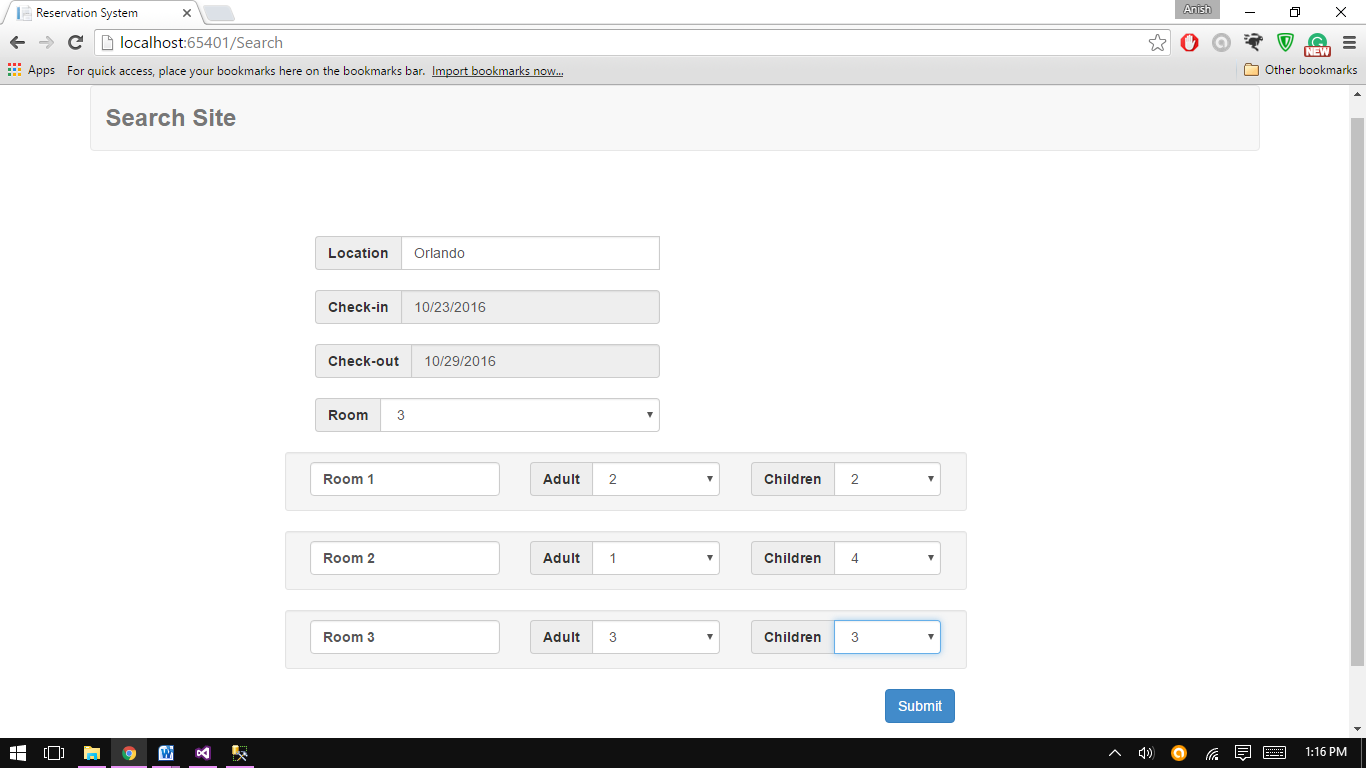


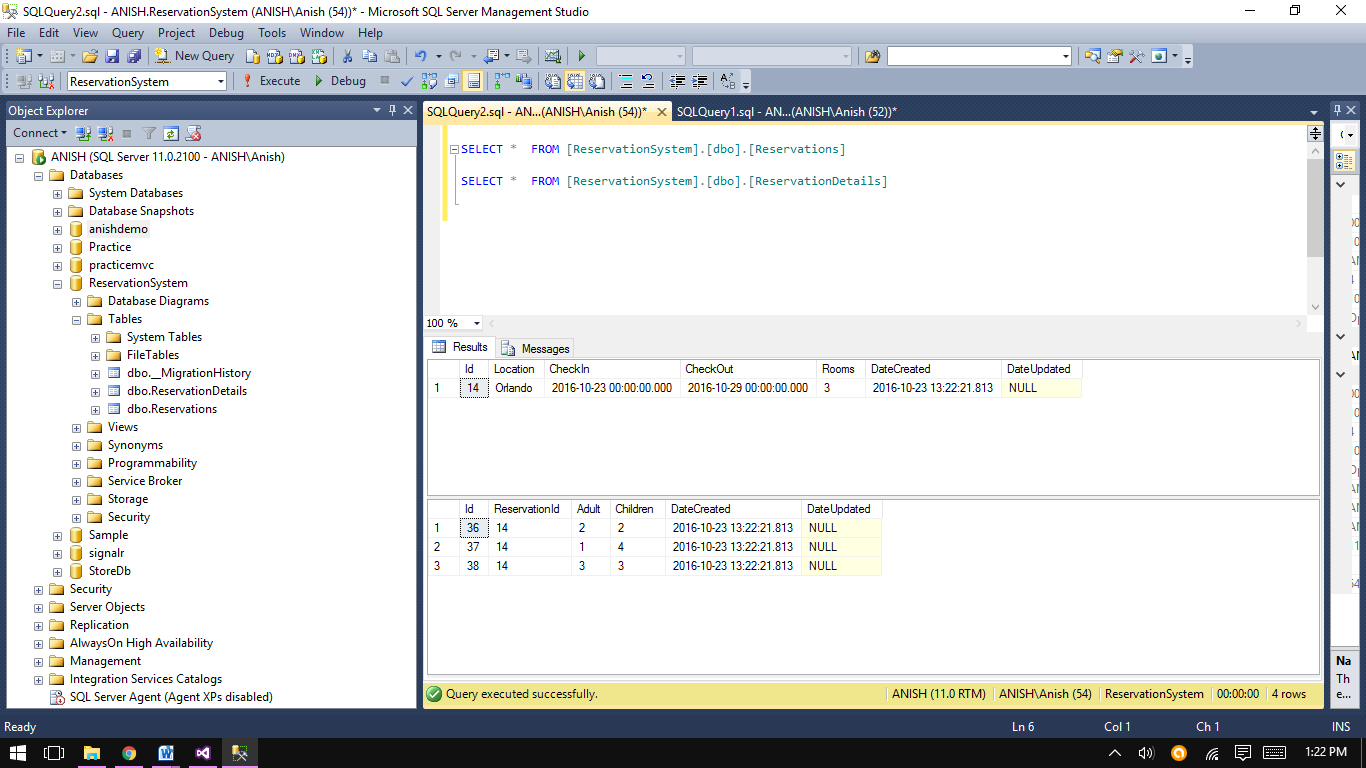
* Check-in date should not be less than today’s date.



1. **Success**







**Thanking you.**