

## Capstone Project: RentAPlace

**Problem Statement:** RentAPlace is an online platform for renting homes for short and long-term durations. The application serves two types of users: regular users (renters) and owners. Users can search for properties, while owners can manage their listings.

### User Stories

#### User Stories (Renters)

1. As a user, I should be able to log in, log out, and register in the application.
2. As a user, I should be able to view top-rated properties in different categories.
3. As a user, I should be able to search for properties based on various criteria:
  - Check-in and check-out dates
  - Desired location
  - Property type (e.g., flat, villa, apartment)
  - Features (e.g., pool, beach-facing, garden, etc.)
4. As a user, I should be able to see selected properties in a list or card view.
5. As a user, I should be able to view 4 to 5 pictures of each property.
6. As a user, I should be able to reserve a property.
7. As a user, I should be able to send messages to the owner through the application.

#### Owner Stories

1. As an owner, I should be able to log in, log out, and register in the application.
2. As an owner, I should be able to add multiple properties to the platform.
3. As an owner, I should be able to update, delete, and view my properties.
4. As an owner, I should be able to view messages from users.
5. As an owner, I should be able to reply to user messages.
6. As an owner, I should receive email notifications when a user reserves a property.
7. As an owner, I should be able to confirm the reservation status.

### Instructions

- Use a designated folder on the server for uploading property images.
- Share the database structure in a `.sql` file.
- Create a separate Web API for sending and receiving messages.
- Use Entity Framework (EF) or any other ORM tool for database communication.
- Use ASP.NET Core MVC for building the back end and providing RESTful APIs.
- Use React for the front end, ensuring seamless interaction with the .NET Core API.

### Reference

For design inspiration and functionality, refer to [Airbnb](#).

# Sprint Plan

## Sprint I Objectives

1. Create a Use Case document.
2. Design the database schema, including all tables and their relationships.
3. Select an ORM tool (e.g., Entity Framework Core) for database interaction.
4. Identify required controllers for the API.
5. Create a static UI template in React to display questions and answers.

## Sprint II Objectives

1. Set up the database context object using the chosen ORM.
2. Implement CRUD operations for user management (login, logout, registration).
3. Develop user and owner page layouts in React.
4. Implement CRUD operations for property registration.

## Sprint III Objectives

1. Develop search and filter functionality by consuming the backend API.
2. Add Swagger UI for API documentation and testing.
3. Implement the reservation module for users.
4. Develop notification functionalities for both users and owners (via email and in-app notifications).

---

## Implementation Tips:

- **React Frontend:** Use React Router for navigation, Axios for API requests, and a state management library (like Redux) for handling global state.
- **.NET Core Backend:** Implement controllers for handling API requests, services for business logic, and models for data representation. Secure your API endpoints using JWT authentication.
- **Database:** Use SQL Server or any other preferred database with Entity Framework Core for ORM functionality.