Софтуерна документация

Онлайн система за наем и отдаване под наем на недвижимо имущество.

**House Renting System**

**ПРЕДНАЗНАЧЕНИЕ**

Целта на този документ е да послужи като ръководство при реализацията на онлайн система за наем и отдаване под наем на недвижимо имущество. Тук се очертава техническия дизайн, функционалните, нефункционалните и потребителските изискванията, като и архитектурната рамка, структурата на базата данни, верификацията на данни и софтуерните тестове.

**ЗА КОГО Е ПРЕДНАЗНАЧЕН СОФТУЕРНИЯТ ПРОДУКТ?**

Тази онлайн система може да бъде ползвана както от агенция за недвижими имоти, която иска да организира своя екип и да направи услугите си по достъпни за своите клиенти, така също и като платформа предлагаща на обикновени хора да търсят или да предлагат имоти под наем.

**БЪРЗ ПРЕГЛЕД**

Системата "House Renting System" ASP.NET Core MVC App е уеб приложение за наемане на къщи. Потребителите могат да разглеждат всички къщи с техните детайли, да наемат къща и да видят наетите от тях къщи. Те също така могат да станат Агенти. Агенти могат да добавят къщи, да виждат техните детайли и да редактират или изтриват само къщите, които са добавили. Администраторът разполага с всички привилегии на Потребителите и Агенти, като може да вижда всички регистрации в приложението и всички направени наеми.

**ПОТРЕБИТЕЛИ И РОЛИ**

1. **Нерегистриран потребител**

* Вижда само началната страница.

1. **Обикновен потребител (General Registered User):**

**Роли и възможности:**

* Разглежда всички налични къщи с подробна информация за тях.
* Може да наема къщи.
* Може да преглежда списък с вече наетите от него къщи.
* Има възможност да стане **Агент.**

1. **Агент (Agent):**

**Роли и възможности:**

* Добавя нови къщи в системата.
* Преглежда детайлите за къщите, които е добавил.
* Може да редактира или изтрива само къщите, които е добавил лично.

1. **Администратор (Admin):**

**Роли и възможности:**

* Има всички права на **Обикновен потребител** и **Агент**.
* Може да вижда всички регистрации в системата.
* Има достъп до пълна информация за всички направени наеми.
* Вижда списък с всички обикновени потребители и агенти.

**АНАЛИЗ НА ИЗИСКВАНИЯТА**

**Технологични изисквания**

1. **ASP.NET Core Framework**

* **Identity -** библиотека, предоставена от Microsoft, за управление на автентикация и авторизация. Тя позволява създаване, управление и удостоверяване на потребители чрез вградени функции като регистрация, вход и възстановяване на парола.
* **MVC и Razor Pages – (Model-View-Controller) –** шаблон и конвенциипредоставящи структуриране на сложни логически процеси и зависимости. Организира комуникацията между различните слоеве на приложението, като защитава и ограничава достъпа до базата данни.

1. **База данни:**
   * **Entity Framework Core** - **ORM** (Object-Relational Mapping) е технология за работа с данните на **Identity**, която свързва обектно-ориентираните програми с релационни бази данни, като позволява разработчиците да работят с базата данни чрез обекти от езика за програмиране вместо директно с SQL заявки.
   * **Code-First** подход - дефиниране на модели чрез **Entity** класове (C# код), въз основа на което ORM автоматично генерира базата данни и нейната структура (таблици, колони, връзки). Създаването и промяната на базата данни се извършва чрез миграции.
   * **SQL Server** - релационна система за управление на бази данни.
2. **Валидация:**
   * **Data Annotations** - атрибути, които се използват за дефиниране на правила за валидиране, ограничения за база данни и метаданни за класове и свойства. Те са удобен начин да гарантираме, че данните, с които работим, отговарят на определени изисквания, без да пишем допълнителна логика.

**БАЗИ ДАННИ**

1. **Добавяне на класа Category (Категория)**

**Свойства:**

* **Id –** уникално цяло число, което служи като първичен ключ (Primary Key).
* **Name –** текстово поле (string) с максимална дължина 50 символа (задължително поле).
* **Houses –** колекция от обекти от тип House (навигационно свойство, което представлява свързани имоти).

1. **Добавяне на класа House (Имоти)**

**Свойства:**

* **Id** – уникално цяло число, което служи като първичен ключ (Primary Key).
* **Title** – текстово поле (string) с минимална дължина 10 и максимална дължина 50 символа (задължително поле).
* **Address** – текстово поле (string) с минимална дължина 30 и максимална дължина 150 символа (задължително поле).
* **Description** – текстово поле (string) с минимална дължина 50 и максимална дължина 500 символа (задължително поле).
* **ImageUrl** – текстово поле (string), представляващо URL адрес на изображение (задължително поле).
* **PricePerMonth** – десетично число (decimal) с минимална стойност 0 и максимална стойност 2000 (задължително поле).
* **CategoryId** – цяло число (integer), което е задължително.
* **Category** – обект от тип Category.
* **AgentId** – цяло число (integer), което е задължително.
* **Agent** – обект от тип Agent.
* **RenterId** – текстово поле (string).

1. **Добавяне на класа Agent (Агент)**

**Свойства**:

* **Id** – уникално цяло число, което служи като първичен ключ (Primary Key).
* **PhoneNumber** – текстово поле (string) с минимална дължина 7 и максимална дължина 15 символа (задължително поле).
* **UserId** – текстово поле (string), което е задължително.
* **User** – обект от тип **IdentityUser**.

**Структура на базата данни:**

A screenshot of a computer

Description automatically generated