**Rent-a-scooter**

Скутер на час (Scooter per hour) е система, която служи за връзка между потребител и rent-a-scooter компания. Чрез системата даден потребител може да разгледа свободните тротинетки за определени дати и да резервира, че иска да наеме дадена тротинетка.

В системата има два вида потребители: администратори (Administrator) и клиенти (Client). Системата работи с потребители, като достъпът до нея е отказан, ако клиента не е вписан с профила си. За всеки потребител се пази следната информация:

* потребителско име
* парола

Като клиент даден потребител може да разглежда предлаганите тротинетки.

Като клиент даден потребител може да филтрира свободните тротинетки в даден диапазон от дати (начална и крайна дата).

Като клиент даден потребител може да наеме тротинетка само ако тя е свободна в даден диапазон от начална и крайна дата.

Като клиент даден потребител може да пусне заявка за наем за избрана тротинетка, като заявката съдържа:

* начална дата
* крайна дата
* избрания екземпляр
* потребителя, който създава заявката
* крайна цена за избрания период
* статус - „изчакваща”, „отменена“, „активна“, „използвана“, като статус „изчакваща” е активен преди датата на вземане, статус „отменена“ е активен, ако заявката е отменена, „активна“ показва, че тротинетката е в потребителя, а „използвана“ индикира, че тя е върната.

Като клиент даден потребител може да вижда всички заявки, които е направил.

Като клиент даден потребител може да отмени заявка преди началната ѝ дата, но не и след като вече началната дата е минала. Заявките, които са отмнени придобиват статус „отменена“.

Като администратор даден потребител може да добавя, разглежда, променя и изтрива тротинетки. За всеки се пази:

* марка на тротинетката
* модел на тротинетката
* локация на тротинетката
* кратко описание (съдържащо техническа информация)
* цена за наем на тротинетката на час
* снимка

Като администратор даден потребител има пълен достъп върху всички заявки за наем (създаване, преглед, редактиране, изтриване).

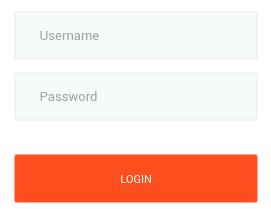
Като администратор даден потребител може да променя статуса на заявка за наем.

Като администратор даден потребител може да филтрира заявките за наем според техния статус.

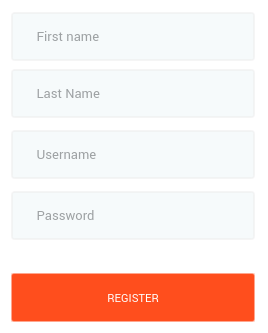
Като администратор даден потребител има контрол върху създаването и управлението на други потребители.

При стартиране на системата посетителят има възможност да влезе в своя потребителски профил или да си създаде такъв. Ролята на създадения профил е на обикновен потребител.

Влизането в системата става, като се предоставят име и парола.



Ако няма регистрация, даден потребител може да се регистрира.



За всички формуляри трябва да има подходящи валидации. Създайте възможност за филтриране на заявките по техния статус. Създайте възможност за филтриране на тротинетките по марка, модел и цена (от - до).

Не се допускат два потребителя с еднакво потребителско име или електронна поща.

Дадена тротинетка не може да бъде наемана, ако през дадения период вече ще бъде наета от друг потребител.

# **Система за управление на rent-a-scooter**

За успешното полагане на ДИ по практика трябва да се разработи трислойно приложение.

## Общи изисквания

Проектът трябва да бъде приложение реализирано чрез слой за данни, слой за услуги и презентационен слой.

Приложението трябва да има Web базиран интерфейс, за който са използвани HTML и CSS.

Приложението трябва да работи с релационна база от данни.

Проектът трябва да спазва официалните препоръки (style guide) за използваните технологии и да разполага със смислени коментари.

## Критерии за оценяване

Точки се присъждат на базата на три компонента:

* Изграждане на презентационна част
* Структура на базата от данни
* Разработка на функционалности

Критерии и скала за оценяване за разработка на проект:

|  |  |  |
| --- | --- | --- |
| **№** | **Критерии за формиране на оценката** | **Максимален** |
|  | **брой точки** |
|  |
| 1. | Реализация на презентационна част (HTML, CSS) | **15** |
| 1.1. | Oформление на страница - позициониране на елементи, чрез HTML тагове (например: div) и CSS атрибути (например: float, clear и display) | 5 |
| 1.2. | използване на семантични HTML тагове (header, nav, main и др.) | 5 |
| 1.3. | използване на CSS селектори, изнасяне на CSS във външен файл, използване на класове в HTML кода | 5 |
| 2. | Умения за работа с БД | **20** |
| 2.1. | Изграждане на структурата на базата от данни - таблицa | **10** |
| 2.1.1. | Създаване на таблица/и за потребители и роли | 3 |
| 2.1.2. | Създаване на таблица/и за тротинетки | 3 |
| 2.1.3. | Създаване на таблица/и за заявки за наем | 4 |
| 2.2. | Изграждане на структурата на базата от данни - връзки между таблиците | **10** |
| 2.2.1. | Създаване на връзка между потребител и роля | 3 |
| 2.2.2. | Създаване на връзка между потребител и заявка за наем | 3 |
| 2.2.3. | Създаване на връзка между тротинетки и заявки за наем | 4 |
| 3. | Разработка на сървърна система, базирана на трислоен модел | **20** |
| 3.1. | Създаване на потребителска система - вход и регистрация на потребител | 1 |
| 3.2. | Управление на потребители (преглед, добавяне, редактиране, изтриване) | 1 |
| 3.3. | Управление на тротинетки (преглед, добавяне, редактиране, изтриване) според условието | 4 |
| 3.4. | Управление на заявки (преглед, добавяне, редактиране, изтриване, промяна на статус) според условието | 4 |
| 3.5. | Филтриране на тротинетки според условието | 2 |
| 3.6. | Филтриране на заявки за наем според условието | 2 |
| 3.7. | Възможност за качване на изображение към кола | 2 |
| 3.8. | Наличие на всички роли и коректната им работа, според условието | 2 |
| 3.9. | Налична валидация за всички формуляри | 2 |
| 4. | Здравословно и безопасно упражняване на професията. | **3** |
| 4.1. | Ученикът притежава знания и умения за безопасна работа на работното място | 1 |
| 4.2. | Ученикът притежава умения за адекватно реагиране в критични ситуации в рамките на компетенциите си | 1 |
| 4.3. | Ученикът притежава умения за оказване на първа помощ на пострадал при авария (при токов удар, пожар, наранявания и др.) | 1 |
| 5. | Професионално-личностни качества. | **2** |
| 5.1. | отговорност към извършената работа | 1 |
| 5.2. | трудова и технологична дисциплина - създаване на четим и ясен код, а в по-комплексни ситуации, детайлно описание, посредством коментари | 1 |
|  | **Общо** | **60** |

Максималният брой точки е 60, като тези точки се превръщат в оценка по формулата:

**Оценка = брой точки : 10**