Дніпровський національний університет

імені ОЛеся Гончара

Факультет прикладної математики

Кафедра математичного забезпечення ЕОМ

Модульна робота

з дисципліни “Бази даних”

на тему: «Система керування задачами»

Студента 2 курсу групи ПЗ-17-2

Спеціальності 121 Інженерія програмного забезпечення

Виконав:

Саввін М. О.

Перевірив:

Мащенко Л.В.

м. Дніпро

2019 р.

Зміст

[**Вступ** 3](#_Toc9963584)

[**1. Проектування системи баз даних** 3](#_Toc9963585)

[1.1. Постановка задачі 3](#_Toc9963586)

[1.2. Основні задачі, що треба розв'язати 3](#_Toc9963587)

[1.3 Можливість створення задач та лейблів 4](#_Toc9963588)

[1.4 Основні запити користувачів за рівнями доступу 5](#_Toc9963589)

[1.5 Виділення об’єктів, встановлення функціональних залежностей 5](#_Toc9963590)

[1.5.1 Об’єкти 5](#_Toc9963591)

[1.5.2 Функціональні залежності 6](#_Toc9963592)

[1.6 Нормалізація відношень та її обгрунтування 6](#_Toc9963593)

[1.7 Побудова граф-схеми інфологічної моделі та перевірка її коректності 7](#_Toc9963594)

[**2. Реалізація системи баз даних** 8](#_Toc9963595)

[2.1. Обґрунтування вибору СУБД 9](#_Toc9963596)

[2.2 Опис та структурна схема програми 11](#_Toc9963597)

[2.3 Опис розробки інтерфейсу 12](#_Toc9963598)

[2.3.1 Опис програми 13](#_Toc9963599)

[2.4 Використання та призначення основних функцій класів, тригерів та процедур 22](#_Toc9963600)

[2.5 Cтворення запитів в середовищі розробки та результати їх виконання 22](#_Toc9963601)

[2.6 Інструкції по використанню користувачами за рівнями доступу 24](#_Toc9963602)

[2.7 Опис результатів тестування та налагодження, рекомендації щодо супроводження 24](#_Toc9963603)

[2.7.1 Тестування 24](#_Toc9963604)

[2.7.2 Рекомендації щодо супроводження 24](#_Toc9963605)

[**Висновки** 25](#_Toc9963606)

[**Список літератури** 26](#_Toc9963607)

# **Вступ**

Метою моєї модульної роботи є створити систему, яка б змогла допомогти автоматизувати роботу продавця гітарного магазину, полегшити редагування інформації на сайті та оптимізувати систему, яку ми маємо сьогодні.

# **1. Проектування системи баз даних**

## 1.1. Постановка задачі

Розробити базу даних, засобами вибраної СУБД, що забезпечує введення, перегляд, редагування та виконання усіх запитів користувача.

Загальні вимоги до бази даних та роботи:

1. Провести обстеження предметної області;
2. Виявити основні інформаційні потоки та розв’язувані задачі
3. Провести оптимізацію системи баз даних та визначити якість отриманої системи відношень;
4. Виконати побудову інфологічної моделі предметної області та перевірити її коректність;
5. Для відповідної інфологічної моделі виконати реалізацію в одній з СУБД, що відповідає певним вимогам;
6. Розробити індивідуальні форми для користувачів різних рівнів, що забезпечують дружній інтерфейс, орієнтований на предметну область . Ці форми мають надавати можливість навігації по таблиці та виконання задач користувачів.
7. Створити запити різноманітних типів.
8. Контроль даних при введенні інформації по типу та діапазону, значенню.
9. Виконати тестування та налагодження програмного продукту;
10. Розробити інструкцію для користувачів різних рівнів;

Предметна область модульної роботи: «Система керування задачами»

## 1.2. Основні задачі, що треба розв'язати

- інтерфейс, який розрахований для клієнтів

- створення можливості додавання, редагування, видалення записів

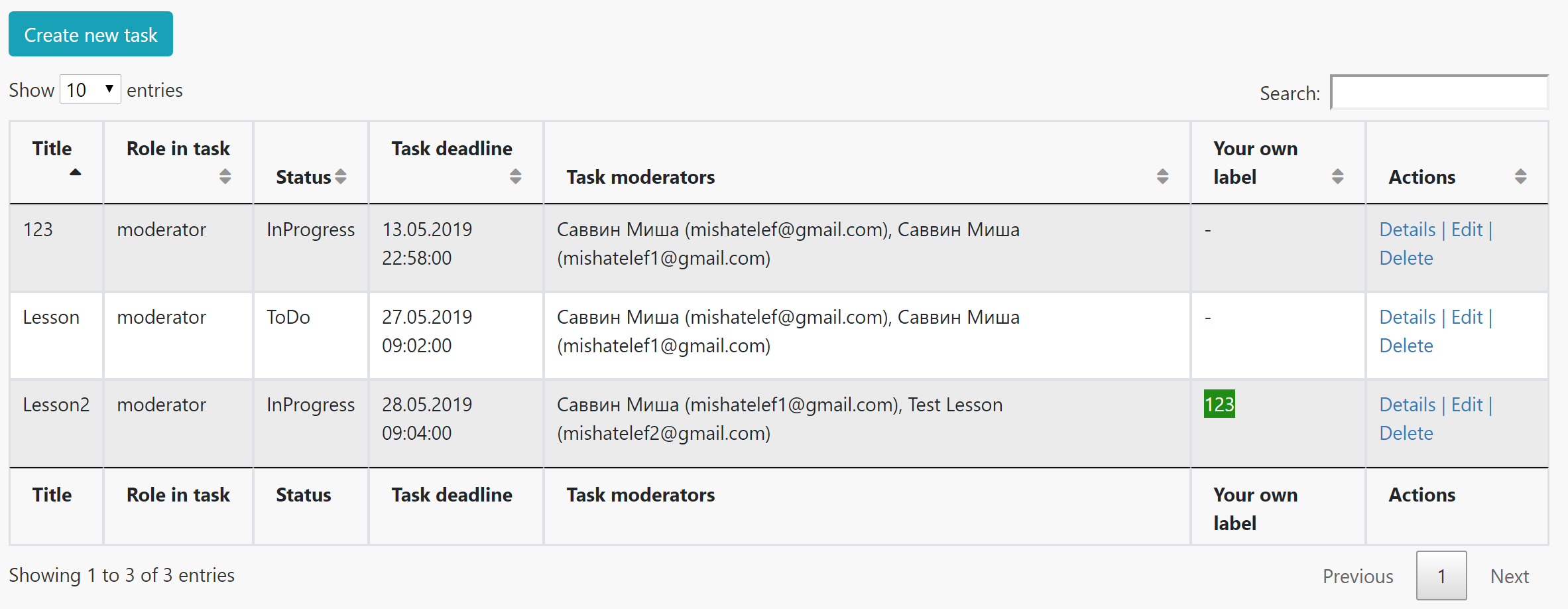
- перегляд задач

- перегляд статистики

- сортування та пошук

## 1.3 Можливість створення задач та лейблів

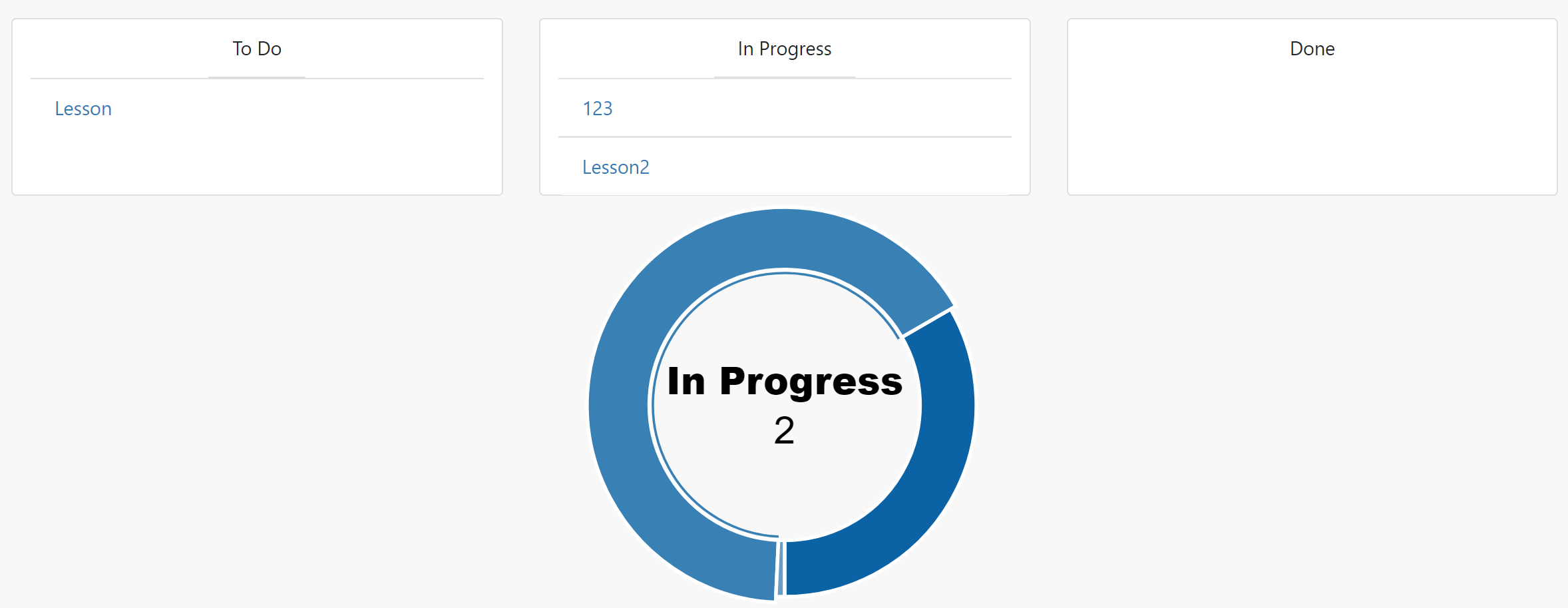
* Задачі



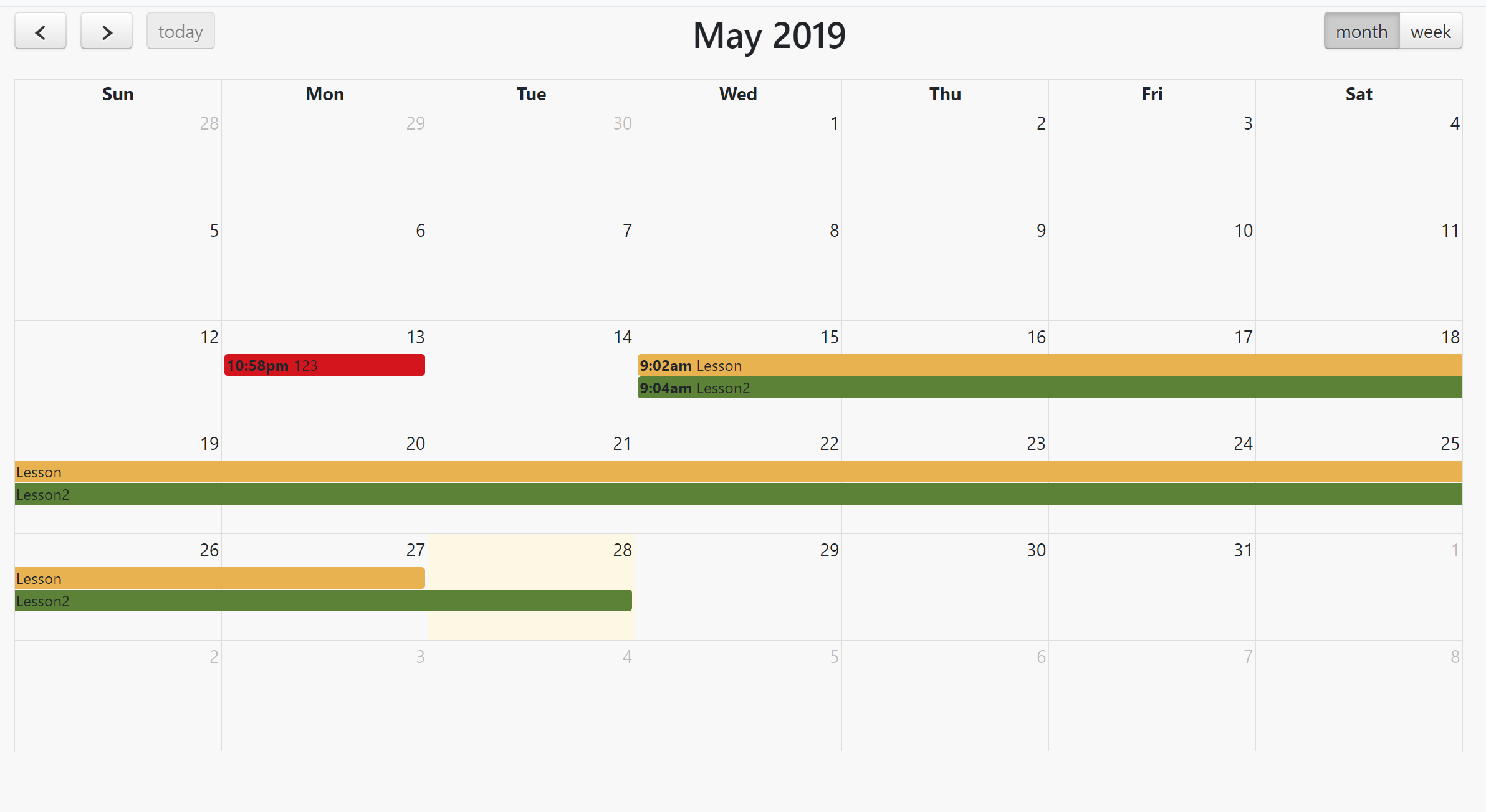
* Лейбли



* Дашбоард



* Календар задач



## 1.4 Основні запити користувачів за рівнями доступу

- Тільки ті користувачі бачать задачу, які є її модератором або глядачем

- Тільки ті користувачі можуть змінювати задачу, які є її модератором

- Тільки ті користувачі які є у системі можуть щось створювати

## 1.5 Виділення об’єктів, встановлення функціональних залежностей

### 1.5.1 Об’єкти

|  |  |
| --- | --- |
| **Об’єкт** | **Інформація** |
| AspNetUsers | Identity |
| AspNetUserTokens | Identity |
| AspNetUserRoles | Identity |
| AspNetUserLogins | Identity |
| AspNetUserClaims | Identity |
| AspNetRoleClaims | Identity |
| Labels | ID, UserId, Title, Color |
| NotificationTypes | ID, Title |
| Task\_Label\_User | LabelId, TaskId, UserId |
| TaskModerator\_User | UserId, TaskId |
| Tasks | Id, Title, Description, Weight, EndDate, CreationTime, ClosingTime, StatusId |
| TaskStatus | Id, Title |
| TaskViewer\_User | UserId, TaskId |

### 1.5.2 Функціональні залежності

*Функціональна залежність*. Поле Б таблиці функціонально залежить від поля А тієї ж таблиці в тому і тільки в тому випадку, коли в будь-який заданий момент часу для кожного з різних значень поля А обов'язково існує тільки одне з різних значень поля Б. Зауважимо, що поля А і Б можуть бути складе-ними.

*Повна функціональна залежність*. Поле Б знаходиться в повній функціона-льної залежності від складеного поля А, якщо воно функціонально залежить від А і не залежить функціонально від будь-якої підмножини поля А.

Виходячи із визначення і введених даних, між полями таблиць можна простежити такі функціональні залежності.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID заказу | ID корист. | ID продав | ID продукту | Ціна |

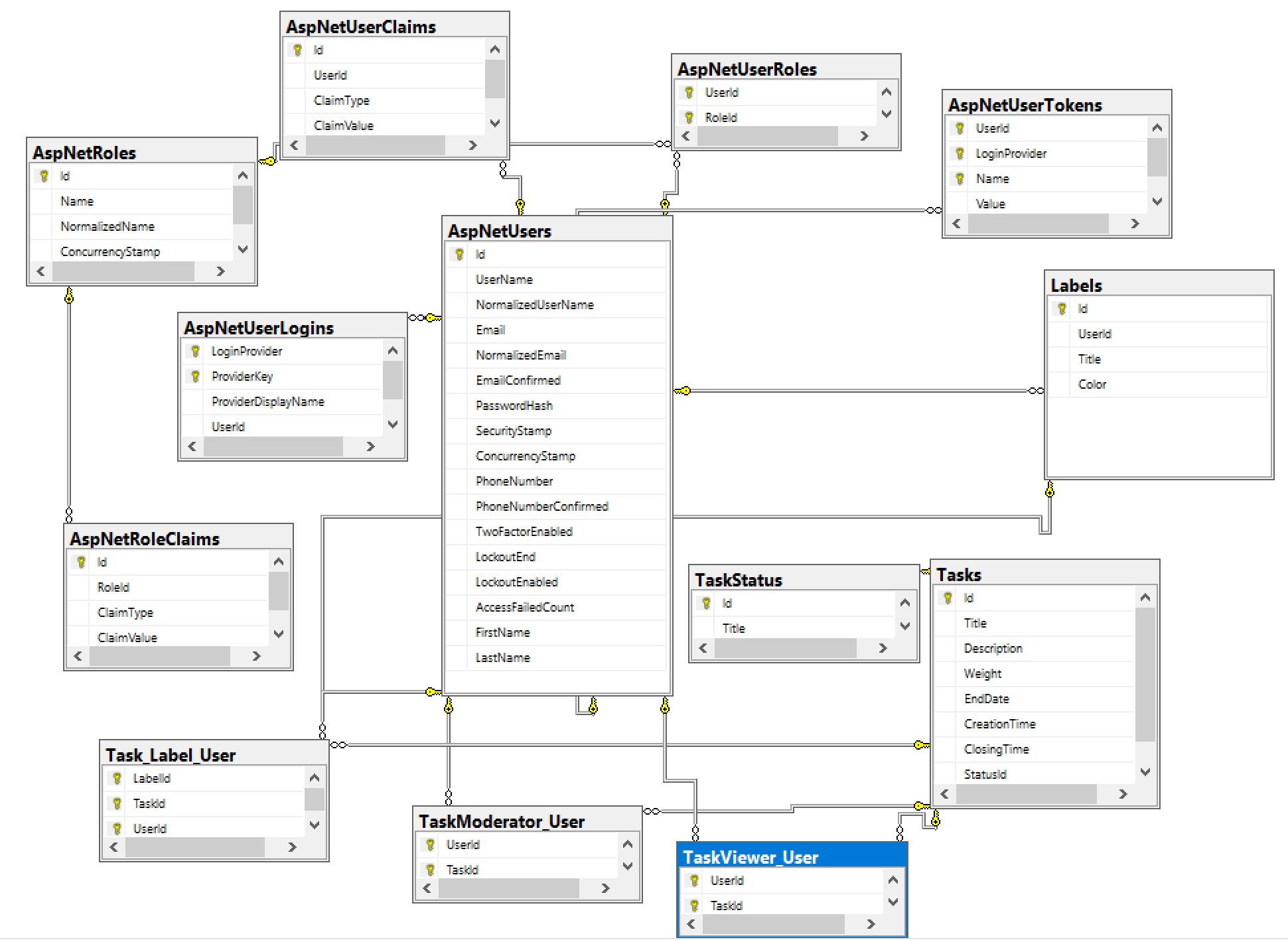
## 1.6 Нормалізація відношень та її обгрунтування

Нормалізація - це розбиття таблиці на дві або більше, що володіють кращими властивостями при включенні, зміні і видаленні даних.

Відношення (таблиця) знаходиться в першій нормальній формі (1НФ) тоді і тільки тоді, коли вона не має тотожних кортежів і кожен атрибут відношення є атомарним.

Таблиця знаходиться в другій нормальній формі (2НФ), якщо вона задовольняє визначенню 1НФ і всі її поля, що не входять в первинний ключ, зв’язані повною функціональною залежністю з первинним ключем.

Таблиця знаходиться в третій нормальній формі (3НФ), якщо вона задово-льняє визначенню 2НФ і жодне з її неключових полів не залежить функціонально від будь-якого іншого неключових поля.



*Рис. 1.6.1 -зображення таблиць і зв’язків*

Усі таблиці знаходяться у 3НФ.

## 1.7 Побудова граф-схеми інфологічної моделі та перевірка її коректності

Граф-схема є коректною, оскільки:

* Включає всі необхідні об’єкти області
* Відсутні об’єкти, що повторюються
* Враховані всі зв’язки предметної області
* Якщо розглядати схему як граф, він ациклічний
* Якщо розглядати схему як орієнтований граф, то він зв’язний, тобто існує шлях між будь якими двома вершинами

Для доведення коректності інфологічної моделі розглянемо декілька складних запитів, що знадобились у системі та покажемо, що пересуваючись по зв’язкам їх можна реалізувати.

«Task-Label-User»:

User

TaskLabelUser

UserRole

# **2. Реалізація системи баз даних**

Label

TaskStatus

Task

## 2.1. Обґрунтування вибору СУБД

Я обрав PhpMyAdmin, але були й альтернативи. Розглянемо найяскравіші приклади з їх перевагами і недоліками.

1. **MySQL**

**MySQL - це найпоширеніша повноцінна серверна СУБД. MySQL дуже функціональна, та вільно розповсюджувальна СУБД, яка успішно працює з різними сайтами і веб додатками. Навчитися використанню цієї СУБД досить просто, так як на просторах інтернету ви легко знайдете більшу кількість інформації.**

Незважаючи на те, що в ній не реалізований весь SQL функціонал, MySQL пропонує досить багато інструментів для розробки додатків. Так як це серверна СУБД, додатки для доступу до даних, на відміну від SQLite працюють зі службами MySQL.

**Переваги MySQL**

* Простота в роботі - встановити MySQL досить просто. Існують різноманітні програми, наприклад GUI, що дозволяє досить легко працювати з БД
* Багатий функціонал - MySQL підтримує більшість функціоналу SQL.
* Безпека - велика кількість функцій забезпечують безпеку, які підтримується за замовчуванням
* Масштабованість - MySQL легко працює з великими обсягами даних і легко масштабується
* Швидкість - спрощення деяких стандартів дозволяє MySQL значно збільшити продуктивність.

**Недоліки MySQL**

* Відомі обмеження - за задумом в MySQL закладені деякі обмеження функціонала, які іноді необхідні в особливо вимогливих додатках.
* Проблеми з надійністю - через деякі способи обробки даних MySQL (зв'язку, транзакції, аудити) іноді поступається іншим СУБД по надійності.
* Повільна розробка - Хоча MySQL технічно відкрите ПЗ, існують скарги на процес розробки. Варто зауважити, що існують інші досить успішні СУБД створені на базі MySQL, наприклад MariaDB.

1. **PostgreSQL**

PostgreSQL є професійним з усіх трьох розглянутих нами СУБД. Вона вільно розповсюджується і максимально відповідає стандартам SQL. PostgreSQL або Postgres намагаються повною мірою використовувати ANSI / ISO SQL стандарти своєчасно з виходом нових версій.

Від інших СУБД PostgreSQL відрізняється підтримкою затребуваного об'єктно-орієнтованого і / або реляційного підходу до баз даних. Наприклад, повна підтримка надійних транзакцій, тобто атомарність, послідовність, ізоляційність, міцність (Atomicity, Consistency, Isolation, Durability (ACID).) Завдяки потужним технологіям Postgre дуже продуктивна. Паралельність досягнута не за рахунок блокування операцій читання, а завдяки реалізації управління різноманітним паралелізмом (MVCC), що також забезпечує відповідність ACID. PostgreSQL дуже легко розширювати своїми процедурами, які називаються збережені процедури. Ці функції спрощують використання постійно повторюваних операцій.

Хоча PostgreSQL і не може похвалитися великою популярністю на відміну від MySQL, існує досить велика кількість додатків, що полегшують роботу з PostgreSQL, незважаючи на всю потужність функціоналу. Зараз досить легко встановити цю СУБД використовуючи стандартні менеджери пакетів операційних систем.

**Переваги PostgreSQL**

* Відкрите ПЗ відповідає стандарту SQL - PostgreSQL - безкоштовне ПЗ з відкритим вихідним кодом. Ця СУБД є дуже потужною системою.
* Велике співтовариство - існує досить велика спільнота в якому ви запросто знайдете відповіді на свої питання
* Велика кількість доповнень - незважаючи на величезну кількість вбудованих функцій, існує дуже багато доповнень, що дозволяють розробляти дані для цієї СУБД і управляти ними.
* Розширення - існує можливість розширення функціоналу за рахунок збереження своїх процедур.
* Об'єктно - PostrgreSQL це не тільки реляційна СУБД, але також і об'єктно-орієнтована з підтримкою успадкування і багато іншого

**Недоліки PostgreSQL**

* Продуктивність - при простих операціях читання PostgreSQL може значно уповільнити сервер і бути повільніше своїх конкурентів, таких як MySQL
* Популярність - за своєю природою, популярністю ця СУБД похвалитися не може, хоча і є досить велика спільнота.
* Хостинг - в силу названих вище чинників іноді досить складно знайти хостинг з підтримкою цієї СУБД.

1. **MS SQL Server**

Переваги MS SQL Server:

* Масштабованість
* Дуже потужна продуктивність
* Простота у користуванні
* Готовність до використання в Інтернеті, інтрамережі і для електронної комерції
* Сховища даних
* Інтеграція з іншими продуктами Microsoft

## 2.2 Опис та структурна схема програми

Авторизація

Усі сторінки додатку, але обмежена інформація згідно ролей у задачі

Сторінка задач

Дашбоард та календар

**Основний рівень**

Сторінка лейблів

## 2.3 Опис розробки інтерфейсу

Програма була реалізована за допомогою технологій:

Back-end:

* + - MVC
    - Razor
    - Identity
    - Entity Framework Core
    - Embedded dependency injection
    - Automapper
    - Xunit
    - Moq

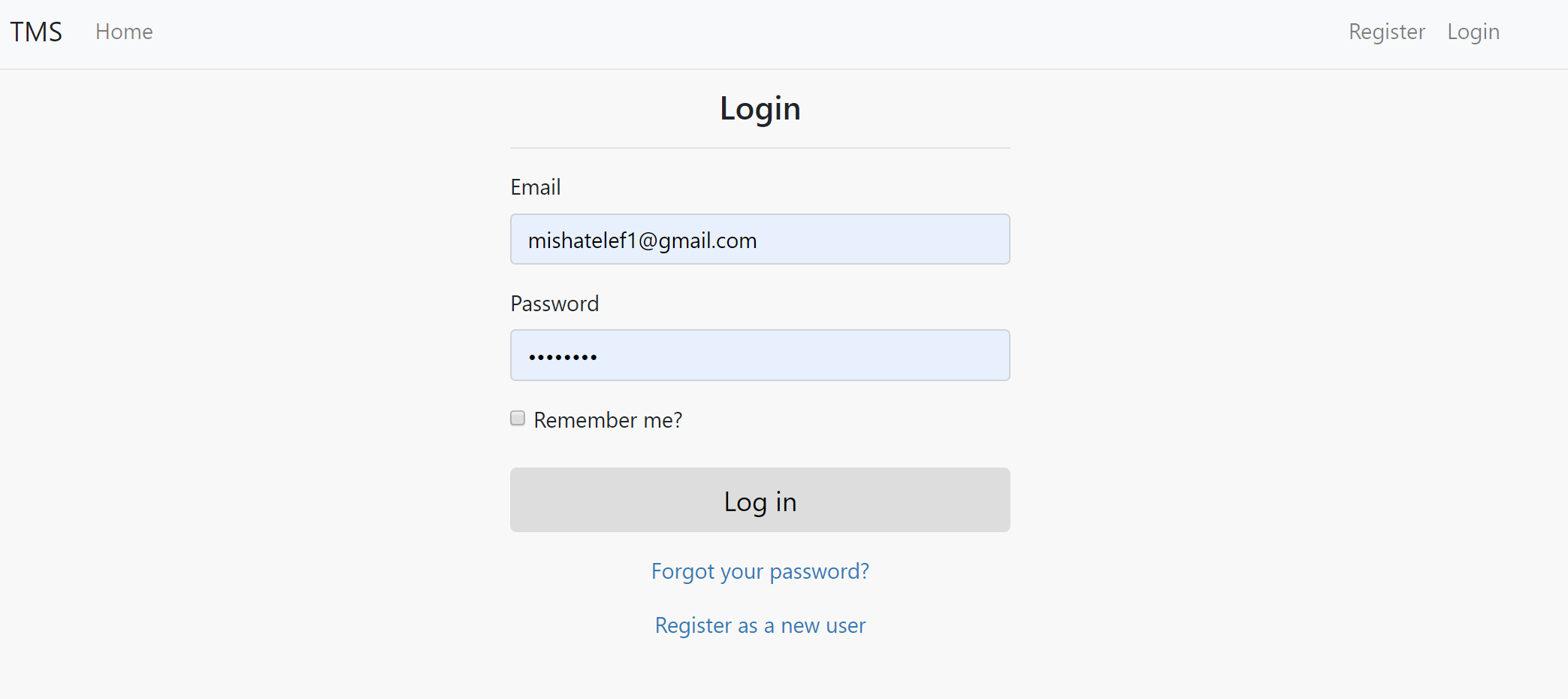
Front-end:

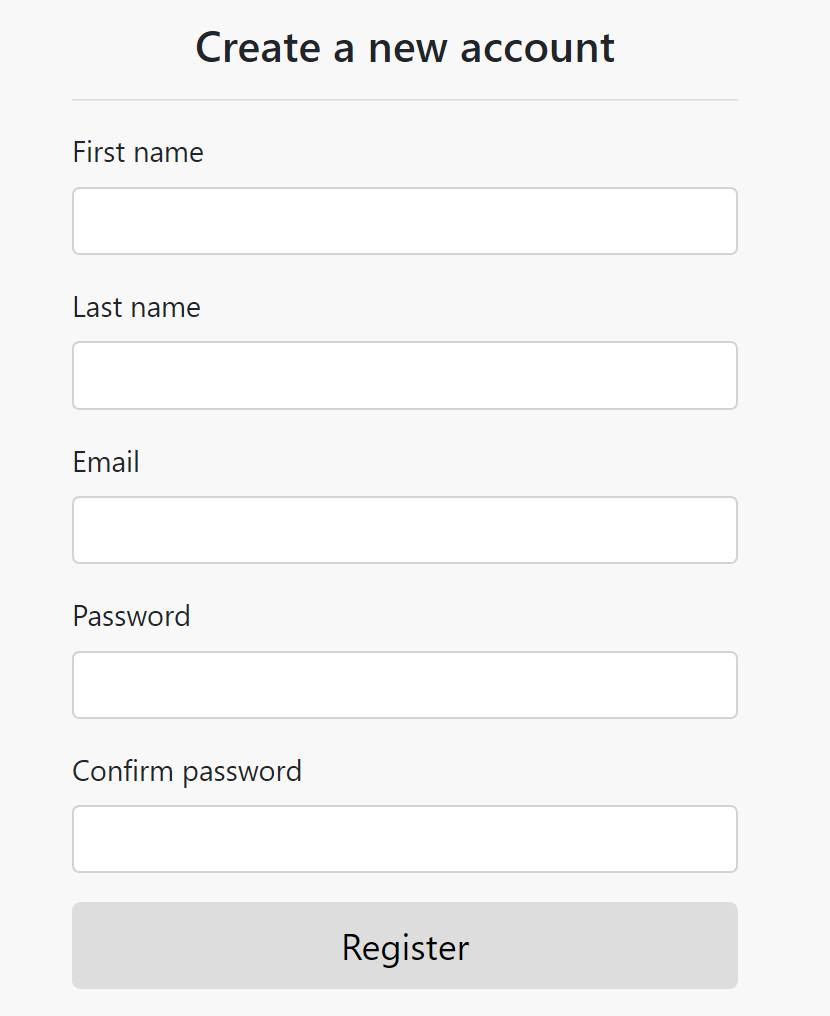
* + - Bootstrap 4.1
    - jQuery
    - Datatables
    - Bootstrap components:
      * + Calendar
        + Colorpicker
        + Select
        + Datetimepicker

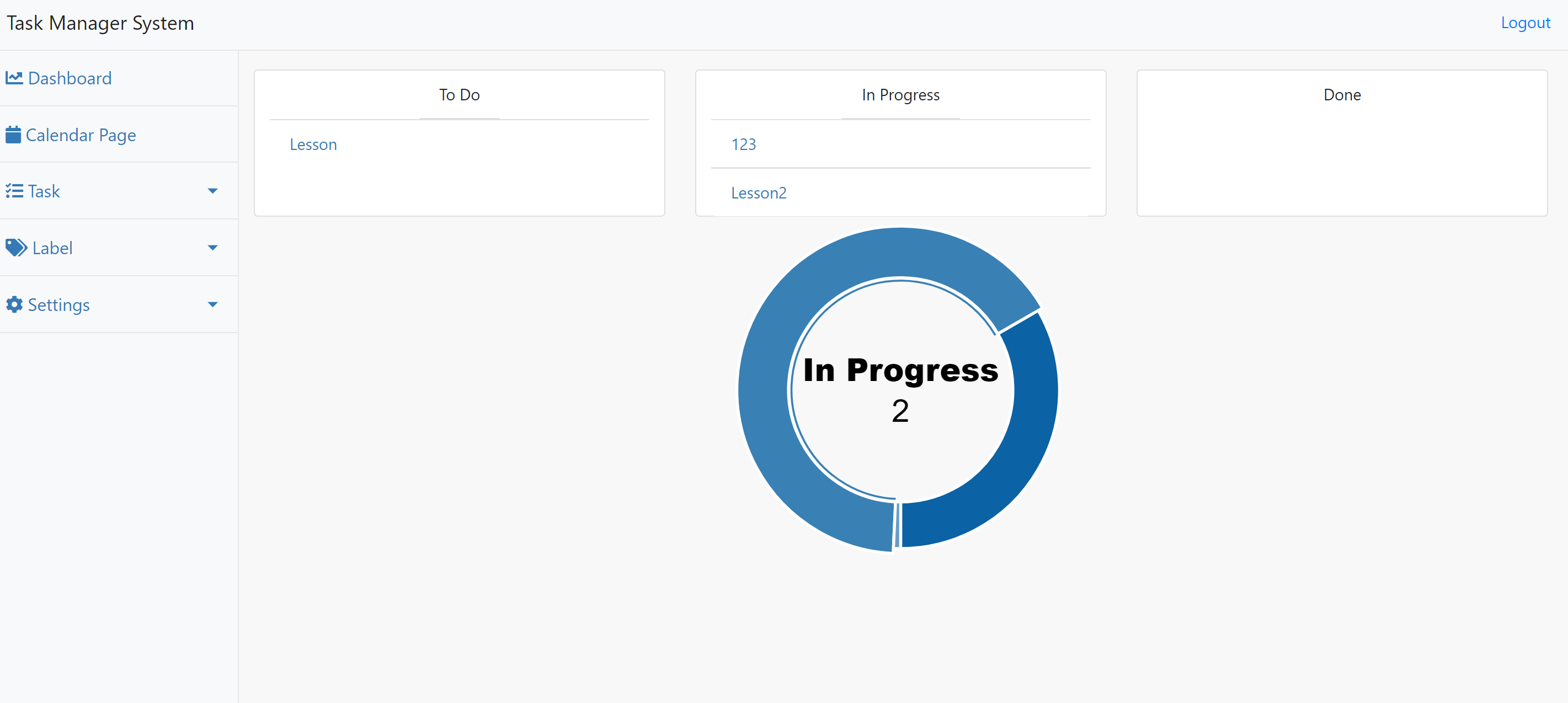


### 2.3.1 Опис програми

Після відкриття сайту нам відкривається авторизації/регистрації



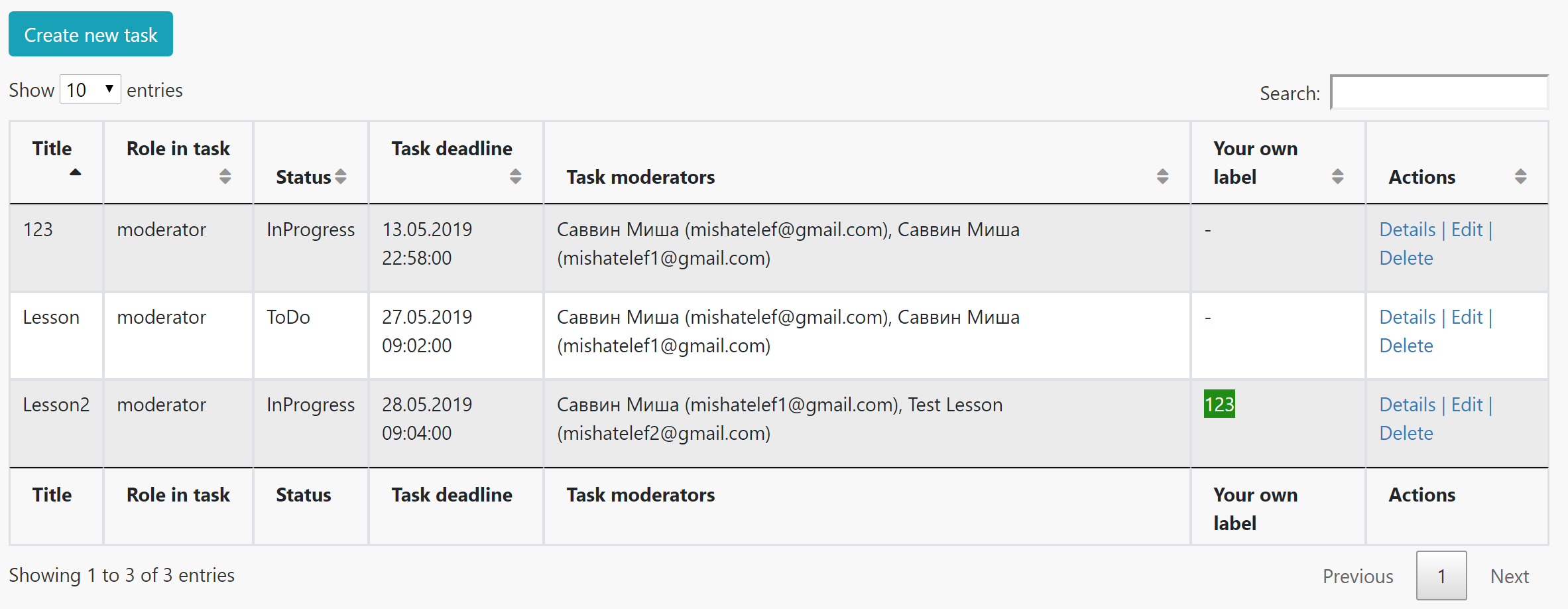


Далі ми відкривається головна сторінка 

В лівого боку буде меню де мі можемо обрати необхіодну сторінку для переходу

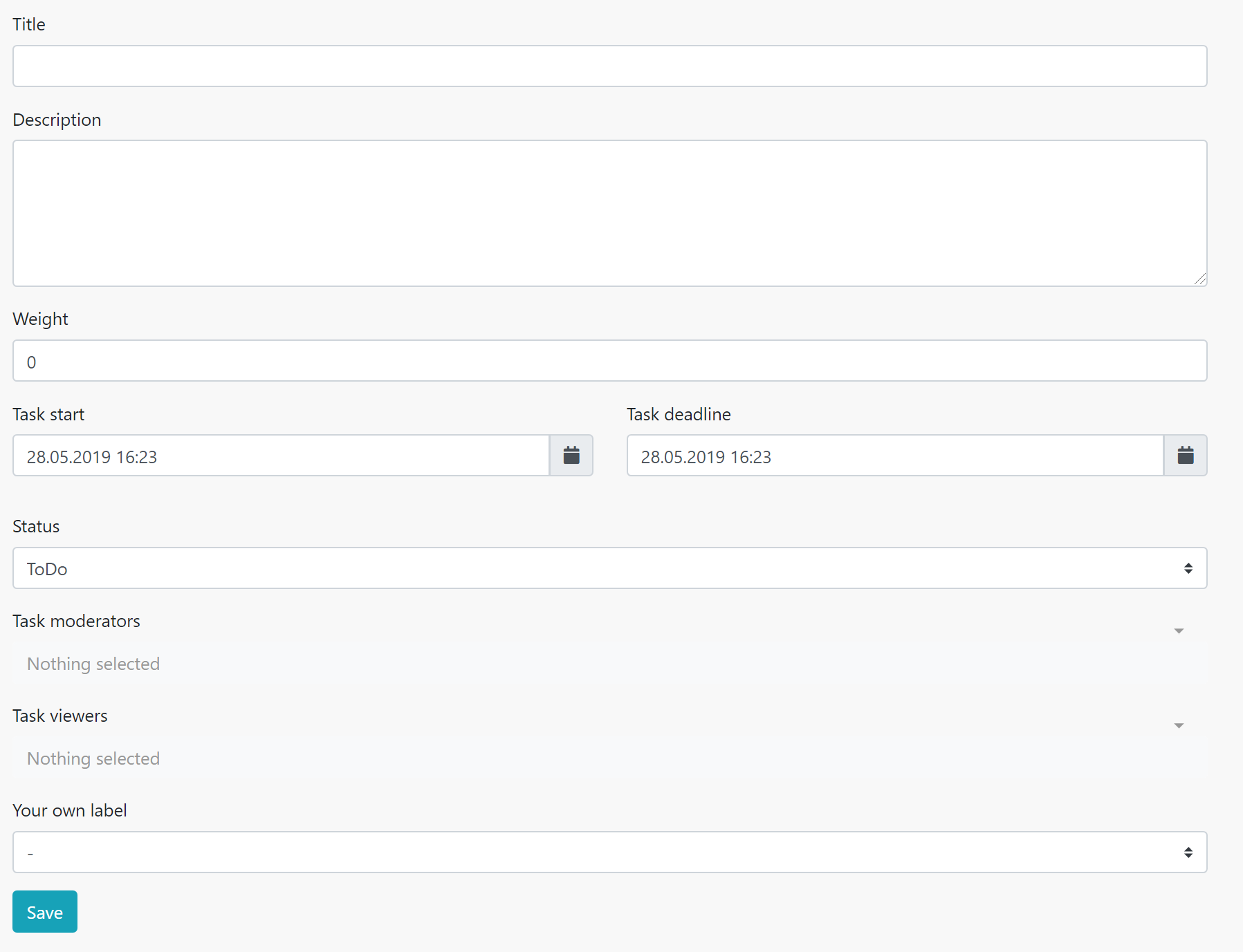


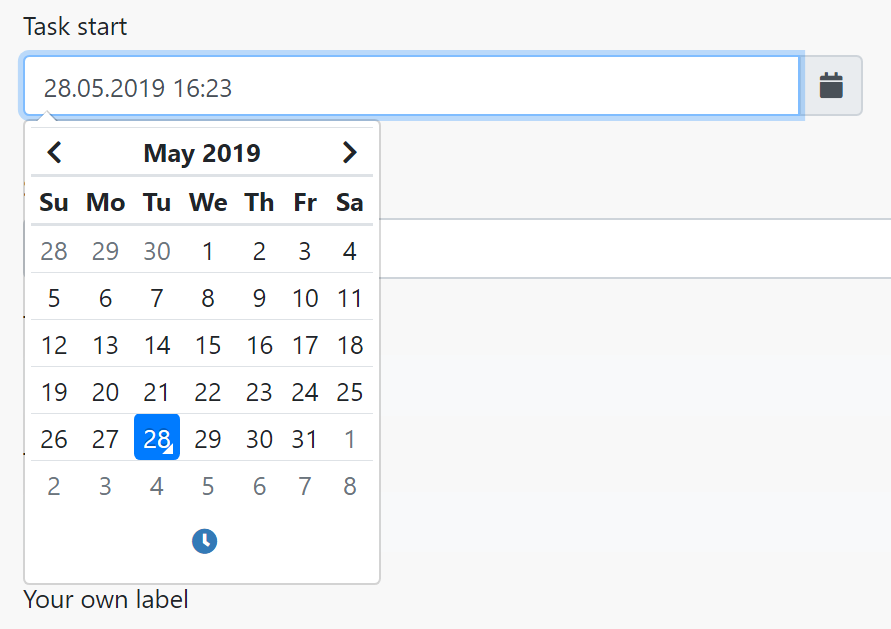
Приклад з таблицею “Задачі”

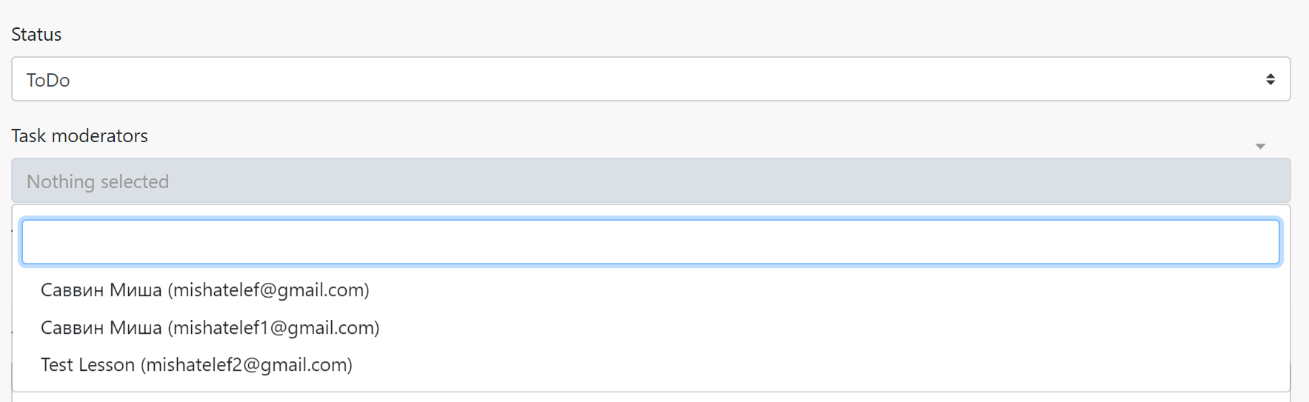


* *Додавання записів у таблицю*

Приклад з таблицею “ Задачі ”

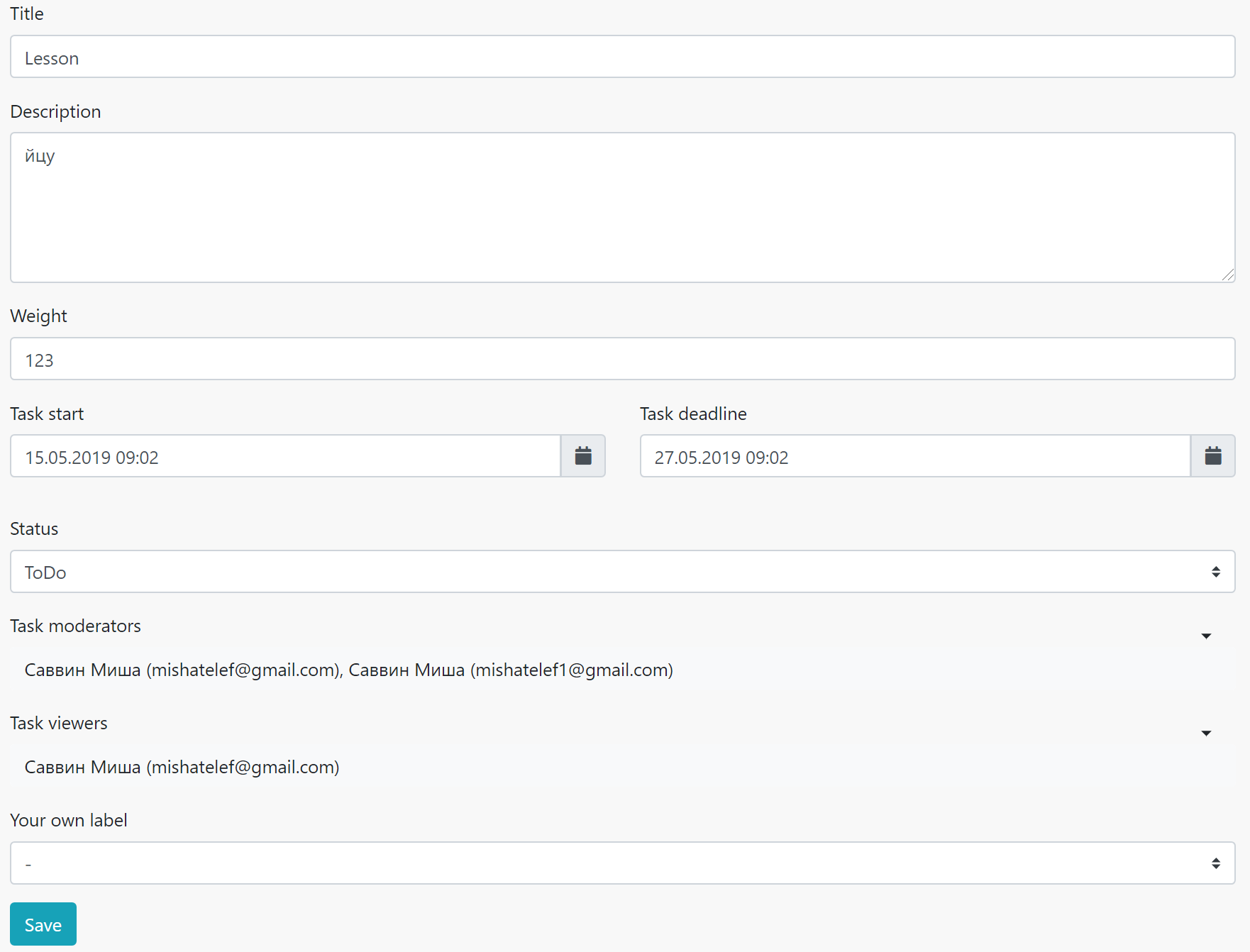




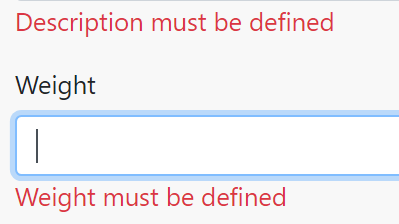


* *Зміна запису*

Спробуємо змінити запис



Спроба ввести некоректні дані

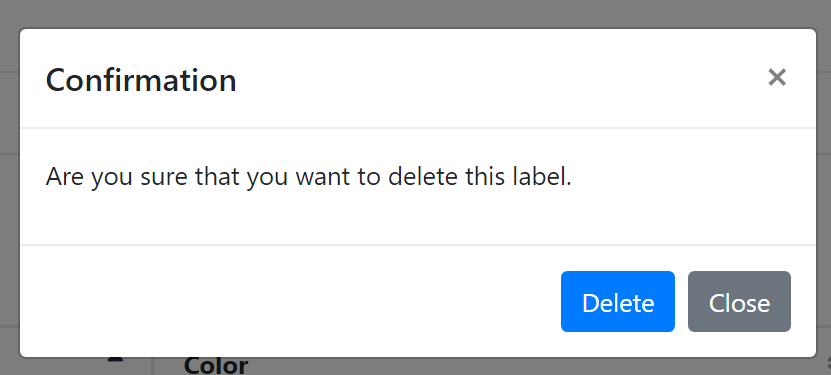


Перевірка на правильність введених даних відбувається у всіх таблицях. На front-end та на back-end

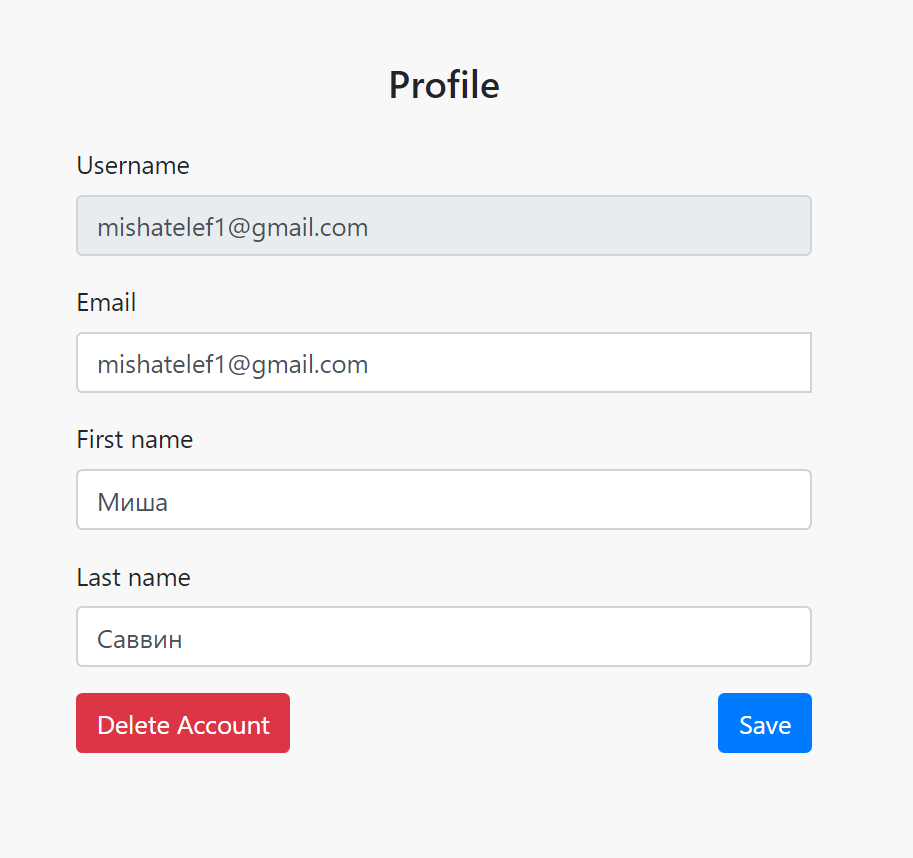
Всі маніпуляції у всіх інших таблицях аналогічні.

* *Видалення інформації*

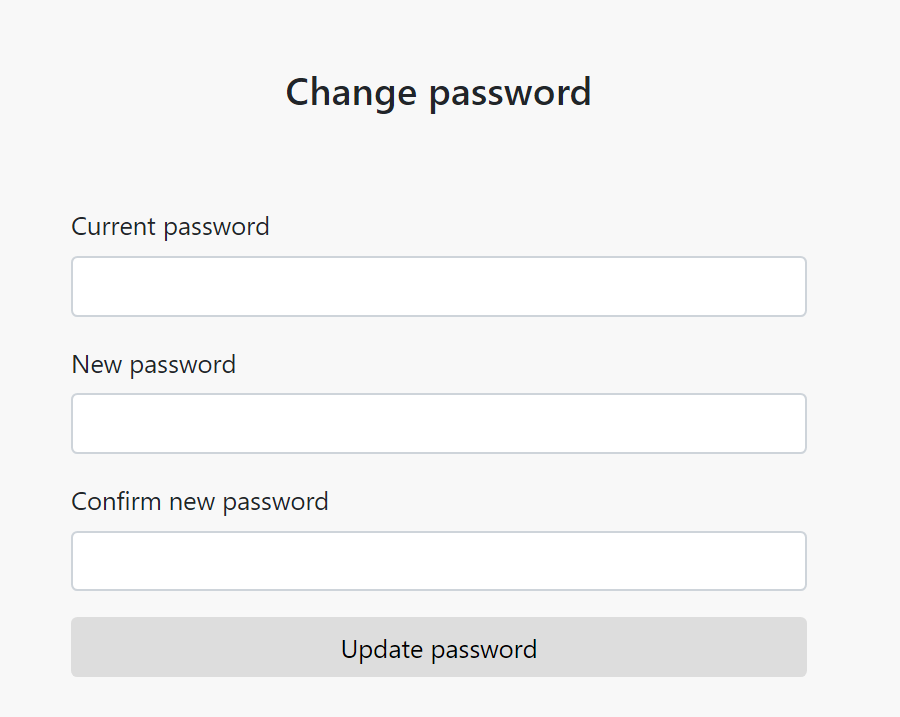




* *Редагування профайлу*

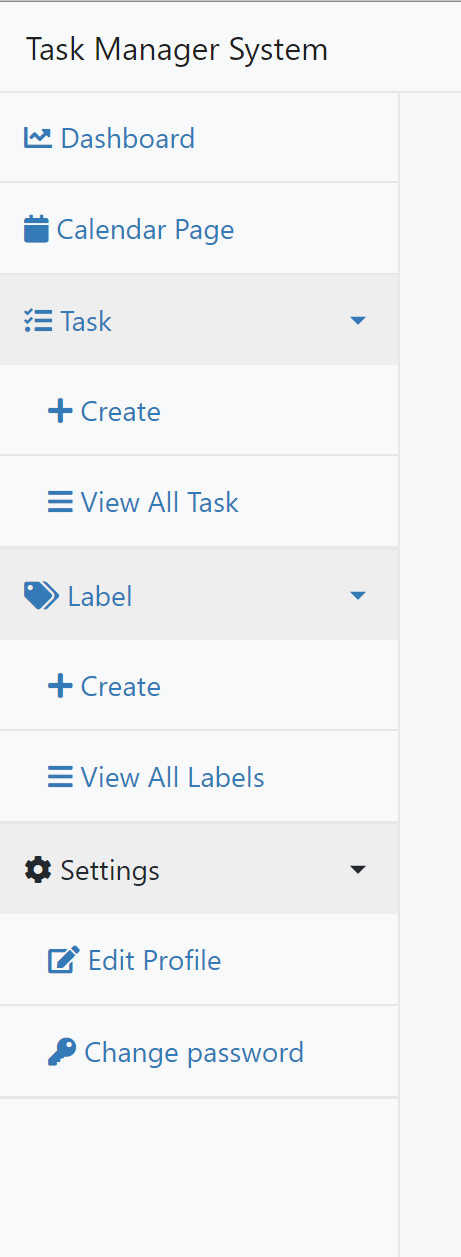


* *Зміна пароля*

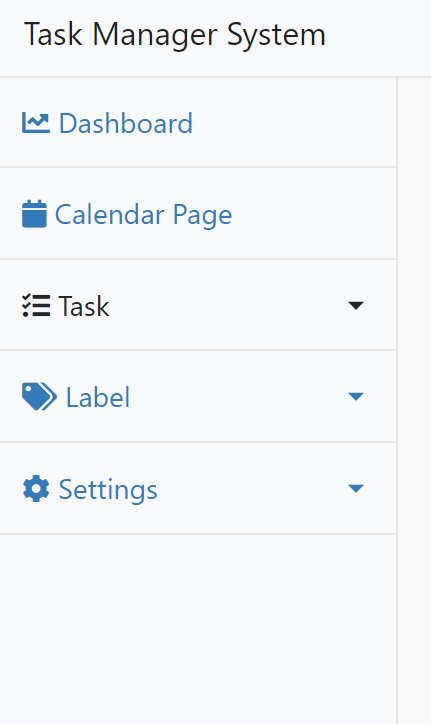


Навігаційне меню:

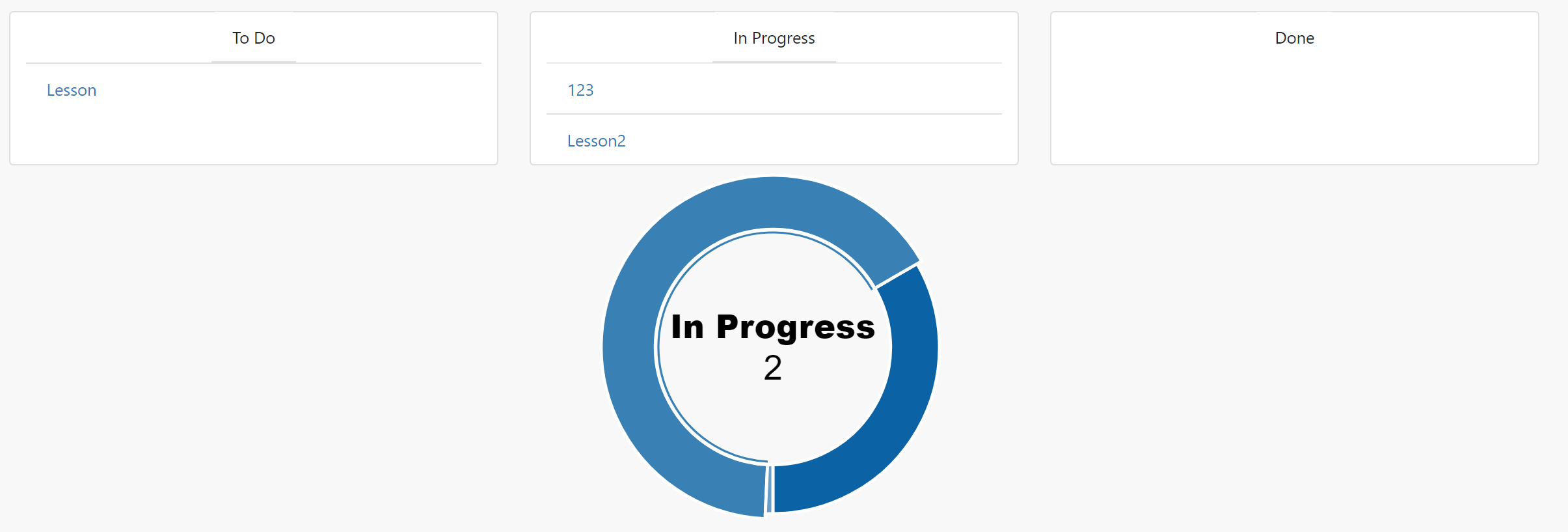
* *Навігаційне меню (розкрите)*



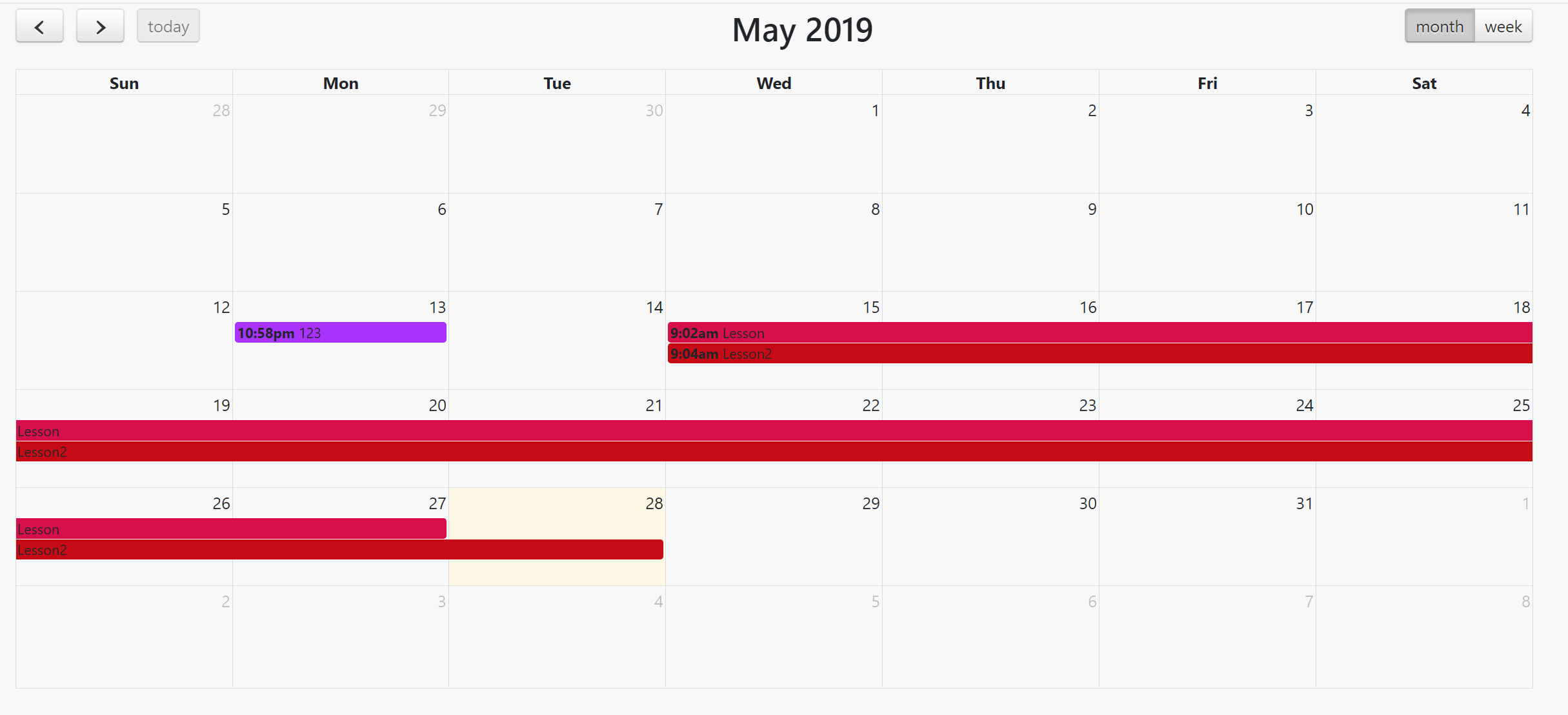
* *Навігаційне меню (скрите)*

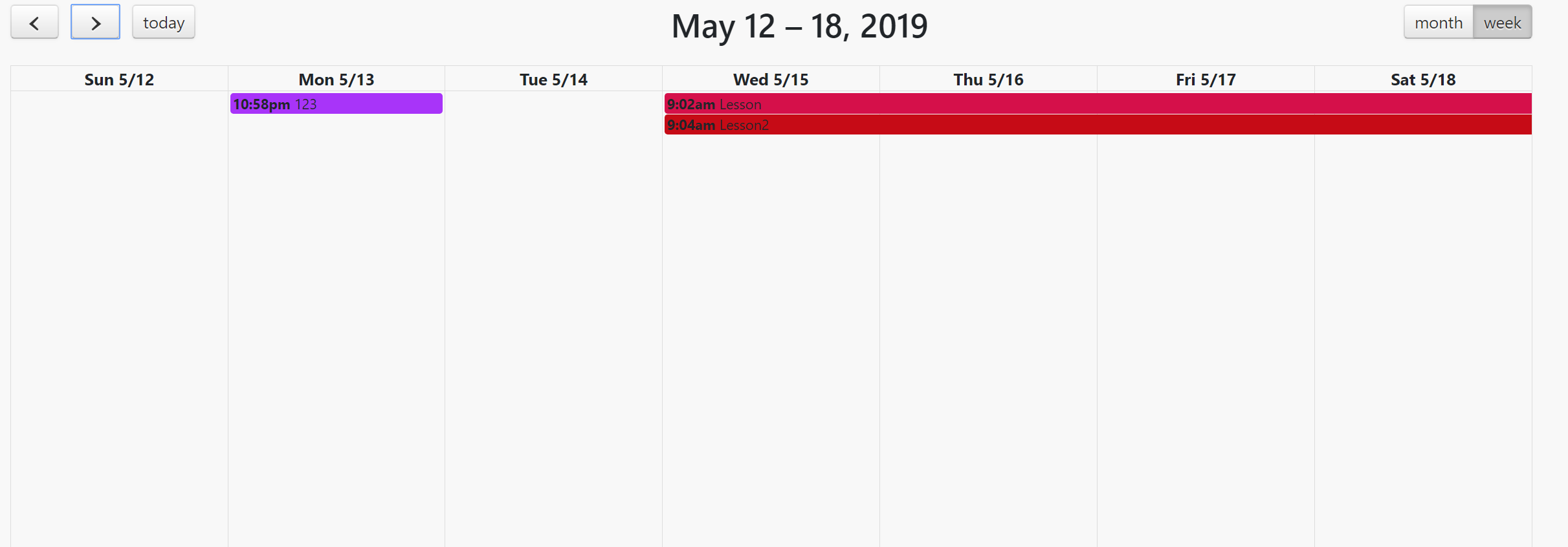


* *Дашбоард сторінка, задачі користувача за статусами*



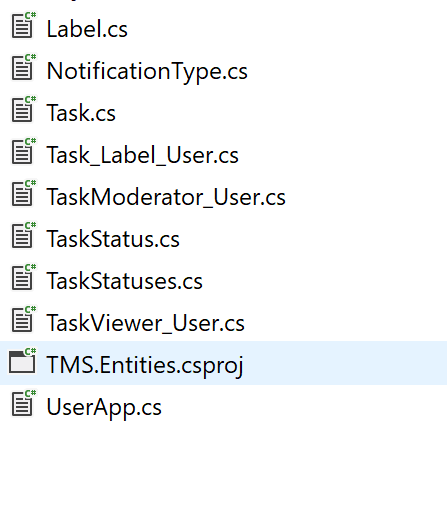
* *Календар, початок та кінець певної задачі за вибраний період*

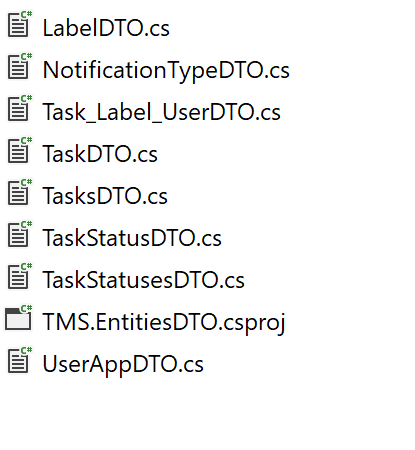


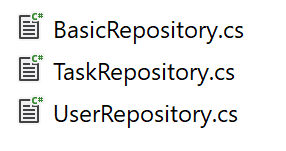


## 2.4 Використання та призначення основних функцій класів, тригерів та процедур

Всі створені файли для створення системи

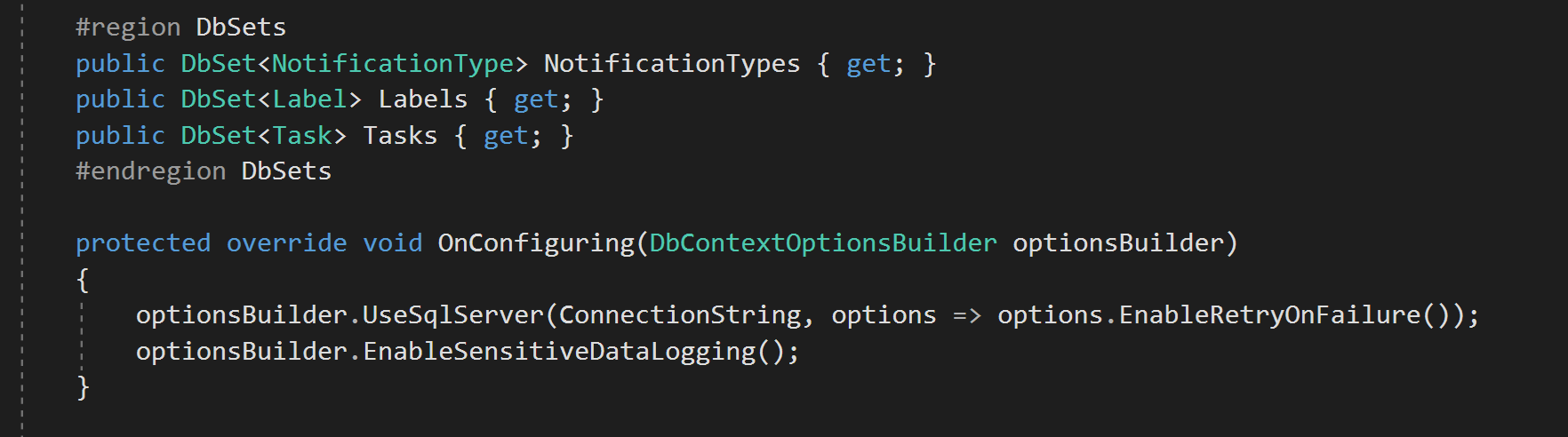




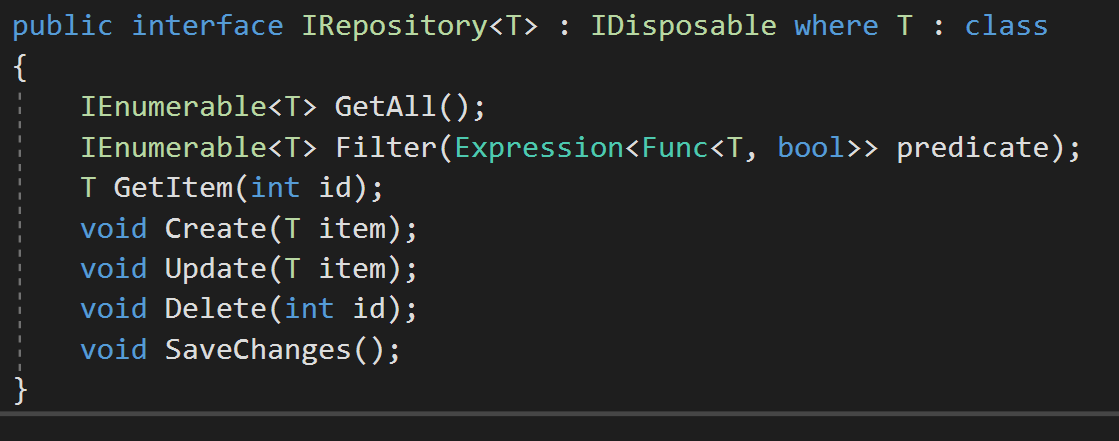


## 2.5 Cтворення запитів в середовищі розробки та результати їх виконання

Запроси у системі реалізовані за допомогою Entity Framework.

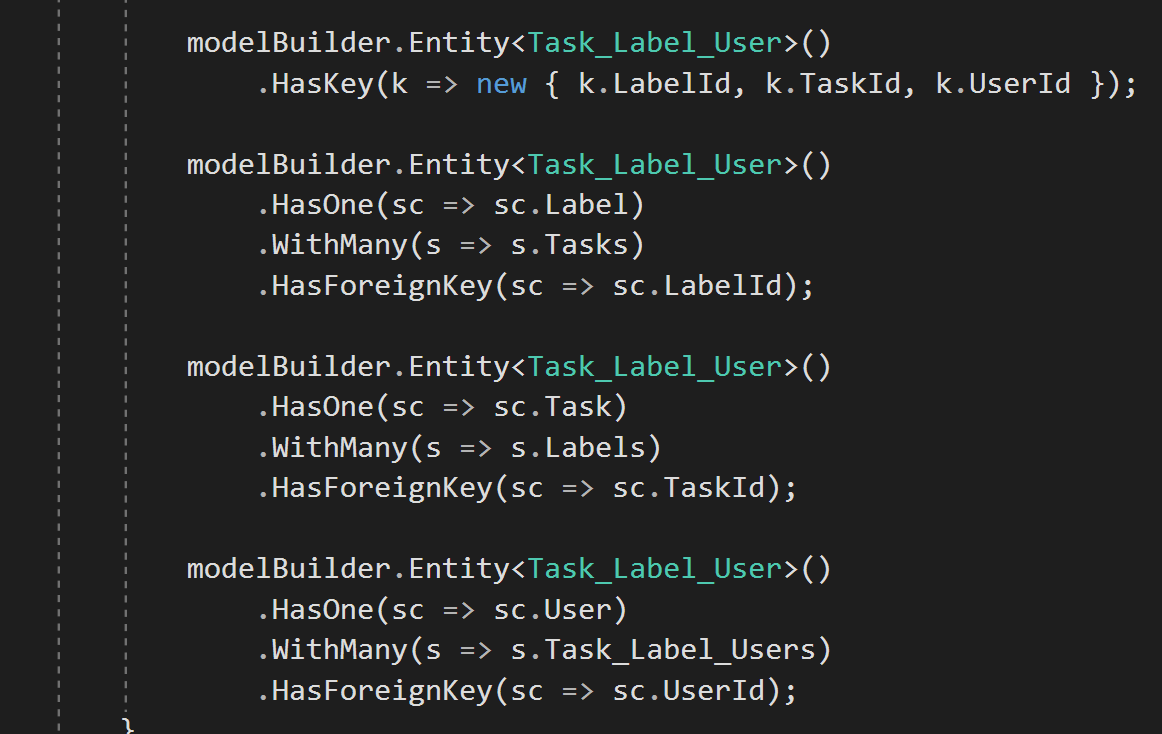


Як паттерн для роботи з даними використовувався IRepository



Для реєстрації залежностей використовуються ініціалізація у EF

Приклад: Для створення відношення many to many використовуються така ініціалізація у EF



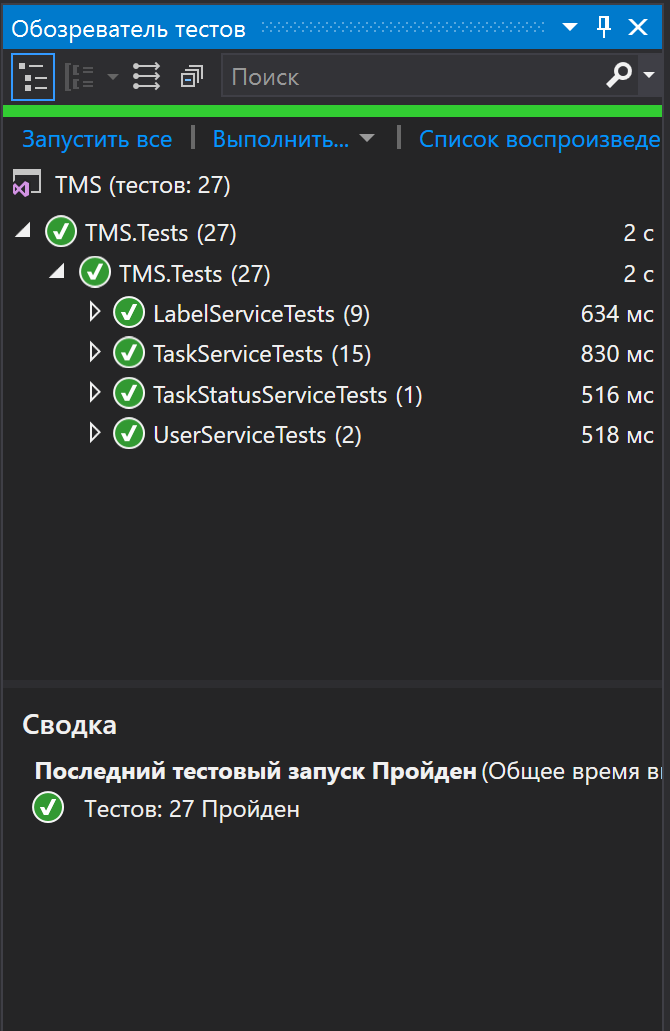
## 2.6 Інструкції по використанню користувачами за рівнями доступу

Інструкції щодо користування системою були розписані вище.

## 2.7 Опис результатів тестування та налагодження, рекомендації щодо супроводження

### 2.7.1 Тестування

Бізнес-логіка на 100% охоплюється модульними тестами. Moq був використаний для заміни Repository



Тестування пройшло успішно.

### 2.7.2 Рекомендації щодо супроводження

Ця система є досить невеликою і в майбутньому її можна розширювати і розробляти нові корисні функції.

# **Висновки**

* Створена система, яка описує предметну область «Система керування задачами»
* При проектуванні були виділені основні об'єкти предметної області та по ним створені таблиці і зв'язки між ними
* Для роботі з бд використовується EF 6 та паттерн IREPOSITORY
* Обране середовище створення БД – MSSQL + .NET
* Створено адаптивний інтерфейс за допомогою Bootstrap 4
* Наявна серверна на клієнтська валідація даних

# **Список літератури**

1. [Електронний ресурс] <https://bootstrap-4.ru/>
2. [Електронний ресурс] https://phppot.com/php/php-user-registration-form/
3. [Електронний ресурс] https://codewithawa.com/posts/complete-user-registration-system-using-php-and-mysql-database
4. [Електронний ресурс] http://bootstraptema.ru/stuff/templates\_bootstrap/1
5. [Електронний ресурс] <https://www.apachefriends.org/ru/index.html>
6. Коннолли Томас, Бегг К. Базы данных. Проектирование, реализация и сопровождение. 3-е изд. - М.: Вильямс, 2003. – 1440 с.
7. Клайн К.SQL. Справочник. - М.: КУДИЦ-ОБРАЗ, 2006. – 832 с.
8. Дейт К.Дж. Введение в системы баз данных. 8-ое изд. — М.: Вильямс, 2005. — 1328 с.