

7. Форми та автентифікація в Django.

Контрольні запитання

- 07.01. Для чого потрібні форми в Django?
- 07.02. Які є способи створення форм у Django?
- 07.03. Що таке поле форми і чим воно відрізняється від віджета поля?
- 07.04. Яким чином описати власну валідацію форми?
- 07.05. Яким чином увімкнути автентифікацію користувачів в Django?
- 07.06. Як здійснюється обмеження прав доступу для користувачів?

Завдання для аудиторної роботи

[A]07.01. Доповнити Django-проект бібліотеки, який описано в задачі [A]06.01. Додати автентифікацію та інтерактивну взаємодію зареєстрованих користувачів з даними. Проект має бути розміщений у Git-репозиторії. Кожен пункт повинен бути окремим комітом в Git-репозиторії з прикріпленим тегом, який можна додати командою

```
git tag <tagname> <commit-hash>
```

Для того щоб вивантажити на віддалений репозиторій коміти разом з тегами, потрібно виконати команду

```
git push origin <main-branch> --tags
```

11. Додати в проект групи користувачів: бібліотекар та читач.

- Створити новий застосунок **accounts**.
- У файлі **accounts/app.py** визначити метод **ready()**, який використовує Django-сигнал для створення груп

```
# accounts/signals.py
from django.db.models.signals import post_migrate
from django.dispatch import receiver
from django.contrib.auth.models import Group, Permission

@receiver(post_migrate)
def create_default_groups(sender, **kwargs):
    ...
```

- Бібліотекар має право створювати, видаляти та редагувати екземпляри книг бібліотеки.

12. Забезпечити автентифікацію користувачів.

- Створити форму реєстрації користувачів.
- При реєстрації користувач стає читачем.

- Створити форми входу та виходу із системи.
- Забезпечити у файлі **library/settings.py** перенаправлення на ці сторінки

```
LOGIN_URL = "/accounts/login/"
LOGIN_REDIRECT_URL = "/"
LOGOUT_REDIRECT_URL = "/accounts/login/"
```

13. Додати форму додавання та редагування екземпляра книги.

- Для перевірки чи користувач належить заданій групі, використати запис

```
{% if perms.appname.perm_name %}...{% endif %}
```

14. Додати функціонал видалення екземпляра книги.

Завдання для самостійної роботи

[B]07.01. Доповнити Django-проект блогу, який описано в задачі [B]06.01. Додати автентифікацію та інтерактивною взаємодією зареєстрованих користувачів з даними. Проект має бути розміщений у Git-репозиторії. Кожен пункт повинен бути окремим комітом в Git-репозиторії з прикріпленим тегом, який можна додати командою

```
git tag <tagname> <commit-hash>
```

Для того щоб вивантажити на віддалений репозиторій коміти разом з тегами, потрібно виконати команду

```
git push origin <main-branch> --tags
```

11. Додати в проект групу користувачів: модератор та автор.

- Створити новий застосунок **accounts**.
- У файлі **accounts/app.py** визначити метод **ready()**, який використовує Django-сигнал для створення груп

```
# accounts/signals.py
from django.db.models.signals import post_migrate
from django.dispatch import receiver
from django.contrib.auth.models import Group, Permission

@receiver(post_migrate)
def create_default_groups(sender, **kwargs):
    ...
```

- Автор має право залишати коментарі до статті, а також редагувати та видаляти тільки свої коментарі.
- Модератор має права автора, а також видаляти будь-які коментарі.

12. Забезпечити автентифікацію користувачів.

- Створити форму реєстрації користувачів.
- При реєстрації користувач стає автором.

- Створити форми входу та виходу із системи.

13.Провести зміну моделей.

- Додати поле користувача як автора статті або коментаря.
- Як в статті, так і в коментарі автор може залишитися анонімними.

14.Додати форму створення коментаря.

- Зокрема, додавати коментар, може незареєстрований користувач. В такому випадку форма повинна містити ім'я для автора коментаря, інакше ім'я береться з профіля автора.

15.Додати форму редагування коментаря.

16.Додати функціонал видалення коментаря.