Застосування фрейворку Django для побудови безпечної інформаційної системи управління доступом на прикладі медичної сфери

Марія Балобольченкова, студент[1], ORCID: 0009-0004-5732-1558,
Ольга Ізмайлова, канд. техн. наук, доцент [1] ,ORCID: 0000-0002-2905-1827

[1]*Київський національний університет будівництва і архітектури, Київ, Україна*

АНОТАЦІЯ

У статті розглянуто роль фреймворку Django у забезпеченні безпеки веб-застосунків на прикладі медичних інформаційних систем. Окреслено основні особливості архітектури MVT та переваги використання Django для роботи з конфіденційними даними. Детально проаналізовано механізми управління доступом, контроль дій користувачів та вбудовані засоби захисту від поширених веб-загроз. Особлива увага приділяється реалізації рольової моделі доступу, використанню адміністративної панелі та інструментам аудиту.

Ключові слова: управління доступом, Django, механізми захисту, шаблон MVC.

1. ВСТУП

У сучасних умовах, коли значна частина взаємодії користувачів і організацій відбувається через веб-застосунки, особливого значення набуває питання забезпечення їх надійним захистом. Управління та контроль доступу – це ключові аспекти безпеки веб-додатків. Саме за їх допомогою визначається, хто має право виконувати дії з даними та за яких умов. Особливо це важливо для медичних інформаційних систем, де обробляються персональні та чутливі дані пацієнтів, і потрібно дотримуватись вимог конфіденційності та міжнародних стандартів безпеки. Django, як сучасний веб-фреймворк, має вбудовані механізми автентифікації, авторизації та захисту від поширених загроз. Це робить його зручним інструментом для створення гнучких і безпечних систем управління доступом.

2. ФРЕЙМВОРК DJANGO

Мова програмування Python посідає провідне місце у сучасній розробці завдяки своїй простоті синтаксису, широкій екосистемі бібліотек та придатності для реалізації різноманітних завдань — від наукових обчислень і аналізу даних до створення веб-застосунків та систем штучного інтелекту. Для веб-розробки на Python існує кілька фреймворків, які відрізняються рівнем абстракції та сферою застосування. Серед них виокремлюється Django.

Django – це високорівневий фреймворк, що використовує мову програмування Python для веб-розробки. Він дотримується принципів «швидкої розробки» та «не повторюй себе». Його основною метою є надання розробникам інструментів для створення безпечних, масштабованих і функціональних веб-застосунків з мінімальними витратами часу та ресурсів.

Django дотримується архітектурного шаблону MVT (Model-View-Template), який є варіацією традиційного шаблону проектування MVC (Model-View-Controller), що використовується у веб-розробці. Цей шаблон розділяє застосунок на три основні компоненти:[1]

- моделі (model) відповідають за роботу з даними, їх зберігання та логіку (через об'єктно-реляційне відображення – ORM);
- представлення (view) реалізують бізнес-логіку та обробляють HTTP-запити;
- шаблони (Template) відповідають за відображення даних користувачеві за допомогою HTML-сторінок. Шаблони зазвичай складаються з HTML, CSS та JavaScript.

На рисунку 1 наведено схему архітектури MVT у Django, яка ілюструє взаємодію моделей, представлень та шаблонів у веб-застосунку.

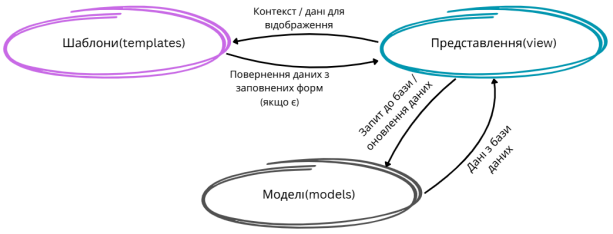

Рисунок 1. Архітектура MVT

Завдяки поєднанню гнучкості та надійності, Django широко застосовується для створення систем, що працюють із конфіденційними даними. У медичних інформаційних системах це особливо важливо, оскільки такі системи повинні не лише підтримувати зручну роботу з медичними записами, але й гарантувати захист персональних даних пацієнтів. Django забезпечує це завдяки:

- вбудованій системі автентифікації та авторизації, яка дозволяє створювати ролі (наприклад, лікар, медсестра, пацієнт, адміністратор) та керувати доступом до даних;
- механізмам ORM, які мінімізують ризик SQL-ін'єкцій та забезпечують безпечну роботу з базами даних;
- підтримці сучасних стандартів безпеки (CSRF-захист, захист від XSS, безпечне зберігання паролів тощо).

Таким чином, Django виступає не лише як інструмент для швидкої розробки веб-застосунків, а й як надійна основа для побудови медичних інформаційних систем, що відповідають високим вимогам безпеки та конфіденційності.

3. ЗАСОБИ БЕЗПЕКИ У ФРЕЙМВОРКУ DJANGO

3.1. Механізми управління доступом

У Django комплексний підхід до безпеки реалізовано через механізми управління доступом, контролю дій користувачів та вбудовані засоби захисту від поширених

загроз, що дозволяють будувати надійні та безпечні інформаційні системи.

Механізми управління доступом у Django забезпечують визначення та адміністрування прав користувачів і груп, що дозволяє ефективно контролювати доступ до ресурсів та функціоналу веб-застосунку. Вбудований модуль *django.contrib.auth* надає інструменти для створення користувачів, формування груп і ролей, а також призначення індивідуальних або групових дозволів (permissions)[2].

Django підтримує рольову модель доступу, яка особливо корисна у медичних інформаційних системах. За її допомогою можна з легкістю реалізувати такий розподіл користувачів на ролі:

- Лікар, який має доступ до медичних карток своїх пацієнтів, може вносити діагнози та призначати лікування;
- Медсестра, яка може редагувати лише певні записи, контролювати назначені процедур;
- Адміністратор системи, який керує всією системою, включно з правами користувачів і груп.
- Пацієнт, що має обмежений доступ до власних медичних даних, наприклад перегляд результатів аналізів чи історії лікування без можливості редагування.

Додатково Django надає вбудовану адміністративну панель, яка автоматично генерується на основі моделей застосунку. Адмін-панель дозволяє адміністраторам керувати користувачами, групами, ролями та правами доступу через зручний веб-інтерфейс, що спрощує реалізацію політик безпеки та знижує ймовірність помилок при налаштуванні доступу.

3.2. Механізми контролю доступу

Механізми контролю доступу у Django забезпечують перевірку прав користувачів під час виконання запитів та гарантують, що дії виконуються лише користувачами з відповідними дозволами. Для цього фреймворк надає широкий набір інструментів:

- декоратори (@login_required, @permission_required) для обмеження доступу до окремих функцій та представлень;
- кастомні перевірки для реалізації специфічних політик доступу, наприклад, обмеження перегляду медичних карток лише для лікарів конкретного відділення.

Додатково Django підтримує аудит дій користувачів через адміністративну панель. Увімкнення логування змін моделей дозволяє відстежувати, хто, коли і які зміни вніс у систему, що є важливим інструментом контролю доступу та забезпечення безпеки у медичних інформаційних системах. Наприклад, адміністрація може перевірити, який користувач редагував медичні записи пацієнтів, що допомагає запобігти зловживанням і забезпечити відповідність стандартам конфіденційності.

3.3. Інші механізми захисту

Окрім управління та контролю доступу, Django надає вбудовані механізми захисту від поширених веб-загроз, що робить його надійною платформою для побудови медичних інформаційних систем:[3]

- Захист від CSRF (Cross-Site Request Forgery) – Django автоматично додає CSRF-токени до форм і під час обробки POST-запитів перевіряє на відповідність CSRF-токена між клієнтом і сервером. Це запобігає виконанню шкідливих дій від імені користувача.
- Захист від XSS (Cross-Site Scripting) – шаблони Django за замовчуванням екрануютьHTML-символи - перетворюють спеціальні символи HTML у безпечний текст). Це унеможливлює виконання шкідливих скриптів у браузері користувача.
- Захист від SQL-ін’єкцій – ORM Django автоматично параметризує запити до бази даних, мінімізуючи ризик шкідливого втручання в SQL-запити.
- Безпечне зберігання паролів – паролі користувачів хешуються за допомогою сучасних алгоритмів (PBKDF2, Argon2, bcrypt), що забезпечує їх стійкість до злому
- Валідація та санітизація даних – форми і моделі Django автоматично перевіряють введені дані, знижуючи ризик некоректної або шкідливої інформації у системі.

У контексті медичних інформаційних систем ці механізми захисту набувають особливого значення, оскільки будь-яка вразливість може призвести до витоку чутливих даних пацієнтів. Використання Django дозволяє поєднувати управління доступом, контроль дій користувачів та комплексні технічні механізми захисту, забезпечуючи надійну і безпечну роботу системи.

4. ВИСНОВОК

Фреймворк Django завдяки архітектурі MVT, вбудованим механізмам автентифікації та авторизації, засобам захисту від поширених веб-загроз і підтримці сучасних стандартів безпеки є ефективним інструментом для створення медичних інформаційних систем.

Фреймворк надає можливість реалізувати рольову модель доступу, організувати зручне адміністрування користувачів через адміністративну панель, а також здійснювати базовий аудит дій, що підвищує рівень прозорості та контролю. Завдяки автоматичному екрануванню HTML-символів, захисту від CSRF- і XSS-атак, а також безпечному зберіганню паролів, Django забезпечує комплексний захист веб-застосунків.

Таким чином, використання Django дозволяє не лише скоротити час і ресурси на розробку, але й створити надійну основу для побудови медичних систем, що відповідають високим вимогам конфіденційності, безпеки та стандартам роботи з персональними даними пацієнтів.

Список літератури

[1] Django Project MVT Structure. 2025. UiRL: https://www.geeksforgeeks.org/python/django-project-mvt-structure/

[2] Django — Authentication — Django documentation. URL: https://docs.djangoproject.com/en/5.2/ref/contrib/auth/

[3] О. В. Братковський, А. М. Тушич. Удосконалення захисту даних у вебзастосунках із використанням DJANGO. 2024