

**Кучер В. В.**

*студент фізико-математичного факультету  
Житомирського державного університету імені Івана Франка*

**Науковий керівник: О. М. Кривонос**

*кандидат педагогічних наук, доцент,  
доцент кафедри прикладної математики та інформатики  
Житомирський державний університет імені Івана Франка*

## **ОСОБЛИВОСТІ ВИКОРИСТАННЯ ФРЕЙМВОРКУ DJANGO ДЛЯ НАПИСАННЯ ВЕБ-ДОДАТКІВ**

На сучасному етапі інформатизації всіх сфер життя, перед людством постають нові задачі, які вимагають простих та ефективних рішень. Так із

розвитком та поширенням мережі Інтернет виникає потреба в інструментах, які дозволяють створювати веб-додатки із раціональним використанням людських і часових ресурсів. Одним із таких інструментів є веб-фреймворк Django.

Django (Джанго) — високорівневий вільно поширювальний фреймворк нового покоління для розробки веб-додатків на мові програмування Python.[1] Для того щоб зрозуміти, навіщо при створенні сайту використовувати Django, подивімось як буде виглядати процес створення без цього фреймворку. Одним із найпростіших способів створення веб-додатків на Python з нуля – це скористатися стандартом Common Gateway Interface. CGI-сценарії – це виконувальні файли, які виконуються веб-сервером коли в URL запрошується відповідний сценарій. Стандарт CGI є інтерфейсом для з'єднання зовнішніх програм (в нашому випадку скриптів Python) з сервером через стандартні *stdin* і *stdout*. [2] Алгоритм написання найпростішого додатка, в загальному, виглядає наступним чином: створюємо сценарій на мові Python, який буде повертати HTML, зберігаємо його на веб-сервері з розширенням *\*.cgi* і відкриваємо цю сторінку в браузері.

Наведемо приклад CGI-сценарію, який виводить назви десяти останніх надрукованих статей із бази даних.

```
#!/usr/bin/env python3
import MySQLdb

print ("Content-Type: text/html\n")

print ("<html><head><title>Нові статті</title></head>")

print ("<body>")

print ("<h1>Нові статті:</h1>")

print ("<ul>")

DBConnect = MySQLdb.connect(user='Roman', passwd='RomaGo',
db='db')

cursor = DBConnect.cursor()
```

```
cursor.execute("SELECT title FROM articles ORDER BY publ_date
DESC LIMIT 10")

for item in cursor.fetchall():
    print("<li>%s</li>".format(item[0]))

print("</ul>")

print("</body></html>")

DBConnect.close()
```

Для створення однієї динамічної сторінки цей спосіб є прийнятним, але він має цілий ряд проблем і незручностей. Серед них:

**Дублювання коду.** Якщо необхідність синхронізації з базою даних виникає в різних місцях програми, спричиняє небажане дублювання коду в кожному із CGI сценаріїв.

**Зниження продуктивності роботи програміста.** Через необхідність пам'ятати про вивід HTTP заголовків і по необхідності закривати з'єднання з БД програміст повинен писати повторюваний у багатьох місцях код, що створює зайву можливість зробити помилку в ньому і знижує загальну продуктивність написання такого коду. Такого роду ініціалізації та закриття краще доручити спеціальним компонентам системи.

**Незручність розширення і підтримки.** Якщо код буде застосовуватись у різних умовах, наприклад буде змінюватись БД і пароль доступу до неї, тоді виникає потреба у створенні спеціальних конфігураційних файлів. Значною проблемою стане зміна дизайну сайту, адже HTML шаблон і бізнес-логіка знаходяться в одному місці, що значно збільшує ймовірність того, що дизайнер не знаючи мови програмування Python порушить цілісність алгоритму через неправильно набраний символ, тощо.

Саме такі задачі призваний вирішувати Django. Який надає набір взаємопов'язаних компонентів для створення веб-додатків, допомагаючи

зосередитися на написанні зрозумілого, зручного для підтримки та розширення сайту.[3] Django додаток складається із декількох модульних частин, які структурно поділені між собою. Проект Django зароджувався як інструмент для швидкої розробки новинних сайтів, що залишило значний відбиток на його архітектурі. Тому він надає деякі засоби для розробки й експлуатації інформаційних ресурсів. Серед них: вбудований адміністративний інтерфейс для керування моделями даних. Він надає змогу редагувати, створювати, видаляти будь який об'єктний вміст сайту.

Через вдалу архітектуру Django був портований на інші мови програмування, пропонуючи багато платформну підтримку. Деякі проекти скопіювали основну архітектуру, інші – взяли лише деякі концепції. Прикладами є: Swig (JavaScript), Liquid (Ruby), Template::Swig (Perl), Twig (PHP), Jinja (Python), ErlyDTL (Erlang).[4]

Серед основних переваг фреймворку можна виділити такі:

**Object-relational mapping** (Об'єктно-реляційне відображення) – це технологія програмування, яка дозволяє перетворювати несумісні типи моделей в ООП, наприклад, між сховищем даних і об'єктами програмування. ORM використовується для спрощення процесу збереження об'єктів в реляційну базу даних і їх вилучення, при цьому ORM сам дбає про перетворення даних між двома несумісними станами. Загальна схема роботи показана на рисунку 1.



Рис. 1 – Схема роботи ORM.

Після створення моделі, Django автоматично створює API для роботи з базою даних, який дозволяє створювати, змінювати, отримувати і видаляти об'єкти.

**Автоматична побудова інтерфейсу для адміністрування (Django CMS).** Після створення нового проекту Django автоматично створює інтерфейс для адміністрування моделей. В даному інтерфейсі можна створювати, видаляти та змінювати об'єкти відповідно до описаних моделей.

**Елегантні URL.** Уніфікований локатор ресурсів або адреса ресурсу (англ. Uniform Resource Locator, URL) – стандартизована адреса певного ресурсу в інтернеті. Наприклад, HTML-сторінки, аудіофайлу, тощо.

Оскільки Django – Python фреймворк, відповідно на нього розповсюджуються принципи написання Python коду – описані в The Zen of Python. І першим принципом є «Красиве є краще за потворне».[5] Що і наслідує система Django URL dispatcher, яка дозволяє створювати «красиві й елегантні» адреси сторінок, які позбавленні від закінчень .php .html.

**Розширювана система шаблонів з тегами та наслідуванням.** Шаблон в Django зазвичай застосовується для створення HTML розмітки. В шаблоні можуть зустрічатися маркери та прості логічні конструкції (шаблонні теги), що керують відображенням документа. Наступний шаблон Django описує HTML сторінку, на якій відображається подяка користувачу, який розмістив замовлення.

```
<html>
<head><title>Повідомлення про Ваше замовлення</title></head>
<body>
<h1>Повідомлення про Ваше замовлення</h1>
<p>Шановний(на) {{ person_name }}!</p>
<p>Дякуємо, що зробили замовлення в {{ company.name }}. Воно
буде виконане {{ done_date|date:"F j, Y" }}.</p>
<p>Замовленні Вами товари:</p>
<ul>
{% for item in list %}
  <li>{{ item.name }}</li>
{% endfor %}
```

```
</ul>  
</body>  
</html>
```

Шаблонізатор підтримує вставку змінних, логічних конструкцій, фільтрів безпосередньо в HTML шаблон.

**Повна документація.** Офіційний сайт Django ([djangoproject.com](http://djangoproject.com)) включає в собі повну документацію по всьому фреймворку. Для зручності, документація поділена на частини, які відповідають основним частинам. Також є алфавітний покажчик, і навігація по модулям. Документація регулярно оновлюється, і тому там можна знайти відповіді на всі основні питання.

**Велика спільнота розробників по всьому світу.** Тисячі розробників по всьому світу розробляють свої проекти на Django. Тому якщо й виникають труднощі у розумінні деяких речей, то знайти потрібні відповіді допоможуть тематичні блоги розробників, форуми, Q&A сервіси, тощо. Розробники завжди готові поділитися своїм досвідом і дати поради новачкам щодо реалізації певних речей.

Отже, Django є ефективним інструментом, який дозволяє швидко та зручно створювати різноманітні веб-додатки, від звичайних сайтів-візиток до масштабних високонавантажених проектів.

#### **Список використаних джерел:**

1. Django [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Django> – Назва з екрана.
2. Python CGI программирование [Електронний ресурс] – Режим доступу: [http://www.w3ii.com/ru/python/python\\_cgi\\_programming.html](http://www.w3ii.com/ru/python/python_cgi_programming.html) – Назва з екрана.
3. Головатый А., Каплан-Мосс Дж. Django. Подробное руководство, 2-е издание. – Пер. с англ. – СПб.: Символ- Плюс, 2010. – 560 е., ил

4. Django (web framework) [Электронный ресурс] – Режим доступа:  
[https://en.wikipedia.org/wiki/Django\\_\(web\\_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework)) – Назва з екрана.
5. The Zen of Python [Электронный ресурс] – Режим доступа:  
<https://www.python.org/dev/peps/pep-0020/> – Назва з екрана.