

ПОРІВНЯЛЬНИЙ АНАЛІЗ DJANGO, FLASK ТА FASTAPI

Солохін А. Є.

Науковий керівник – д.т.н, проф. Смеляков К. С.

Харківський національний університет радіоелектроніки, каф. ПІ

м. Харків, Україна

e-mail: artem.solokhin@nure.ua

This comprehensive comparative analysis delves into the intricate features and functionalities of Django, Flask, and FastAPI. By meticulously scrutinizing their attributes, this analysis aims to offer a nuanced understanding of their capabilities, strengths, and limitations across various parameters. The evaluation encompasses crucial aspects such as scalability, ease of use, and performance, providing invaluable insights to developers navigating the labyrinth of framework selection. With a keen focus on deciphering how these frameworks measure up against each other, this analysis serves as a guiding beacon, empowering developers to make well-informed decisions tailored to the unique requisites of their projects.

Процес створення веб-додатків з часом еволюціонував і став простішим [1]. У світі веб-розробки існує безліч інструментів та технологій, але Django, Flask та FastAPI вирізняються як одні з найпопулярніших фреймворків для розробки веб-додатків на Python.

Вибір між Django, Flask та FastAPI визначається потребами та характеристиками кожного конкретного проєкту веб-розробки. Проблема стає важливою через велику різноманітність цих фреймворків, яка дозволяє розробникам вибрати оптимальний інструмент залежно від розміру та складності проєкту, а також власних уподобань у роботі. Від вибору фреймворку залежить ефективність розробки, швидкість розгортання та загальна продуктивність веб-додатку, роблячи це питання ключовою у сфері веб-розробки.

За допомогою Django ви можете перевести веб-додатки від концепції до запуску за лічені години. Django бере на себе більшу частину клопоту веб-розробки, тому ви можете зосередитися на написанні свого додатка без потреби винаходити колесо. Це безкоштовно та з відкритим кодом [2].

Django, один з найдосконаліших фреймворків Python, вражає своєю повнотою та готовністю до використання. Основні переваги включають вбудовану адмін-панель, ORM (Object-Relational Mapping) для легкого взаємодії з базою даних, і високий рівень безпеки. Django підтримує традиційну архітектуру Model-View-Controller (MVC) та має широкий спектр розширень для розширення функціоналу.

Однак його невелика гнучкість може бути недоліком у випадках, коли потрібна більша вільність у виборі компонентів.

Flask – це легкий веб-фреймворк для застосунків, що використовують WSGI. Він призначений для швидкого і простого старту, а також може масштабуватися до складних додатків. Почавши як простий обгортка навколо Werkzeug і Jinja, він став одним із найпопулярніших фреймворків для веб-розробки на Python. Flask пропонує поради, але не нав'язує жодних залежностей чи структури проєкту. Розробнику залишається вибір інструментів та бібліотек, якими він хоче користуватися. За допомогою розширень, які розробляє спільнота, легко додати новий функціонал. Хоча Flask надає вищу вільність, розробка складніших проєктів може вимагати більше ручної роботи [3].

FastAPI – це новаторський фреймворк, спроектований для максимальної продуктивності та швидкості. Однією з ключових особливостей є автоматична генерація документації на основі анотацій Python. FastAPI підтримує стандарти OpenAPI та JSON Schema, що робить його ідеальним вибором для розробників, які цінують автоматизовану документацію та переваги валідації даних.

Недоліками FastAPI можуть бути складніша настройка та менша кількість готових розширень порівняно з Django та Flask.

Вибір між Django, Flask та FastAPI визначається специфікою проєкту та його унікальними потребами. Django, будучи повнофункціональним фреймворком, ідеально підходить для великих та складних програм, де вбудовані рішення, такі як ORM та адміністративний інтерфейс, полегшують розробку та масштабування.

Flask, навпаки, пропонує легкість та гнучкість. Це ідеальний вибір для невеликих проєктів або ситуацій, де розробник хоче більше контролю над вибором бібліотек та компонентів. Flask надає базовий фундамент, залишаючи розробнику свободу вибору та конфігурації.

FastAPI відзначається своєю високопродуктивністю в галузі розробки API та програм реального часу. Зокрема, він підтримує асинхронність, що дозволяє ефективно взаємодіяти з багатьма запитами одночасно, що робить його відмінним вибором для проєктів із високими потребами в паралелізмі.

За допомогою експерименту було вивчено та оцінено різні аспекти цих фреймворків, такі як швидкість, продуктивність, легкість використання та масштабованість.

Експеримент був побудований на основі створення трьох окремих веб-додатків, кожен з яких реалізований за допомогою відповідного фреймворку – Django, Flask або FastAPI. Для кожної системи було розроблено однаковий набір функціональності, щоб забезпечити об'єктивність порівняння.

Після створення веб-додатків проводилася серія тестів та вимірювань різних показників, таких як час відповіді сервера, кількість запитів,

продуктивність обробки запитів тощо. Результати тестів аналізувалися та порівнювалися для кожного фреймворку.

Цей експеримент є цінним дослідженням, оскільки він надає об'єктивні дані та порівняння між різними веб-фреймворками Python. Результати можуть бути корисними для розробників, які мають вибрати найбільш підходящий фреймворк для свого проекту. Крім того, експеримент дозволяє краще зрозуміти сильні та слабкі сторони кожного фреймворку, що допомагає в прийнятті обґрунтованих рішень при розробці веб-додатків.

При виборі фреймворка слід орієнтуватися на конкретні потреби та характеристики проекту, такі як його розмір та складність. Важливо враховувати рівень гнучкості, що потрібен для реалізації унікальних вимог проекту, а також обране спрощення чи повноту функціоналу, які надає конкретний фреймворк. Врахування швидкості розробки, підтримки та активності спільноти також може бути вирішальним фактором при виборі. Кінцевим чинником є особисті вподобання розробника та його досвід з конкретними фреймворками, що може впливати на продуктивність та задоволення від роботи над проектом.

Список використаних джерел:

1. Kyrychecenko I., Pronina D. Comparison of Redux and React Hooks Methods in Terms of Performance, CEUR Workshop Proceedings, 2022, 3171, pp. 791–800.
2. Uzayr S. b. Mastering Django: A Beginner's Guide. Taylor & Francis Group, 2022, 322 p.
3. Flask | Read the Docs. Full featured documentation deployment platform – Read the Docs. URL: <https://readthedocs.org/projects/flask/> (дата звернення: 04.03.2024).