

ДОДАТОК А

Перелік джерел посилання за науковими напрямками керівника та науковців
кафедри програмної інженерії

10. Davydova, I., Marina, O., Peleshchyshyn,, A., Serhii Marin. Webometric Analysis of National Libraries Websites 2nd International Workshop on Control, Optimisation and Analytical Processing of Social Networks (COAPSN-2020), Lviv, Ukraine, May 21, 2020. – Lviv, 2020. – P. 165-176

12. Krakowczyk D.G., Reich D.R., Chwastek J., Jakobi D.N., Prasse P., Süss A., Turuta O., Kasprowski P., Jäger, Lena A. A. Pymovements: A Python Package for Eye Movement Data Processing. Eye Tracking Research and Applications Symposium (ETRA), 2023, 53.

ДОДАТОК Б

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Ім'я користувача:
Кардаш Євген Вікторович каф.ПІ

Дата перевірки:
12.06.2024 07:05:17 EEST

Дата звіту:
12.06.2024 07:27:53 EEST

ID перевірки:
1016350550

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100013622

Назва документа: 2024_М_ПІ_ІПЗм_22_5_Солохін_А_Є_скорочений

Кількість сторінок: 64 Кількість слів: 9403 Кількість символів: 68186 Розмір файлу: 4.12 MB ID файлу: 1016152770

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

9.96%
Схожість

Найбільша схожість: 4.83% з джерелом з Бібліотеки (ID файлу: 1012773300)

4.62% Джерела з Інтернету 534 Сторінка 66

8.43% Джерела з Бібліотеки 111 Сторінка 70

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування 16 сторінок

ДОДАТОК В

Слайди презентації



Дослідження ефективності
використання фреймворків
веб-розробки на
продуктивність веб-додатку:
порівняльний аналіз Django, Flask, та
FastAPI



Солохін Артем Євгенович, ПЗМ-22-5
Науковий керівник: професор, Смеляков С.В.

18 червня 2024

Рисунок В.1 – Титульний слайд

Дослідження

Актуальність: Веб-розробка стрімко розвивається, збільшуючи попит на швидкі та надійні веб-додатки. Вибір фреймворку суттєво впливає на продуктивність та зручність розробки.

Напрямок дослідження: Порівняльний аналіз продуктивності та ефективності Django, Flask і FastAPI для визначення оптимального фреймворку.

Об'єкт дослідження: Веб-додатки, створені з використанням Django, Flask та FastAPI, та їх продуктивність у різних сценаріях.



Рисунок В.2 – Слайд опису дослідження

Огляд літератури (аналогів)

Основні джерела:

1. Офіційні документації
2. <https://blog.jetbrains.com/pycharm/2023/11/django-vs-flask-which-is-the-best-python-web-framework/>
3. <https://blog.jetbrains.com/pycharm/2023/12/django-vs-fastapi-which-is-the-best-python-web-framework/>

Прогалини у наявних дослідженнях:

1. Недостатність порівняльних аналізів
2. Відсутність масштабованих тестів
3. Вплив нових версій



Рисунок В.3 – Слайд «Огляд літератури»

Постановка задачі

Проблема: Вибір найбільш ефективного фреймворку веб-розробки (Django, Flask, FastAPI) для забезпечення оптимальної продуктивності веб-додатку.

Очікувані результати:

1. Визначення сильних та слабких сторін кожного фреймворку.
2. Рекомендації щодо вибору фреймворку залежно від типу та вимог веб-додатку.
3. Підвищення ефективності розробки та продуктивності веб-додатків.



Рисунок В.4 – Слайд «Постановка задачі»

Методологія

Методи дослідження:

- Експериментальний аналіз
- Порівняльний аналіз
- Опитування

Технології:

- Django, Flask, FastAPI
- Python
- Pycharm
- PostgreSQL
- Docker
- HTML, CSS



Рисунок В.5 – Слайд опису методології

Архітектура система для проведення експериментального дослідження

Ключові компоненти:

1. Клієнтський інтерфейс
2. API Сервер
3. База даних
4. Контейнеризація та розгортання

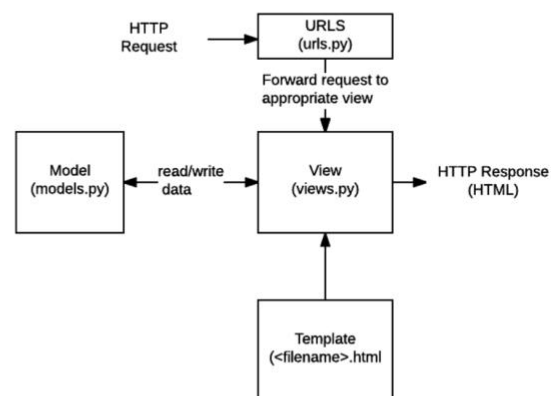


Рисунок В.6 – Слайд з описом архітектури

Опис програмного забезпечення, що було використано у дослідженні

Процес розробки:

1. Аналіз вимог та проектування
2. Розробка базової архітектури для кожного фреймворку
3. Розробка основних функцій
4. Тестування продуктивності



Рисунок В.7 – Слайд «Опис процесу розробки»

Зміст проведеного експерименту

Методи: Експериментальний аналіз, Опитування

Вхідні дані: Набори даних для тестування

Критерії: Швидкодія, Масштабованість, Підтримка асинхронності, Безпека, Споживання пам'яті, Розробка API

Послідовність: Розробка прототипів, Налаштування середовища тестування, Виконання тестів продуктивності, Аналіз отриманих результатів

Вимірювання: Використання ресурсів, Час відповіді



Рисунок В.8 – Слайд «Зміст проведеного експерименту»

Результати експерименту

Фрейм	Експерт										Сума	Вага кри.	X	σ	V
	1	2	3	4	5	6	7	8	9	10					
Django	3	3	3	2	3	2	2	3	3	3	27	0,45	2,70	0,48	0,18
Flask	2	1	2	3	1	3	1	1	2	2	18	0,30	1,80	0,79	0,44
FastAPI	1	2	1	1	2	1	3	2	1	1	15	0,25	1,50	0,71	0,47
Швидкодія															

$$R_1 (\text{Django}) = 0,342$$

$$R_2 (\text{Flask}) = 0,352$$

$$R_3 (\text{FastAPI}) = 0,3195$$

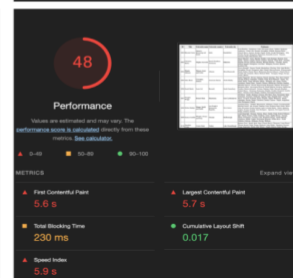
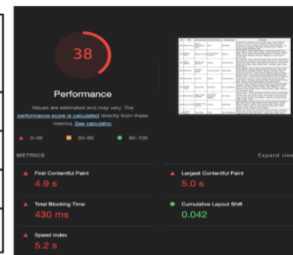
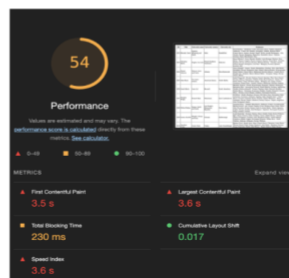


Рисунок В.9 – Слайд «Результати експерименту»

Аналіз отриманих результатів

Django:

Сильні сторони: Включає повний набір функцій, відмінна масштабованість та безпека, велика спільнота та детальна документація.

Слабкі сторони: Висока складність для новачків та можлива затримка через монолітну структуру.

Flask:

Сильні сторони: Простий у використанні, гнучкий, підходить для швидкого створення прототипів, легко розширюється за допомогою плагінів.

Слабкі сторони: Обмежені функції з коробки, менше вбудованих безпекових функцій, менш детальна документація.

FastAPI:

Сильні сторони: Висока продуктивність завдяки асинхронним можливостям, підтримка нових версій Python, автоматична генерація документації API.

Слабкі сторони: Новий фреймворк з меншою спільнотою та ресурсами, складніша конфігурація, обмежена кількість сторонніх плагінів.



Рисунок В.10 – Слайд «Аналіз результатів»

Публікація результатів

<https://openarchive.nure.ua/entities/publication/1d16e319-fde0-4d2d-ae64-f3ade2a135e1>

УДК 004.42

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



Рисунок В.11 – Слайд «Публікація»

Підсумки

Корисність дослідження:

- Підвищення ефективності розробки
- Вибір технологій
- Оптимізація продуктивності веб-додатків

Розвиток досліджень:

- Розширення тестових сценаріїв
- Вплив нових версій
- Використання машинного навчання



Рисунок В.12 – Слайд «Підсумки»

ДОДАТОК Г

Текст наукової публікації за темою кваліфікаційної роботи

УДК 004.42

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

тел.: +380677136722, 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.

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 [1].

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 [2].

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 запитаннями 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 проєкту. 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 проєкту, такі як його розмір та складність. Важливо враховувати

рівень гнучкості, що потрібен для реалізації унікальних вимог проєкту, 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 проєктом.

Список використаних джерел:

1. Uzayr S. b. Mastering Django: A Beginner's Guide. Taylor & Francis Group, 2022, 322 p.
2. Flask | Read the Docs. Full featured documentation deployment platform - Read the Docs. URL: <https://readthedocs.org/projects/flask/> (дата звернення: 04.03.2024).

ДОДАТОК Д

Експертний висновок результатів перевірки кваліфікаційної роботи на
відповідність оформлення вимогам ДСТУ 3008:

Експертний висновок результатів перевірки кваліфікаційної роботи

студент
(посада)

програмної інженерії
(кафедра)

ІПЗМ-22-5
(група)

Солохін Артем Євгенович

(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
	7.7 Переліки	
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні позначки	
	7.15 Додатки	

зауважень немає

Експерт

(підпис)

Олена ОЛІЙНИК
(прізвище, ініціали)

13.06.2024

ДОДАТОК Е

Відгук керівника роботи

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНИКИ

Факультет КН (або ННЦЗФН або ЦПО)

ВІДГУК

на кваліфікаційну роботу магістра
Солохіна Артема Євгеновича, ПЗМ-22-5,
спеціальність *121- Інженерія програмного
забезпечення*

Освітньо-наукова програма «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Тема кваліфікаційної роботи «Дослідження ефективності використання
фреймворків веб-розробки на продуктивність веб-додатку: порівняльний аналіз Django,
Flask, та FastAPI»

Ця робота розглядає найпопулярніші фреймворки для веб-розробки, особливості використання кожного з них, їх сильні та слабкі сторони

Виявлено що всі важливі аспекти фреймворка легко можна описати як порівняльний аналіз. Проте експеримент проводився на локальному середовищі, але даний недолік не знижує якість виконаної кваліфікаційної роботи.

Під час виконання роботи студентом був проведений ретельний аналіз найпопулярніших аспектів розробки веб-додатків на мові Python. Для цього студент опрацював велику кількість літератури та наявних досліджень, що дозволило обґрунтовано сформулювати чітку різницю між цими фреймворками та виявити їх сильні та слабкі сторони.

Кваліфікаційна робота не містить відомостей, що заборонені до поширення поширення (друку), матеріали, що підлягають патентуванню або ліцензуванню.

Кваліфікаційна робота виконана відповідно до вимогам академічної доброчесності та пройшла перевірку на академічний плагіат, який складає – 9.96%.

Оформлення пояснювальної записки до кваліфікаційної роботи виконано відповідно до вимог.

Попередній захист кваліфікаційної роботи проведено «15» 06 2024 р. Студент гр. ПЗМ-22-5 Солохін А.Є. готовий до самостійної інженерної діяльності і заслуговує присвоєння кваліфікації Магістра з інженерії програмного забезпечення.

Кваліфікаційної роботу можна подати до захисту в ЕК за спеціальністю 121- «Інженерія програмного забезпечення», освітньо-науковою програмою «Інженерія програмного забезпечення».

«15» 06. 2024р.



Керівник кваліфікаційної роботи магістра
проф. каф. П. Смеляков С.В.