

ПОРІВНЯЛЬНИЙ АНАЛІЗ DJANGO I EXPRESS.JS: КРИТЕРІЇ ВИБОРУ ТА СФЕРИ ЗАСТОСУВАННЯ У WEB-РОЗРОБЦІ / COMPARATIVE ANALYSIS OF DJANGO AND EXPRESS.JS: CRITERIA FOR SELECTION AND APPLICATION AREAS IN WEB DEVELOPMENT

Печкурова Е.Я., Печкурова О.М./ Piechkurova E., Pyechkurova O.

Національний університет “Києво-Могилянська Академія” / National University of Kyiv-Mohyla Academy

04655, Київ, вул. Григорія Сковороди, 2, факультет інформатики, кафедра мультимедійних технологій

E-mail: piechkurova@ukma.edu.ua, pyechkurova@ukma.edu.ua

The article is dedicated to the analysis and justification of the selection of frameworks for web application development. It examines the types of web applications and the main criteria for choosing a framework, such as technology stack, tools, security, and flexibility. A comparison of two popular frameworks, Django and Express.js, is conducted, describing their architectural features, advantages, and disadvantages. The article emphasizes the importance of the appropriate choice of framework depending on the project type, performance and security requirements, and developer qualifications. The article may be useful for developers and researchers considering the choice of a technological platform for web development.

У сучасній web-розробці використання фреймворків є невід’ємною складовою ефективною реалізації web-застосунків. Фреймворки забезпечують розробникам набір інструментів і шаблонів для спрощення створення, підтримки та масштабування проєктів. Основна мета їх використання полягає в автоматизації повторюваних задач, покращенні організації коду та підвищенні безпеки застосунків.

Типи web-застосунків та вимоги

Web-застосунки реалізують клієнт-серверну архітектуру, де клієнтська частина взаємодіє через браузер, а сервер обробляє логіку і дані. Різновиди web-застосунків включають SPA (single-page applications), які забезпечують плавну динамічну взаємодію користувача без перезавантажень сторінок, та PWA (progressive web apps), які підтримують офлайн-режими та встановлення на пристрій. Незалежно від типу, усі застосунки мають спільні завдання: маршрутизація, автентифікація, взаємодія з базою даних, захист від атак.

Критерії вибору фреймворку

Основні критерії оцінки фреймворків для веб web-розробки:

- Походження і технологічний стек: мова програмування, відкритість і ліцензія, активність спільноти.
- Інтегровані інструменти: маршрутизація, ORM, шаблонізатори, механізми безпеки.
- Гнучкість: підтримка архітектурних патернів, можливість кастомізації, контроль структури проєкту.
- Безпека: вбудовані засоби захисту від атак (CSRF, XSS, SQL injection), автентифікація та авторизація.
- Практичне застосування: популярність, сфери використання, підтримка в індустрії, документація.

Django — це фулстек web-фреймворк на Python, створений з ідеєю «batteries included», що означає наявність великої кількості вбудованих інструментів для швидкої розробки. Він застосовує архітектуру MVT (Model-View-Template), підтримує власну ORM для роботи з реляційними базами даних, має вбудовані засоби маршрутизації та шаблонізації (Django Template Language). Безпека реалізована на високому рівні зі стандартним захистом від типових web-атак, посиленням механізмом автентифікації, хешуванням паролів і системою дозволів. Django ідеально підходить для застосунків із типовою архітектурою, де потрібна висока швидкість розробки, масштабованість і безпека.

Express.js — мінімалістичний фреймворк для Node.js, написаний на JavaScript. Він забезпечує основу для побудови бекенду, зосереджуючись на маршрутизації і middleware для обробки HTTP-запитів. Express.js не містить вбудованих ORM, системи безпеки чи шаблонізатора, надаючи розробнику максимальну свободу вибору архітектури і сторонніх компонентів. Безпека реалізується через підключення сторонніх бібліотек, наприклад Helmet.js чи Passport.js. Фреймворк підходить для проєктів, що потребують гнучких, кастомізованих рішень та контролю за стеком технологій, добре підтримує асинхронну обробку запитів і завоював широку популярність у індустрії.

Вибір фреймворку має базуватися на характеристиках конкретного проєкту, кваліфікації розробників та вимогах до функціональності й безпеки. Django забезпечує ґрунтовний набір

готових рішень і високу безпеку, підходить для проєктів класичної архітектури. Express.js надає гнучкість і свободу вибору, але вимагає більшого рівня технічної компетенції та уваги до підбору сторонніх інструментів.

1. Choma D., Chwaleba K., Dzieńkowski M. The Efficiency and Reliability of Backend Technologies: Express, Django, and Spring Boot // IAPGOŚ, 4/2023. — P. 73–78. — DOI: 10.35784/iapgos.4279. — Available online: <http://doi.org/10.35784/iapgos.4279> (дата звернення: 1.11.2025).
2. BitByteSoft. Express JS vs Django: A Comparison of Development Approaches and Performance. Available at: <https://bitbytesoft.com/express-js-vs-django-comparison> (accessed November 1, 2025).
3. IEEE Xplore Digital Library. A Comparative Analysis of Node.js and Django for Scalable Web Applications. IEEE. Режим доступу: <https://ieeexplore.ieee.org> (дата звернення: 1.11.2025).
4. Abhishek Jha. Node.js vs Django: A Performance and Scalability Comparison. Режим доступу: https://d197for5662m48.cloudfront.net/documents/publicationstatus/256355/preprint_pdf/f6990ad1813ca8ba2683cb9ee1b01228.pdf (дата звернення: 11.11.2025).
5. Coursera. Node.js vs Django: What's the Difference? 2025, 23 липня. Режим доступу: <https://www.coursera.org/articles/node-js-vs-django> (дата звернення: 1.11.2025)

РОЗРІЗНЕННЯ БЕЗКОШТОВНИХ ЗАСТОСУНКІВ І ПРОБНИХ ВЕРСІЙ ПЛАТНИХ MACOS-ЗАСТОСУНКІВ НА ПУБЛІЧНИХ МАРКЕТПЛЕЙСАХ / DISTINGUISHING FREE APPS AND TRIAL VERSIONS OF PAID MACOS APPS ON PUBLIC APP MARKETPLACES

Літвінчук З.В., Франків О.О, Петелев Є.Р, Кривоблоцький С.І, Стулова Н.С / Litvinchuk Z.V., Frankiv O.O, Peteliev Y.R., Krivoblotsky S.I., Stulova N.S

Національний університет «Києво-Могилянська академія» / National University of Kyiv-Mohyla Academy

04070, м. Київ, вул. Г. Сковороди 2, каф. інформатики, тел. (044) 425 60 64

E-mail: z.litvinchuk@ukma.edu.ua, o.frankiv@ukma.edu.ua, zhenya.peteliev@macpaw.com, krivoblotsky@macpaw.com, nata.stulova@macpaw.com

The lack of transparency in app monetization on public marketplaces misleads users and complicates policy enforcement. The paper presents an algorithm to distinguish truly free macOS apps from those that involve in-app purchases by classifying apps into free, freemium, and paid categories. The research was based on developing a hybrid method that combined technical analysis of Mach-O binaries, embedded frameworks, and receipt file with language-based processing of local bundle resources and external marketplace text content. It yielded a feature set for a supervised machine learning classifier. To evaluate the suggested solution, we used a dataset consisting of 1,219 apps and benchmarked supervised classifiers. The tree-based ensembles performed best: Random Forest and Gradient Boosting achieved ~90% accuracy, precision, recall, and F1-score on an 80/20 train–test split. The findings contribute to classification systems automation, enhancing applicability and transparency in the software distribution ecosystem.

Стрімкий розвиток маркетплейсів програмного забезпечення привів до їх комерціалізації. Межа між справді безкоштовними застосунками та продуктами з платними обмеженнями розмивається. У цих умовах прозорість монетизації та коректність маркування стають критично важливими: різноманіття моделей монетизації (підписки, внутрішні покупки, пробні періоди), непослідовне розкриття платних функцій ускладнюють ручну перевірку й підвищують ризики для користувачів