

розміри. Застосування препроцесорів допомагає підтримувати послідовність стилів, полегшує оновлення у всьому файлі стилів.

Висновок. У результаті проведеного дослідження було спроектовано та реалізовано адаптивний веб-сайт. Використання описаних інструментів дозволяє створити зручний для користувачів дизайн сайту для отримання інформації не зважаючи на розмір екрану.

Список використаних джерел

1. Гломозда Д.К. Проектування, системний аналіз і розробка корпоративних інформаційних систем: навчальний посібник. Київ: НаУКМА, 2015. 95 с
2. Figma. URL: <https://www.figma.com/downloads/> (дата звернення: 26.02.2024).
3. GitLab. URL: <https://about.gitlab.com> (дата звернення: 26.02.2024).
4. Visual Studio Code. URL: <https://code.visualstudio.com> (дата звернення: 26.02.2024).
5. Gulp. URL: <https://gulpjs.com/> (дата звернення: 26.02.2024).

Рецензент к.ф.-м.н., доц. Т.О. Філімонова

УДК 004.42

ТЕХНОЛОГІЧНІ АСПЕКТИ СТВОРЕННЯ API

Вєтохін Д. С., студент, dmitvetokhin911@gmail.com, НУ «Одеська Політехніка»

API (Програмний інтерфейс застосунку) визначається як набір правил та протоколів, що визначають, які функції або сервіси можуть бути використані, а також як вони можуть бути використані. У світі програмного забезпечення API відіграють ключову роль, дозволяючи різним програмам взаємодіяти між собою без необхідності розкриття їхнього внутрішнього коду. [1]

В останні роки, з ростом інтернет-сервісів та мобільних додатків, використання додатків, від вбудованих систем до хмарних сервісів, API забезпечують можливість інтеграції та взаємодії з існуючими системами та послугами.

Забезпечення ефективної інтеграції та взаємодії з різними клієнтами вимагає ретельного врахування конкретних потреб і функціональності проекту. Наприклад, Representational State Transfer (REST) підходить для розроблення простих, зручних у використанні API, які взаємодіють із ресурсами за допомогою стандартних методів HTTP, таких як GET, POST, PUT і DELETE. GraphQL, з іншого боку, забезпечує більш гнучкий підхід, який дає змогу клієнтам

запитувати саме ті дані, які їм потрібні, скорочуючи кількість запитів до сервера і полегшуючи розробку клієнтських додатків. [2]

Таблиця 1 – Порівняння REST та GraphQL

Критерій	REST	GraphQL
Простота	Простий	Складніший
Гнучкість	Негнучкий	Гнучкий
Ефективність	Менш економний	Економний
Продуктивність	Низька	Висока
Зручність розробки	Менш зручний	Зручний
Масштабованість	Легко масштабується	Легко масштабується
Сумісність	Широко сумісний	Менш сумісний

Під час розроблення API також важливо враховувати аспекти безпеки та моніторингу. Важливими аспектами розробки є забезпечення доступу до API тільки авторизованих користувачів, перевірка даних, що вводяться, і захист від таких атак, як SQL-ін'єкції та перехоплення сеансу. Що стосується моніторингу, то важливо мати можливість відстежувати використання API, виявляти проблеми та своєчасно реагувати на них.

Використання таких методологій розробки програмного забезпечення, як Agile і DevOps, дає змогу прискорити випуск нових версій API і забезпечити їхню високу якість. Швидкі ітерації розробки, тестування та впровадження дають змогу оперативно реагувати на зміну вимог і виправляти помилки. [3]

Керування життєвим циклом API вимагає системного підходу і використання спеціалізованих інструментів. Кожна фаза циклу, від початкового проєктування до виведення з експлуатації, має свої вимоги та завдання, і важливо мати корисний інструмент для відстеження використання API й аналізу його ефективності та здатності виконувати бізнес-завдання. Тільки так можна забезпечити успішну експлуатацію та підтримку протягом усього життєвого циклу API.

Важливо також розглянути можливість автоматизації тестування та розгортання за допомогою інструментів безперервної інтеграції/безперервного розгортання (CI/CD) для постійного вдосконалення та оптимізації процесу розробки API. Ці підходи автоматизують процес випуску нових версій API і дозволяють швидко вносити зміни та виправляти помилки. Вони також допомагають забезпечити стабільність і надійність API, оскільки проблеми, що виникають під час розробки, можуть бути швидко виявлені та виправлені. [4]

Крім того, розробка API передбачає аналіз використання та збір показників ефективності. Це дає розуміння того, як клієнти взаємодіють з ресурсами API і як це впливає на загальну продуктивність системи. Ця інформація може бути використана для подальшої оптимізації та вдосконалення API, що призведе до кращого користувацького досвіду та оптимального використання серверних ресурсів. Цей цикл безперервного вдосконалення допомагає гарантувати, що API

залишається актуальним і конкурентоспроможним у світі програмного забезпечення, що швидко змінюється.

Список використаних джерел

1. Subramanian S, Inozemtseva L, Holmes R. Live API documentation. Proceedings of the 36th International Conference on Software Engineering. ICSE 2014. May 2014; pages 643–652. <https://doi.org/10.1145/2568225.2568313>
2. Brito G, Valente MT. REST vs GraphQL: A Controlled Experiment. In: 2020 IEEE International Conference on Software Architecture (ICSA). IEEE; 2020. p. 232-243. <https://ieeexplore.ieee.org/abstract/document/9101226>
3. Al-Saqqa S, Sawalha S, Abdelnabi H. Agile Software Development: Methodologies and Trends. International Journal of Interactive Mobile Technologies [Internet]. 2020 Jul 10;14(11):13269. [cited 2024-02-27]. <https://doi.org/10.3991/ijim.v14i11.13269>
4. Mangla, M. (2023). Securing CI/CD Pipeline: Automating the detection of misconfigurations and integrating security tools [Master's thesis, National College of Ireland]. <https://norma.ncirl.ie/6529/>

Рецензент к.т.н., доцент кафедри інформаційних технологій НУ «Одеська політехніка» М. Д. Рудніченко

УДК 004.4

РОЗРОБКА ІГРОВОГО ПРОГРАМНОГО ПРОДУКТУ НА БАЗІ UNITY

Войніков Н.А., студент, voinikov.8895593@stud.op.edu.ua, кафедра ІТ НУ«ОП»

Розробка ігрового програмного продукту на базі платформи Unity є високотехнологічним процесом, що вимагає від розробників не просто поверхневих знань у сфері програмування та дизайну, а глибокого розуміння комплексу технічних, артистичних і геймдизайнерських нюансів. Unity, як визнаний лідер у сфері розробки ігрових рушіїв, пропонує безпрецедентний набір інструментів і можливостей, що дозволяють реалізовувати ігрові проекти найрізноманітнішої складності та спрямованості, від незалежних інді-проектів до великих багатокористувацьких онлайн-ігор. Платформа Unity забезпечує підтримку майже всіх сучасних ігрових платформ, у тому числі, але не обмежуючись, Windows, macOS, Linux, Android, iOS, WebGL, PlayStation, Xbox, що відкриває перед розробниками можливість звернутися до глобальної аудиторії без необхідності адаптації коду під кожну конкретну платформу.