

Іващенко Олександр-Данііл Олександрович, здобувач вищої освіти

КПІ ім. Ігоря Сікорського, Україна

Науковий керівник: Уляницька Ксенія Олександрівна, кандидат технічних наук,

доцент кафедри інформаційних систем та технологій

КПІ ім. Ігоря Сікорського, Україна

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ УПРАВЛІННЯ ТА РОЗПОВСЮДЖЕННЯ НАВЧАЛЬНИХ КУРСІВ

Анотація. Дана робота присвячена опису реалізації програмного забезпечення для ефективного управління та поширення навчальних матеріалів та курсів. Зокрема, у роботі розглядаються аспекти автоматизації процесів розробки, моніторингу та адміністрування курсів, з метою підвищення якості та доступності освіти. В статті також було розглянуто та запропоновано технології та можливу архітектуру для створення подібного застосунку. Результати цього дослідження можуть сприяти створенню нових та оптимізації уже існуючих систем управління навчальним процесом.

КЛЮЧОВІ СЛОВА: програмне забезпечення, управління, курси, архітектура.

Abstract. This paper describes the implementation of software for the effective management and distribution of educational materials and courses. In particular, the paper discusses aspects of automating the processes of developing, monitoring and administering courses in order to improve the quality and accessibility of education. The paper also considers and proposes technologies and a possible architecture for creating such an application. The results of this study can contribute to the creation of new and optimization of existing learning management systems.

KEYWORDS: software, management, courses, architecture.

Вступ. З розвитком інформаційних технологій та їх інтеграції у навчальний процес, курси стали все більш актуальним інструментом для організації навчання. Підхід з використанням курсів дозволяє автоматизувати та оптимізувати процеси взаємодії з навчальними матеріалами та оцінювання учнів. Розділення та систематизація інформації у зручні модулі та окремі одиниці допомагає учням краще обирати потрібну їм інформацію й орієнтуватись загалом. Використання гнучких дедлайнів та автоматизованих систем оцінювання сприяє зручності використання курсів і дає учням можливість адаптуватись до темпу навчання, що в комбінації з використанням інтерактивних задач дозволяє значно підвищити залученість учнів у навчальний процес та якість засвоєння інформації.

Хоч курси і є ефективним інструментом, для їх використання потрібна відповідна інфраструктура, яка повинна забезпечувати деякі основні функції:

- платформа для доступу до курсів, на якій учні зможуть проходити ці курси та отримувати доступ до їх функціоналу і матеріалів;
- система розповсюдження курсів, завдяки якій учні зможуть отримати інформацію про конкретні курси та платформу;
- сервіс сертифікації, який забезпечить верифікацію виконання певних курсів та їх результатів чи прогресу;
- зручна точка доступу до усієї системи та інструкції по її використанню;

- модулі і сервіси, які відповідають за технічну підтримку, оновлення, супровід та розробку курсів, в тому числі і збір методичних матеріалів;

- канал зв'язку з учнями, який дозволить отримати відгуки про роботу системи і забезпечить можливість їх покращення.

Оскільки система буде обслуговувати велику кількість користувачів, то виникає потреба в забезпеченні масштабованості підсистем та пропускної здатності каналів зв'язку.

Архітектура систем. Загальна архітектура застосунку може складатись з наступних головних компонентів:

1) веб-сайт в якості точки доступу, де буде розміщена загальна інформація та посилання на інші частини системи;

2) веб-сервер, на якому розміщені усі головні сервіси програми;

3) платформа у вигляді веб-сайту чи настільного застосунку, яка надає доступ до курсів;

4) сервіси та програмні компоненти для зв'язку модулів системи між собою.

Для зв'язку елементів системи можна використати типову систему з запитом, але при кратному збільшенні навантаження та користувачів може знадобитись більш складна система з використанням модулів балансування навантаження та переадресацією запитів.

Точка доступу. Тут користувач отримує доступ до усіх потрібним йому ресурсам та інформації, яка йому може знадобитись для використання системи. Може бути виконана у вигляді звичайного веб-сайту з зручним та зрозумілим інтерфейсом.

Найпростіше цей модуль буде реалізувати з використанням наступних технологій: JavaScript, CSS, HTML для створення інтерфейсу, Node.js і його бібліотеки для реалізації функціоналу та Axios для запитів до інших модулів системи.

Головний сервер. Від вибору архітектури цього модуля дуже залежить більша частина того, як буде виглядати система загалом. Якщо система розробляється для відносно малої кількості користувачів, то доцільно використати монолітну архітектуру, оскільки це збереже дуже багато витрат на розробку

та налаштування зв'язків у системі. Для великої кількості користувачів рекомендується використати мікросервісну архітектуру, оскільки вона дозволяє легко масштабувати окремі частини системи та додавати новий функціонал.

Платформа. Для створення платформи є два потенційних підходи: веб-сайт з використанням тих самих технологій що і попередні модулі, або окремий настільний застосунок. Перший варіант має перевагу у простоті реалізації та інтеграції в уже існуючі частини системи, а другий в незалежності від Інтернету та обмеженнях браузера користувача. Найважливіше в цьому модулі це: надійність, зручність інтерфейсу та підтримка основних функцій, таких як відображення мультимедіа та інтерактивного контенту. Якщо реалізувати цей модуль як настільний застосунок, то слід приділити увагу зв'язку з іншими підсистемами та модулем оновлення, оскільки платформу потрібно регулярно оновлювати, щоб та відповідала актуальному функціоналу курсів.

Бази даних. При мікросервісному підході, найкращим рішенням буде надати окрему базу даних кожному вагомому сервісу, оскільки це спростить формування таблиць, запитів та загальної схеми сховища даних. Тут можна використати реляційну SQL базу даних, наприклад PostgreSQL та об'єктно-реляційну проекцію, таку як Prisma чи Drizzle. Це значно полегшить розробку. Головною вимогою до баз даних у цій системі є надійність та розширюваність, оскільки в систему одночасно може потрапити багато нової інформації.

Зв'язок та запити. Оскільки система розділена на модулі, важливо забезпечити ефективний зв'язок між ними. Для зв'язку в подібних системах зазвичай використовується API, як для зовнішніх запитів, так і для запитів між модулями, але

тут має місце велика кількість користувачів, тому знадобиться окремий сервіс, який буде обробляти та верифікувати запити від користувачів. Для такого сервісу можна використати Node.js, а для запитів між модулями створити прості інтерфейси з використанням Rest API.

Висновки. Представлене в даній роботі програмне забезпечення дозволяє ефективно використовувати курси у навчальному процесі та підтримувати їх актуальність. Запропонована архітектура враховує навантаження, яке виникає в подібних системах та функціонал, який потрібен для зручного користування, підтримки та розробки нових курсів. Описано окремі модулі та компоненти системи, а також способи реалізувати зв'язок між ними. Використовуючи усі описані вище підходи, можна покращити та оптимізувати уже існуючі системи управління процесом навчання або розробити нові, які будуть відповідати сучасним стандартам та вимогам.

Список інформаційних джерел

1. Al-Ajlan, A. and H. Zedan, 2007. E-learning (MOODLE) based on service-oriented architecture. Proceeding of the EADTU's 20th Anniversary Conference, Nov. 8-9, Lisbon-Portugal, pp: 60-72.
2. Nilles, J. "Some Historical Thoughts on the ee-Learning Renaissance". Innovate. Innovate Online. URL: https://www.researchgate.net/publication/239801662_Some_Historical_Thoughts_on_the_ee-Learning_Renaissance (дата звернення: 29.11.2023)
3. Omair Ameerbakhsh. (2018). Towards the Use of Interactive Simulation for Effective e-Learning in University Classroom Environment, P 8-12.
4. Pappano, Laura. "The Year of the MOOC". The New York Times. ISSN 0362-4331. URL: <https://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html> (дата звернення: 29.11.2023)
5. Вишнівський В.В., Гніденко М.П., Гайдур Г.І., Ільїн О.О. Організація дистанційного навчання. Створення електронних навчальних курсів та електронних тестів. Київ : ДУТ, 2014. 141 с.