

**Головіна К.П.**

Національний університет «Запорізька політехніка»

**Касьян К.М.**

Національний університет «Запорізька політехніка»

**Касьян М.М.**

Національний університет «Запорізька політехніка»

## РОЗРОБКА ВЕБДОДАТКУ ДЛЯ ВИВЧЕННЯ ІТ-КУРСІВ

У наші часи більшість галузей цифровізується, зокрема й навчальна галузь. З розвитком пандемії у світі дистанційне навчання набирає популярність та актуальність завдяки своїй безпечності й зручності. З кожним роком швидкість розвитку технологій збільшується, постає потреба в спеціалістах цієї галузі. Вебдодаток, який є результатом роботи, пропонує онлайн-платформу для вивчення ІТ-курсів з функціями самостійного контролювання студентом процесу навчання, проходження тестів після кожної завершеної теми. Пропонуються проектування й реалізація вебдодатку для вивчення ІТ-курсів, що допоможе суттєво зекономити час учителя та студента, допомогти студенту самостійно обирати курси, які йому необхідні. Також студент матиме можливість відстежувати свій прогрес у навчанні, вивчати печатні матеріали з електронної бібліотеки вебдодатку, прослуховувати лекції з відеозаписів уроків курсу. Водночас можна проходити декілька курсів із запам'ятовуванням останнього уроку, на якому зупинився студент. У процесі роботи виділено основні функції та вимоги до вебдодатку, проведено аналіз і вибір актуальних сучасних технологій для серверної та клієнтської частин вебдодатку. Спроектовано й розроблено базу даних вебдодатку. У результаті роботи спроектовано й розроблено вебдодаток, який відповідає поставленим вимогам і виконує всі перелічені функції. Також проведено навантажувальне та стрес-тестування вебдодатку, порівняльний аналіз результатів навантажувальних і стрес-тестів. Для проведення тестування використовувалася утиліта *jMeter*. У результаті отримано статистику за зведеним метрикам продуктивності вебдодатку.

**Ключові слова:** вебдодаток, база даних, розробка вебдодатку, навантажувальне тестування, стрес тестування, *ReactJS*, *Laravel*, *jMeter*.

**Постановка проблеми.** Інформаційні технології поступово ввійшли майже в усі галузі нашого життя. Освітня галузь не виняток. Цифровізація освіти значно спрощує більшість процесів, суттєво економить час учнів і вчителів.

Онлайн-платформа для вивчення ІТ-курсів дасть змогу проходити всі заняття дистанційно, самостійно оцінювати рівень знань за кожною темою, а потім легко застосовувати всі отримані знання на практиці.

Завдяки онлайн-сервісу з вивчення курсів сфера ІТ стає доступною для кожного. Студент має великий вибір серед різноманіття інформаційних джерел у бажаній формі (відео або печатні матеріали у формі статей). Під час проходження курсу студент зможе відстежувати свій прогрес і контролювати власні успіхи.

**Аналіз останніх досліджень і публікацій.** Сьогодні студент має величезний вибір платформ для вивчення різноманітних тем, що стосуються ІТ-технологій тощо. Нижче наведено перелік платформ, які є популярними серед студентів:

- Coursehunter – ресурс відеоуроків із програмування для front і back розробників;
- HTMLAcademy – курси для вивчення роботи верстальника;
- Netology – курси для підготовки ІТ-спеціалістів за різними спеціальностями.

Ці ресурси мають велику кількість курсів з відео, вибір серед платних і безкоштовних курсів, досить багато гарних відгуків. Але разом із тим немає можливості відстежувати прогрес у навчанні, видачі сертифікатів про закінчення курсу (у деяких ресурсах із наведеного списку) і тестування після проходження уроків. Контроль знань є вагомим аргументом на користь запам'ятовування матеріалу, адже після проходження уроку треба виділити найважливіші моменти та проконтролювати уважність студента.

**Постановка завдання.** Мета статті – забезпечити доступність вивчення ІТ-курсів дистанційно, дати змогу студенту самостійно корегувати свій план навчання.

Для досягнення поставленої мети необхідно виконати такі пункти:

- провести аналіз і зробити вибір на користь найбільш ефективних технологій при роботі з великим обсягом даних;
- визначити основні вимоги до сервісу;
- спроектувати сервіс згідно з визначеними вимогами;
- розробити реалізацію сервісу з використанням обраних на першому етапі технологій і з дотримання вимог, визначених на другому етапі роботи;
- проаналізувати ефективність розробленого сервісу.

**Виклад основного матеріалу дослідження.**

Аналіз сучасних технологій для розробки сервісу. При виборі технологій для розробки необхідно врахувати такі фактори:

- рівень надійності рішення;
- рівень безпеки рішення;
- можливість обробки великих масивів даних, не впливаючи на швидкість роботи сервісу.

Для роботи з клієнтською частиною вибір проводився між двома фреймворками й бібліотекою – Angular, Vue.js і React.js.

Відповідно до статистики із сайту <https://www.npmtrends.com/>, кількість завантажень за останні 6 місяців зображена на рисунку 1.

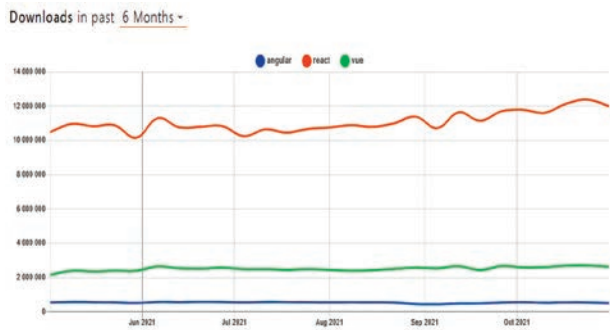


Рис. 1. Обсяги завантажень пакетів angular, react і vue

AngularJS – фреймворк із відкритим вихідним кодом, використовується з мовою програмування – Typescript [1]. Основою розробки на Angular є класово-модульний підхід і використання готових убудованих рішень для масштабування, тестування й оптимізації додатків.

React – це бібліотека JavaScript. Однією з переваг React є простий дизайн, використання JSX для шаблонів і досить докладної документації [2].

Vue.js – це JavaScript-фреймворк, заснований у 2013 році, який ідеально підходить для створення високоадаптованих інтерфейсів користувача і складних односторінкових додатків [3].

Серед двох фреймворків і бібліотеки обрано саме бібліотеку React через її універсальність, простоту та швидкий процес навчання.

React дає змогу створювати вебдодатки, що змінюють відображення без перезавантаження сторінки. Саме завдяки цьому вебдодатки швидко реагують на дії користувача, наприклад, на заповнення форм, застосування фільтрів, додавання товарів у кошик тощо.

Для роботи із серверною частиною обраний Laravel через низку таких переваг:

- MVC [6];
- ORM;
- шаблонізатор – уніфікація відображення програми;
- аутентифікація та інтеграція із сервісами;
- продуктивність;
- система міграцій БД.

Вебдодаток, який є результатом роботи, корисний для вивчення мов програмування або нових ІТ-технологій. Студент має можливість самостійно обрати необхідні для нього курси, проходити урок за уроком, відстежуючи прогрес та історію занять. Курси представлені переважно у відеоматеріалах, які має можливість додати будь-який бажаючий. Після перевірки ІТ-спеціалістом курс буде або відхилено, або відправлено на доробку (якщо не вистачає якихось пунктів для повного розкриття зазначеної теми). Також курси доповнені прикладами коду або архівом із проектом. Крім того, сервіс має бібліотеку книг в електронному форматі, частина яких є платним ресурсом. Вебдодаток має загальну структуру, яка є різною для зареєстрованих користувачів і гостей сайту. Структура вебдодатку для незареєстрованих користувачів:

- головна сторінка (форма пошуку, список курсів, популярні статті);
- сторінка допомоги спеціаліста (з питань користування вебдодатком);
- сторінка реєстрації;
- сторінка входу у вебдодаток(форма для входу у вебдодаток);

Структура вебдодатку зареєстрованих користувачів:

- головна сторінка (форма пошуку, список курсів);
- сторінка профілю користувача (список активних і завершених курсів, редагування особистих даних і налаштувань профілю);

Цей вебдодаток має виконувати такі функції:

- авторизація та реєстрація;
- перегляд навчального матеріалу;
- перевірка знань після проходження навчального матеріалу;

- відстеження прогресу в навчанні;
- створення навчального курсу.

Щоб не випустити з уваги потреби окремих користувачів, виділимо їх у групи. Наприклад, за частотою роботи з ПЗ, використовуваними функціями, рівнем привілеїв і навичками роботи [8].

Таблиця 1

**Характеристики користувачів**

Роль користувача	Характеристики користувача
Відвідувач	Особа, яка може переглядати доступні курси, ознайомитися з темами, які будуть вивчені в процесі. Для проходження курсів необхідно зареєструватися.
Користувач	Особа, яка зареєстрована на сайті, може вибирати й проходити курси, додавати в обране.
Адміністратор	Особа, яка має право редагувати інформацію вебдодатку, допомагати користувачам у режимі чату.
ІТ-спеціаліст	Особа, яка контролює контент курсів і їх коректність. Може видалити, схвалити або відправити на доопрацювання автору.

Реалізацію сервісу можна розділити на три основні частини:

- проектування та розробка бази даних;
- реалізація серверної частини;
- реалізація клієнтської частини.

Для бази даних вебдодатку для вивчення ІТ-курсів знадобиться 7 таблиць:

- користувачів: ім'я, прізвище, номер телефону, місто проживання, роль користувача (звичайний користувач, адміністратор або ІТ-спеціаліст);

- courses – таблиця, зберігає інформацію про курс: назву, короткий опис, прев'ю (картинка), складність курсу (легкий, середній, складний), кількість уроків, унікальний ідентифікатор категорії;

- lessons – таблиця, зберігає інформацію про уроки курсу. Ця таблиця має такі поля: назву уроку, короткий опис, дату створення, унікальний ідентифікатор курсу, якого стосується цей урок, порядковий номер уроку в курсі;

- media – таблиця, зберігає шлях до файлів з медіа, які стосуються певного уроку, та унікальний ідентифікатор уроку;

- progress – таблиця, що пов'язує користувачів і курси, які вони проходять на цей момент. Має такі поля: унікальний ідентифікатор користувача, унікальний ідентифікатор курсу, урок, на якому зараз знаходиться користувач (унікальний ідентифікатор уроку), статус закінчення курсу (прапор, який визначає, користувач у процесі знаходиться чи вже завершив курс);

- categories – таблиця, зберігає дані про категорії, на які можна поділити курси;

- comments – таблиця, зберігає дані про коментарі до кожного курсу, складається з таких полів: унікальний ідентифікатор курсу, унікальний ідентифікатор користувача, що є автором коментаря, заголовок коментаря, кількість балів (за шкалою від 0 до 5 – для визначення рейтингу курсів у своїй категорії) і текст коментаря. Коментарі може залишати лише зареєстрований користувач і тільки по завершенню курсу (перегляду всіх уроків, із яких складається курс).

Серверну частину вебдодатку розроблено мовою PHP на основі фреймворку Laravel з використанням вебсерверу nginx і sql-сервера MySQL.

У вебдодатку для реалізації функції авторизації/реєстрації/відновлення паролю використано пакет Laravel Breeze, який після встановлення та міграцій додасть необхідні таблиці до бази даних, створить нові роути для переходу по цих сторінках і шаблони цих сторінок за замовчуванням.

Після аналізу спроектованої бази даних додано міграції для створення таблиць у базі даних. Для комфортної роботи з ресурсами – курси, уроки, медіафайли, коментарі та категорії, створено моделі цих сутностей і контролери для додавання функціоналу CRUD – для читання, створення, видалення й редагування кожної сутності.

Вебдодаток побудовано за шаблоном, де всю логіку роботи з моделями винесено в репозиторій. Репозиторій дає змогу використовувати об'єкти, не знаючи, як ці об'єкти зберігаються. Це абстракція шару даних. Це дає змогу розділити бізнес-логіку та логіку роботи з даними. Бізнес-логіка накладається на репозиторій в отриманні правильних даних.

Клієнтська частина розроблена за допомогою бібліотеки React, для якої характерна наявність головного файлу App.js, у який далі додаються інші компоненти ієрархічно. Кожний компонент має функцію render(), що є головною в компоненті й дає змогу відмалювати компонент на сторінці. Вебдодаток використовує функціональні компоненти та функції хуки для відстежування стану елементів сторінок.

Вебдодаток має такі сторінки: одразу при вході на сайт користувач може побачити всі доступні курси, відфільтрувати їх або відсортувати за бажанням чи потребою. У списку видно лише назву та складність курсу. Для перегляду детальної інформації про курс необхідно перейти за посиланням на сторінку курсу, де наявний короткий опис, перелік уроків, які будуть на курсі.

Вебдодаток має платні й безкоштовні курси, усе залежить від користувачів, які при додаванні свого курсу обирають цю опцію. Тож безкоштовні курси доступні одразу після реєстрації, а платні – після покупки курсу.

Структура клієнтської частини вебдодатку збудована так: стартовий компонент, де починається відмалювання всіх інших компонентів (зазвичай це блок з ідентифікатором root), далі сайт ділиться на навігаційну панель, бокову панель та основний контент. Навігаційна панель зберігає компонент роутера, який збудовано за допомогою router-dom, що дає змогу переходити між сторінками без перезавантаження сторінки.

Для перевірки продуктивності й часу відгуку вебдодатку необхідно провести його навантажувальне тестування. Тестування проводиться за допомогою jMeter. Також цей вид тестування допоможе знайти вузькі місця в сценарії користувача. Метою є збір та аналіз отриманих метрик навантажувального тестування [9].

Виходячи з аналізу предметної галузі, визначили оптимальне число потоків, що водночас працюють, – 50 користувачів. Ця кількість потоків відповідає профілю навантажувального тестування. Також планується провести стрес-тестування. Профіль стрес-тестування становить 80 потоків, що водночас працюють.

Складники навантажувального тестування:

- загальна кількість користувачів у навантажувальному тестуванні = 50 користувачів;
- крок навантаження = 3 користувача кожні 10 секунд;

Сценарій навантажувального тестування:

- авторизація;
- профіль користувача;
- список курсів;
- створення курсу.

Таблиця 2

**Апаратне забезпечення тестового стенду**

Server	vCPUs	Fractional vCPUs	Memory	Maximum egress bandwidth (Gbps)
Ee2	22	00,5	22 GB	1

Після завершення навантажувального тестування сформовано загальний звіт зі зведеними метриками продуктивності

Загальна статистика відображає метрики “Aggregate report” jMeter:

- загальна кількість отриманих відповідей під час тестування – 250;

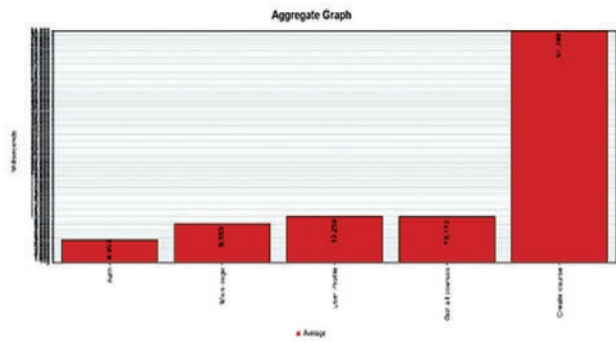


Рис. 2. Aggregate graph

- помилки отримані протягом тесту – 0;
- середній час отримання відповіді – 9703 мс ≈ 10 с;
- мінімальний час отримання відповіді – 1383 мс; тут варто звернути увагу на ендпоінт “Create course”, мінімальний час отримання відповіді якого становить 11068 мс. Це помітна відмінність від інших ендпоінтів;

- максимальний час отримання відповіді – 91162 мс ≈ 91 с; варто звернути увагу на отримання відповіді від сервера при багатопоточних запитах (50 користувачів). Система помітно деградує;

- загальні показники максимального часу відповіді – дуже низькі;
- загальний час 95-го перцентілю – 72298 мс ≈ 72 с;
- запити/секунду – 1,9; навіть без наявності явних вимог до швидкості роботи ендпоінтів можна відзначити, що показник швидкості низький.

Після проведення тестування визначено:

- виходячи з результатів, можна відзначити, що час отримання відповіді в 95% випадках більше ніж 1500 мс, що є дуже низьким показником;
- виявлено вузькі місця системи – “Create Course”;
- помилок під час виконання тесту не сталося;
- система готова витримувати очікуване навантаження на серверну частину програми.

**Висновки.** У підсумку роботи проведено аналіз доступних технологій та обрано найефективнішу технологію для вирішення задачі. Спроековано й визначено основні вимоги для роботи вебдодатку. Виходячи з результатів дослідницької частини, можемо констатувати, що вебдодаток здатен обслуговувати очікувану кількість користувачів в один момент часу. Також, виходячи з результатів стрес-тестування, треба розробити покращення backend-частини для поліпшеної продуктивності вебдодатку в разі стрес-навантаження. Вебдодаток повністю відповідає вимогам і забезпечує доступність вивчення ІТ-курсів для користувачів.

**Список літератури:**

1. Angular. URL: <https://angular.io/> (дата звернення: 10.10.2021).
2. Документація ReactJS. URL: <https://en.reactjs.org/> (дата звернення: 10.10.2021).
3. Використання Vue.js. URL: <https://vuejs.org/> (дата звернення: 10.10.2021).
4. Стаффер М. Laravel. Повне керівництво. 2-ге видання. Санкт-Петербург : Пітер, 2021. 512 с.
5. Symfony. URL: <https://symfony.com/> (дата звернення: 10.10.2021).
6. Сесіл Мартін Р. Чистий код. Створення і рефакторинг за допомогою Agile. Київ : Фабула, 2019. 448 с.
7. Налаштування nginx сервера. URL: <https://nginx.org/ru/> (дата звернення: 10.10.2021).
8. Вігерс К., Бітті Д. Розробка вимог до програмного забезпечення. 3-тє видання. Москва : Російська редакція, 2017. 736 с.
9. Майєрс Г., Баджетт Т., Сандлер К. Мистецтво тестування програм, 3-тє видання. Київ : Діалектика, 2018. 272 с.

**Holovina K.P., Kasian K.M., Kasian M.M. WEB-BASED APPLICATION FOR LEARNING IT-COURSES**

*Nowadays, more and more industries are being digitized, including the education sector. With the development of the pandemic in the world, distance learning is gaining popularity and relevance due to its safety and convenience. Every year the speed of technology development increases, and there is a need for specialists in this field. Web application, which is the result of the work, offers an online platform for learning IT courses with the functions of self-monitoring by the student of learning process, passing tests after each completed topic.*

*It is proposed to design and implement a web application for learning IT-courses, which will significantly save time for teachers and students, help students choose the courses they need. The student will also have the opportunity to track their progress in learning, study printed materials from the electronic library of the web application, listen to lectures on video recordings of course lessons. You can take several courses at a time to memorize the last lesson the student attended.*

*In the course of work the basic functions and requirements to the web application are allocated, the analysis and a choice of actual modern technologies for server and client parts of the web application is carried out. A web application database has been designed and developed. As a result, a web application was designed and developed that meets the requirements and performs all these functions. There was also a load and stress testing of the web application, a comparative analysis of the results of load and stress tests. The jMeter utility was used for testing. As a result, we obtained statistics on the consolidated performance metrics of the web application.*

**Key words:** *web application, database, web application development, stress testing, stress testing, ReactJS, Laravel, jMeter.*