

НАВЧАЛЬНА ПЛАТФОРМА ДЛЯ ВИВЧЕННЯ FRONT-END "4MIN-IT"

Вінницький національний технічний університет

Анотація

У даній роботі досліджено структуру та технології розробки навчальної платформи для вивчення Front-end технологій та подальшого працевлаштування. Розглянуто принципи роботи багатосторінкових WEB-застосунків, стек технологій Full-stack з використанням JavaScript.

Ключові слова: навчальна платформа, Front-end, WEB-застосунок, JavaScript, сайт.

Abstract

This paper investigates the structure and technologies of developing a learning platform for studying Front-end technologies and further employment. The principles of operation of multi-page WEB-applications, Full-stack technology stack using JavaScript are considered.

Keywords: learning platform, Front-end, WEB-application, JavaScript, website.

Вступ

Актуальність теми зумовлена розвиток ринку праці у сфері ІТ галузі. Розроблений продукт дозволить користувач швидко та ефективно вивчити технології пов'язанні з Front-end: HTML, CSS, SCSS, GIT, JavaScript, Ajax, React, Axios, Context, Rest API. Вагомим аргументом є те, що розробка веб-застосунків та подальший розвиток у мобільну розробку є дуже перспективним на ринку праці.

Метою розробки є розробка платформи з вивчення Веб-технологій та подальше працевлаштування з використанням технологій Front-end та Back-end на базі JavaScript.

Огляд технологій для створення навчальних веб-платформ

Аналіз ринку праці, вимог до кандидатів на роботу за фахом Front-end, актуальних оновлень різних сумісних бібліотек, вимоги користувачів показав, що потрібно забезпечити платформу наступним функціоналом: приємний та легкий дизайн, який не заважає очам; авторизація та реєстрація юзера; прив'язка акаунта до пошти; розподілення ролей на студента та викладача; перегляд структурованого та зрозумілого матеріалу; домашнє завдання; надсилання домашнього завдання; зміна пароля, імені та прізвища; спілкування між викладачем та студентом; зі сторони викладача менторити студента; добавляти людей в групу; перегляд викладачем учасників групи; онлайн зустрічі [1].

Багатосторінковий сайт - це колекція веб-сторінок, які зазвичай зберігаються на одному веб-сервері та доступні через Інтернет. Кожна сторінка може бути написана з використанням HTML, CSS і JavaScript, а також може містити динамічний контент, який генерується на сервері або клієнтському боці. В структурі багатосторінкового сайту зазвичай є декілька основних файлів, таких як HTML-файли сторінок, які містять код для відображення контенту [2]; CSS-файли, які відповідають за стилізацію та оформлення вмісту; JavaScript-файли, які забезпечують інтерактивність та функціональність на сторінках.

Крім того, багатосторінкові сайти можуть використовувати серверні мови програмування, такі як PHP, Python, або Node.js, для обробки запитів користувачів та генерації вмісту динамічно. Це дозволяє створювати сайти з різноманітними можливостями, такими як автентифікація користувачів, обробка форм, взаємодія з базами даних та інше. У цілому, багатосторінковий сайт складається зі зв'язку різних технологій та файлів, що працюють разом для створення інтерактивного та інформативного веб-досвіду для користувачів.

Багатосторінковий додаток (MPA) складається з декількох сторінок із статичною інформацією (текстом, зображеннями тощо) та посиланнями на інші сторінки з тим самим вмістом. Під час переходу на іншу сторінку браузер робить новий запит до сервера і знову завантажує всі ресурси, навіть ті компоненти, які повторюються на всіх сторінках (наприклад, заголовок, нижній колонтитул) [3]. Переваги багатосторінкових застосунків: високий рівень SEO, можливість широкої взаємодії з

користувачем та легкість масштабування. Недоліки: розробка є дорожчою, розробка вимагає більшого стеку технологій, а також важча оптимізація.

Для розробки платформи використовувався стек технологій схожий до MERN, який потрібний для створення Full-stack проектів. MERN складається з стеку MongoDB, Express.js, React та Node.js, тільки замість MongoDB я використав базу даних PostgreSQL, але не принцип роботи MERN не впливає. Дані технології надають змогу створити проект як зі сторони клієнта, так і зі сторони сервера. На зображенні нище наведений приклад роботи застосунка. React – це JavaScript бібліотека, яка використовується для створення інтерфейсів користувача, зокрема веб-додатків. Він розроблений компанією Facebook і зарекомендував себе як потужний інструмент для створення масштабованих та ефективних користувацьких інтерфейсів. React базується на концепції компонентів, що дозволяє розбивати інтерфейс на невеликі незалежні частини. Це полегшує розробку, тестування та підтримку додатків. React використовує віртуальний DOM для оптимізації оновлення інтерфейсу. Це дозволяє зменшити кількість маніпуляцій з реальним DOM та підвищити продуктивність додатків. React автоматично оновлює інтерфейс при зміні даних, що забезпечує швидкий та плавний досвід для користувачів.

Однак для новачків може бути важким зрозуміти концепції React, такі як JSX, віртуальний DOM та стейти. Хоча React має широку спільноту, нові версії та підходи можуть вимагати перепрограмування існуючого коду, що може бути трудомістким. Node.js – це виконавче середовище JavaScript, яке дозволяє запускати код JavaScript на серверному боці. Node.js працює на одному потоці з подіями (event-driven), що дозволяє обробляти багато запитів одночасно без блокування. Це забезпечує високу швидкість для серверних додатків. Використання JavaScript як мови програмування як на клієнтському, так і на серверному боці дозволяє розробникам використовувати спільний код, що спрощує розробку та підтримку проектів. Node.js має велику та активну спільноту, що пропонує багато корисних пакетів через npm (Node Package Manager), які допомагають при розробці різноманітних застосунків. Node.js підтримує асинхронний ввід/вивід, що дозволяє ефективно взаємодіяти з файловою системою, мережею та іншими введеннями/виведеннями, не блокуючи потік виконання. У випадках, коли потрібно виконувати великі обчислення, які вимагають використання багатьох ядер процесора, Node.js може бути менш ефективним. Іншою проблемою є так званий зліт Callback Hell може виникнути при роботі з великою кількістю асинхронних функцій, що може ускладнити розуміння та підтримку коду [4]. Також існують проблеми між різними версіями пакетів.

Express.js – це легковаговий та гнучкий веб-фреймворк для Node.js, призначений для розробки веб-додатків та веб-серверів. Він надає простий інтерфейс для обробки запитів HTTP, маршрутизації, відправки та отримання даних з сервера, а також інтеграції з різними шаблонізаторами та базами даних. PostgreSQL використовує реляційну модель даних, що дозволяє ефективно зберігати та керувати структурованими даними. PostgreSQL відома своєю стабільністю та надійністю, а також можливістю відновлення після аварій. Вона має багато додаткових функцій, таких як підтримка JSON, XML, географічних даних, повнотекстовий пошук тощо. В порівнянні з деякими іншими базами даних, PostgreSQL може вимагати більше ресурсів для роботи. Налаштування PostgreSQL може бути складним завданням, особливо для початківців.

Висновки

Описані у публікації технології використано у курсовій роботі. Для розробки клієнтської та серверної сторони застосунка була використана мова програмування JavaScript. Був вибраний правильний підхід Full-stack розробки, що дозволило зробити проект швидким та легким для модифікацій. Для сторони Front-end було використано фреймворк React та багато його бібліотек, щоб зробити застосунок зрозумілим по коду та приємним для користувачів. Зі сторони сервера було використано технології Node.JS та її бібліотеку Express.JS, що дало зробити змогу зробити сервер не занадто великим, швидким та простим для розширення. Була також використана СУБД PostgreSQL. Результатом проробленої роботи є готовий продукт, який можна сміливо запускати та набирати аудиторії через різні соцмережі (TikTok, Instagram, ...).

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Bisikalo, O., Kovalenko, O., & Palamarchuk, Y. (2019). Models of behavior of agents in the learning management system. 2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT) (pp. 222-227). Lviv, Ukraine. <https://doi.org/10.1109/STCCSIT.2019.8929751>
2. Kovalenko, O., & Palamarchuk, Y. (2018). Algorithms of blended learning in IT education. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT) (pp. 382-386). Lviv, Ukraine. <https://doi.org/10.1109/STCCSIT.2018.8526605>
3. Чернишов К. А. Методи збору даних досвіду взаємодії користувача для випробувального етапу розробки через тестування [Текст] / К. А. Чернишов, І. П. Малініч, П. П. Малініч // Міжнародна наукова інтернет-конференція "Інформаційне суспільство:

технологічні, економічні та технічні аспекти становлення, м. Тернопіль, 5 лютого 2019 р. : збірник тез доповідей. – Тернопіль, 2019. – Вип. 35. – 43 с.

4. Чернишов , К., Майданюк , В., Арсенюк , І., & Малініч , І. (2021). Аналіз та класифікація вендингових систем та їх структурних реалізацій. КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ: ОСВІТА, НАУКА, ВИРОБНИЦТВО, (42), 110-118. <https://doi.org/10.36910/6775-2524-0560-2021-42-16>

Смолінський Андрій Васильович – студент групи 2СП-22б, факультет Інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця

Томчук Михайло Миколайович – студент групи ТЗД-21б, факультет Будівництва, цивільної та екологічної інженерії, Вінницький національний технічний університет, Вінниця

Томчук Микола Антонович – кандидат технічних наук, доцент кафедри Обчислювальної техніки, Вінницький національний технічний університет, Вінниця, e-mail: tomchuk@vntu.edu.ua

Andriy Smolinsky – student of group 2SP-22b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia

Mykhailo Tomchuk – student of TZD-21b group, Faculty of Construction, Civil and Environmental Engineering, Vinnytsia National Technical University, Vinnytsia

Mykola Tomchuk – Cand. Sc. (Tech), Docent of Computer Technologies department, Vinnytsia National Technical University, Vinnytsia, email: tomchuk@vntu.edu.ua