

О.С. Свістельник, магістрант**Т.М. Локтікова, ст. викладач****А.В. Морозов, к.т.н., доц.****Ю.І. Лисогор, ст. викладач****Н.О. Кушнір, ст. викладач***Державний університет «Житомирська політехніка»***К.О. Палагута, к.е.н., доц.***Державний торговельно-економічний університет*

Онлайн-сервіс проходження курсів української мови

В Україні залишається актуальною проблема недостатнього використання державної мови. Завдяки геополітичним змінам, зокрема російській агресії, підвищилася важливість української мови для самоідентифікації країни. Це стало підґрунтям для розробки онлайн-сервісу проходження курсів української мови. Вебзастосунок призначений для покращення володіння державною мовою громадянами України шляхом проходження відповідних курсів з обов'язковою перевіркою отриманих знань за допомогою виконання тестових завдань. Кожний курс поділяється на розділи. В кожному розділі користувач має можливість переглянути відеоматеріал та опрацювати опорний конспект. Для побудови додатка обрано клієнт-серверну архітектуру з використанням вебтехнологій, які не вимагають додаткового ліцензування. Серверна частина застосунку базується на вебсервері Apache2 та системі керування базами даних (СКБД) MySQL. Під час проєктування програмного додатка застосовано архітектуру SPA, з метою забезпечення користувачу досвіду, близького до використання настільної програми. Клієнтська частина побудована за сучасними стандартами створення односторінкового застосунку, тобто весь необхідний код завантажується разом зі сторінкою або динамічно підвантажується за потребою зазвичай у відповідь на дії користувача. Для реалізації програмного продукту було спроєктовано базу даних, яка складається з 15 таблиць. Під час проєктування було вирішено використовувати фреймворк Laravel, побудований за допомогою мови програмування PHP, для створення приватного API, тобто бекенд частини додатка. Для реалізації клієнтської частини використовується бібліотека мови програмування JavaScript – React. На даний момент часу це – найпопулярніше рішення для SPA архітектури. Розроблений додаток пройшов як функціональне, так і нефункціональне тестування.

Ключові слова: вебзастосунок; українська мова; SPA; API; Laravel; React.

Актуальність теми. У сучасному світі важко уявити людину, яка би не використовувала інтернет. Щодня ми чуємо нові слова, значення яких нам невідомі, і ми одразу користуємося можливостями Всесвітньої мережі для отримання експертної думки з цього приводу. Ми можемо використовувати потужності інтернету як нам заманеться, але чому би не спрямувати їх у правильне русло для нашої країни. На жаль, історично склалося так, що значна кількість населення України не розмовляє державною мовою, надаючи перевагу спілкуванню мовою окупанта, який століттями намагався нас завоювати. Проте повномасштабне вторгнення російської федерації в лютому 2022 року докорінно змінило думку більшості свідомих громадян України про важливість мови для самоідентифікації держави в світовому суспільстві. Тому й було вирішено розробити вебзастосунок, який може полегшити процес «лагідної українізації» для мешканців України, які невпевнені в своїх знаннях державної мови.

Аналіз останніх досліджень та публікацій, на які спираються автори. Для відповідного аналізу будемо спиратися на додатки зі схожою тематикою, які вже існують на просторах інтернету. В процесі пошуку було знайдено лише два застосунки, які можна вважати аналогами розробки. Розглянемо їх більш детально. Перший додаток – це Udemy, всесвітньо відома онлайн-платформа, яка надає можливість проходження курсів з різних тематик [1]. Вона пропонує величезний вибір курсів, що охоплюють різні сфери, зокрема й з вивчення української мови. Сервіс є зручним у використанні, проте серед мов інтерфейсу відсутня українська мова, а також невеликою є загальна кількість курсів, пов'язаних з вивченням української мови. Другим аналогічним додатком слугуватиме Є-мова [2]. Це освітня платформа громадської організації «Український світ», яка має за мету створити українськомовний культурний простір. До переваг цієї платформи належить наявність курсів як з вивчення, так і з популяризації української мови. Недоліками можна вважати відсутність відеоконтенту, а також те, що тематика окремих розділів є досить вузько спрямованою. На основі проведених огляду й аналізу можна припустити, що розроблений застосунок є унікальним для ринку.

Метою статті є дослідження розробки онлайн-сервісу проходження курсів української мови для покращення знань громадян України.

Викладення основного матеріалу. Для отримання повної картини застосунку було здійснено постановку завдання. Розробка онлайн-сервісу проходження курсів української мови складається зі створення фронтенд-частини застосунку за допомогою бібліотеки React [3] мови програмування JavaScript [4] та бекенд-частини за допомогою фреймворку Laravel [5] мови програмування PHP [6].

Розроблений застосунок являє собою вебсайт, який надає можливість користувачу покращити знання з українською мови шляхом проходження курсів.

Основні етапи розробки системи такі:

1. Створення бекенд-частини додатка шляхом написання приватного API (Application Programming Interface) [7] із використанням фреймворку Laravel, який містить реалізацію таких сутностей:

- сутність «Авторизація»;
- сутність «Профіль користувача»;
- сутність «Курс»;
- сутність «Тег»;
- сутність «Розділ курсу»;
- сутність «Тест»;
- сутність «Оплата»;
- сутність «Завантаження великих файлів».

2. Проектування інтерфейсу користувача:

- вибір бібліотеки React для реалізації користувацького інтерфейсу;
- розробка відповідних сторінок за допомогою вибраної бібліотеки;
- реалізація сповіщення користувача про завантаження контенту за допомогою відповідного елемента.

3. Налаштування обміну даними поміж створеним API й клієнтською частиною та розробка роутингу застосунку за допомогою бібліотеки React:

- реалізація виклику API за допомогою відповідної бібліотеки (Axios);
- налаштування роутингу додатка за допомогою відповідної бібліотеки (React Router);
- реалізація розбиття відеоматеріалу на шматки (chunks) для завантаження його на сервер за допомогою відповідної бібліотеки (React-Uploady).

У результаті виконання перерахованих вище етапів було розроблено систему, що побудована на основі архітектури SPA (Single-Page Application) [8], яка надає користувачу зручну платформу для навчання.

Обґрунтуємо вибір архітектури SPA, на основі порівняльної характеристики з не менш популярною архітектурою MVC (Model-View-Controller) [9].

MVC – це архітектурний шаблон, який дозволяє розділити компоненти додатка на три основні частини: модель (model), вигляд (view) і контролер (controller). Модель представляє дані додатка, вигляд відповідає за відображення інформації користувачу, а контролер обробляє запити користувача та забезпечує взаємозв'язки з моделлю та виглядом.

До переваг архітектури MVC належать:

- забезпечення розподілення обов'язків, що дозволяє чітко розділити логіку додатка на три основні компоненти – модель, вигляд і контролер. Це в свою чергу призводить до кращого розуміння та підтримки програмного коду, спрощення тестування;

- пришвидшення побудови та проектування системи. Завдяки розділенню відповідальностей поміж різними компонентами, команда розробників може паралельно працювати над різними аспектами додатка, що в свою чергу призводить до пришвидшення процесу його розробки.

Недоліками архітектурного шаблону MVC вважаються:

- надмірна складність при створенні примітивних додатків. Для розробки проєктів із невеликим функціоналом, використання повного шаблону MVC може бути надлишковим, це в свою чергу призводить до зайвого навантаження та підвищення складності програмного коду;

- залежність від контролера. В деяких випадках контролер може стати найбільш навантаженою частиною додатка, а це, зокрема, впливає на його швидкодію.

SPA – це архітектурний підхід, за якого вебдодаток повністю завантажується при першому зверненні до нього, а взаємодія з сервером відбувається через асинхронні запити (наприклад, за допомогою AJAX). Цей підхід дозволяє створювати більш інтуїтивно зрозумілий інтерфейс користувача, зменшити час завантаження сторінок і підвищити продуктивність додатка.

Переваги архітектури SPA є такими:

- покращення користувацького досвіду. Код додатка повністю завантажується під час початкового завантаження, а в подальшому навігація та взаємодія з додатком здійснюються без повного

перезавантаження сторінки. Це, зокрема, призводить до більш плавної та швидкої реакції на дії користувача;

- зменшення часу завантаження сторінок. Сторінки не перезавантажуються, тому немає необхідності повторно завантажувати такі ресурси, як HTML, CSS, JavaScript, за кожної навігації. Саме це призводить до зменшення часу завантаження сторінок і відповідно покращення продуктивності додатка;

- гнучкість розробки. Створення клієнтської та серверної частин може відбуватися незалежно одна від одної. Це дозволяє в команді розробників працювати паралельно та використовувати різні технології для розробки бекенду та фронтенду.

Недоліки архітектури SPA:

- клієнтська сторона є більш навантаженою. Оскільки код додатка повністю завантажується на клієнтському пристрої, то це вимагає застосування більших апаратних ресурсів з боку користувача;

- складність SEO-оптимізації. Традиційні пошукові системи не завжди ефективно індексують вміст сторінок додатка, який динамічно генерується на клієнтському боці.

У результаті проведених огляду й аналізу архітектур було обрано архітектуру SPA, для забезпечення досвіду користувачу, близького до користування настільною програмою.

На сьогодні неможливо уявити розробку додатків за допомогою використання тільки певних мов програмування. При цьому використовуються фреймворки та бібліотеки, написані за допомогою цих мов.

Для створення бекенд-частини додатка було обрано мову програмування PHP. Для неї існує чимала кількість фреймворків, найпопулярнішими з яких є Symfony, Laravel та Yii2. Зі вказаних фреймворків було вибрано Laravel, який, на відміну від Yii2, дозволяє створювати API без витрачання додаткових ресурсів та, на відміну від Symfony, має велику спільноту розробників та детальну документацію.

Для реалізації клієнтської частини додатка використано бібліотеку мови програмування JavaScript – React. На даний час це – найпопулярніше рішення для розробки додатків за архітектурою SPA. Якщо порівнювати з іншими засобами, наприклад, Vue та Angular, то бібліотека React має одну з найбільших спільнот розробників, що в свою чергу призводить до більш широкої її підтримки, кількості розширень та доступу до багатофункціональних бібліотек.

Як систему керування базами даних було обрано MySQL. Вибір цієї технології ґрунтується на таких її перевагах:

- широка підтримка завдяки великій спільноті розробників і користувачів;
- висока продуктивність завдяки можливості масштабування для обробки великих обсягів даних та значного навантаження.

Діаграма варіантів використання, яка наочно відображає відношення між акторами та прецедентами в системі, зображена на рисунку 1.

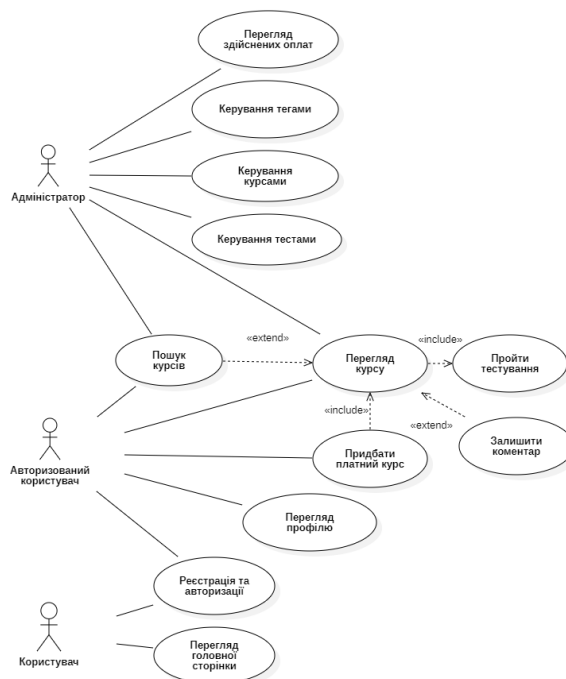


Рис. 1. Діаграма варіантів використання системи

Відповідно до побудованої діаграми варіантів використання були визначені такі вимоги до системи.

Бізнес-вимоги:

1. Основні цілі: проєкт створено з метою вивчення української мови за допомогою онлайн-ресурсу.
2. Можливості: перегляд курсів з різних розділів української мови та придбання платних курсів.
3. Представлення проєкту: проєкт реалізовано у вигляді вебдодатка, що має зручний інтерфейс та необхідну інформацію за даною тематикою.

Вимоги користувачів:

- Зовнішні користувачі – авторизований користувач:
 1. Можливість пошуку відповідного курсу.
 2. Можливість фільтрації та сортування курсів.
 3. Можливість перегляду обраного курсу та залишення коментаря про нього.
 4. Можливість проходження тестування за тематикою курсу.
 5. Можливість придбання платного курсу онлайн й отримання доступу до нього.
 6. Можливість перегляду профілю та інформації про проходження курсів.
- Внутрішній користувач – адміністратор:
 1. Керування вмістом вебдодатка.
 2. Перегляд успішних оплат курсів.

Функціональні вимоги:

1. Підключення сторонніх API для здійснення оплати й авторизації за допомогою Google.
2. Створення функціоналу для проходження тестів.
3. Використання існуючих пакетів для пришвидшення процесу розробки інтерфейсу користувача.

У зв'язку зі швидким розвитком IT-сфери, для вирішення питань, які часто зустрічаються на практиці, використовуються патерни проєктування. Без їхнього використання неможливо уявити розробку сучасних застосунків. Оскільки патерн «Репозиторій» є найбільш популярним вибором для застосунків, розроблених за архітектурою SPA, то було прийнято рішення щодо його використання.

«Репозиторій» являють собою класи або компоненти, що інкапсулюють логіку, необхідну для доступу до джерел даних. Вони забезпечують спільну функціональність доступу до даних. Завдяки відокремленню інфраструктури або технології, які використовуються для доступу до баз даних, від рівня моделі предметної області покращується зручність обслуговування додатка.

На рисунку 2 наведено реалізацію патерна «Репозиторій» для модуля авторизації.



Рис. 2. Реалізація патерна «Репозиторій» для модуля авторизації

Була розроблена база даних додатка, яка складається з 15 таблиць. ER-діаграма спроектованої бази даних зображена на рисунку 3. Для збереження даних було обрано СКБД MySQL. MySQL має широкий функціонал, високі надійність і швидкодію, є безкоштовною та відкритою, що дає можливість економити на ліцензійних витратах.

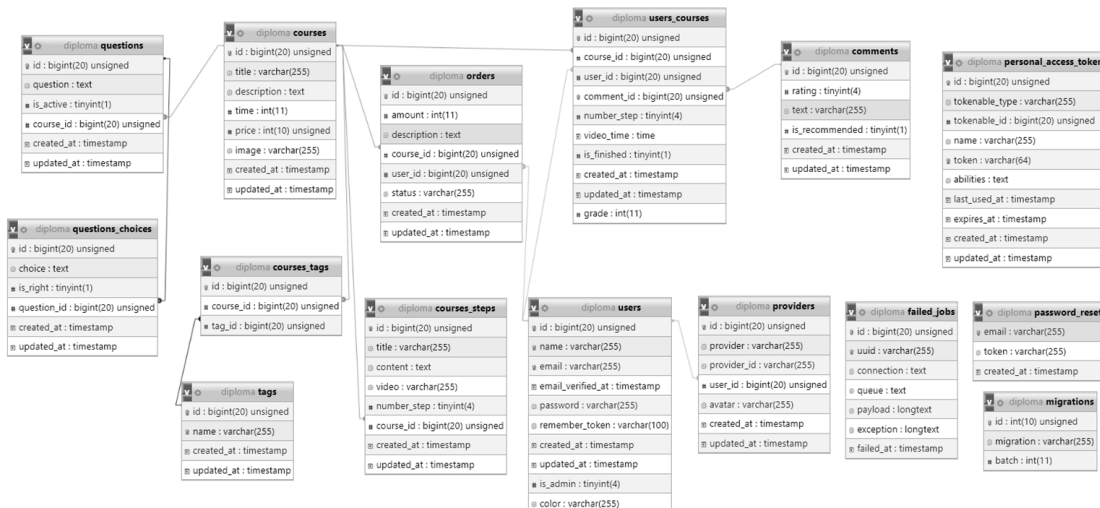


Рис. 3. ER-діаграма спроектованої бази даних

Вибір архітектури SPA для створення додатка передбачає використання підходу RESTful API (Representational State Transfer API) [11].

RESTful API є сукупністю правил та протоколів, які дозволяють різним програмним додаткам взаємодіяти поміж собою через мережу, зазвичай через інтернет.

На рисунку 4 зображено діаграму компонентів розробленого додатка, яка демонструє взаємодію компонентів поміж собою.

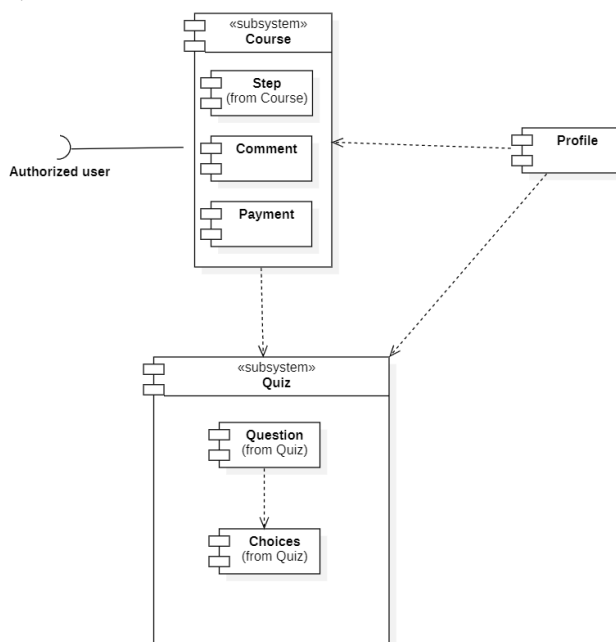


Рис. 4. Діаграма компонентів додатка

Із діаграми рисунку 4 видно:

1. Для роботи з основними компонентами додатка користувач має бути авторизованим у системі.
2. Система має у своєму складі два підкомпоненти (Course та Quiz) й один окремий компонент (Profile).
3. Підкомпонент Course відповідає за проходження певного курсу, що поділений на розділи, його оплату та залишення коментаря до нього.
4. Підкомпонент Quiz відповідає за тестування, що є обов'язковою умовою завершення відповідного курсу.
5. Компонент Profile відповідає за керування профілем та відстежування прогресу користувачем.

Невід’ємною складовою проектування будь-якої системи є побудова алгоритмів її роботи та їхня подальша реалізація. Розглянемо деякі з алгоритмів роботи для розробленого додатка. Важливою частиною застосунку є розділи курсів та їхня послідовність. Тому обов’язковою складовою розробки додатка є реалізація зміни послідовності розділів адміністратором. Спершу отримується початкова послідовність розділів та записується до масиву. Після цього здійснюється циклічна зміна послідовності.

На рисунку 5 показано код реалізації методу `changeRank` класу `CourseStepRepository`, що за це відповідає.

```
public function changeRank(Request $request, int $course_id): JsonResponse
{
    $steps = [];
    foreach ($request->all() as $item) {
        $steps[] = CourseStep::where(['course_id' => $course_id, 'number_step' => $item['number_step']])->first();
    }
    foreach ($steps as $id => $step) {
        $step->number_step = $id + 1;
        $step->save();
    }

    $steps = CourseStep::where(['course_id' => $course_id])->orderBy('number_step')->get();

    return $this->success( message: 'Порядок успішно змінено', ['steps' => $steps]);
}
```

Рис. 5. Код реалізації методу `changeRank` класу `QuestionRepository`

Найбільш важливою та відповідальною частиною додатка є проведення тестування й перевірка правильності відповідей користувача. Розглянемо більш детально відповідний алгоритм. У масив заносяться усі відповіді користувача. Потім циклічно знаходиться варіант відповіді та здійснюється перевірка, чи він є правильним для цього запитання. Якщо так, то кількість правильних відповідей збільшується, інакше правильна відповідь заноситься до масиву. Після підрахунку оцінка зберігається, якщо вона виявляється кращою, аніж у попередніх спробах, або якщо ця спроба є першою. Після цього курс у системі позначається як завершений для такого користувача. На рисунку 6 показано код реалізації методу `checkAnswers` класу `QuestionRepository`, що відповідає за перевірку відповідей.

```
public function checkAnswers(Request $request): JsonResponse
{
    try {
        if (!$userCourse = UserCourse::where(['course_id' => $request->all()[0]['course_id'], 'user_id' => auth()->id()])->first()) {
            return $this->error( message: 'Спочатку необхідно розпочати курс', statusCode: Response::HTTP_FORBIDDEN);
        }

        $answers = [];
        $right = 0;
        foreach ($request->all() as $id => $item) {
            $choice = QuestionChoice::findOrFail($item['choice']['id']);
            $rightAnswer = QuestionChoice::where(['is_right' => 1, 'question_id' => $item['id']])->first();
            if ($choice->is_right) {
                $answers['res'][$id] = ['res' => true];
                $right++;
            } else {
                $answers['res'][$id] = ['res' => false, 'right' => $rightAnswer->choice];
            }
        }

        $total = round( num: $right / count($request->all()) * 100);
        $answers['total'] = $total;
        if ($userCourse->grade < $total || !$userCourse->grade) {
            $userCourse->grade = $total;
        }

        $userCourse->is_finished = 1;
        $userCourse->save();

        return $this->success( message: 'Відповіді успішно опрацьовано', ['answers' => $answers]);
    } catch (\Throwable $th) {
        return $this->error($th->getMessage());
    }
}
```

Рис. 6. Код реалізації методу `checkAnswers` класу `QuestionRepository`

На рисунку 7 представлено сторінку певного курсу розробленого додатка. Вона відображає загальну інформацію про курс, перелік його розділів та коментарі, залишені іншими користувачами.

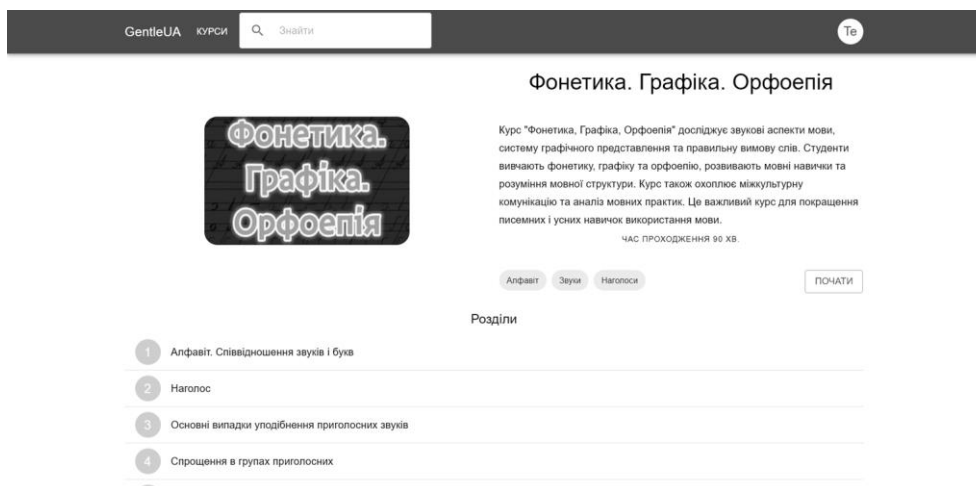


Рис. 7. Сторінка курсу «Фонетика»

На рисунку 8 показано сторінку тесту та результати його проходження.

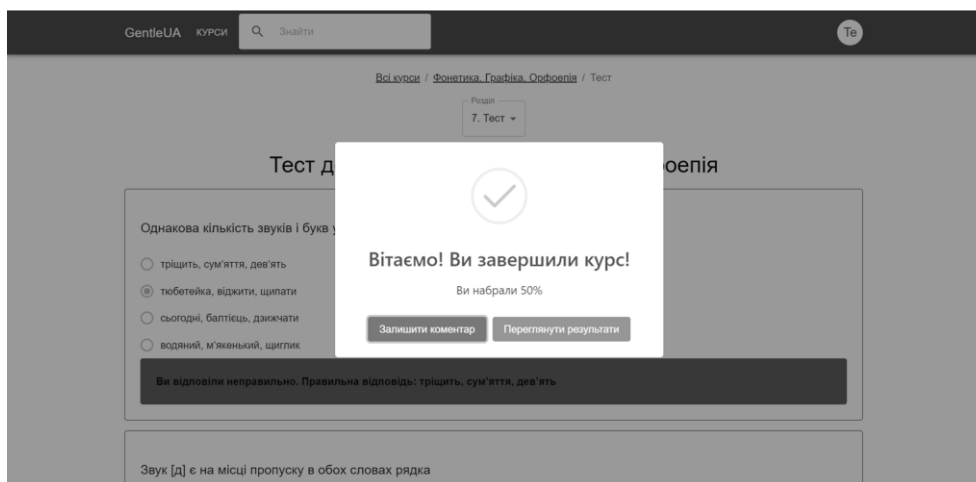


Рис. 8. Результат проходження тесту

Висновки та перспективи подальших досліджень. Під час дослідження було побудовано та спроектовано програмний продукт, який відповідає оголошеним вимогам та виконує поставлені перед ним завдання. Онлайн-сервіс може бути використаний для покращення знань з української мови громадянами України. При розробці вебдодатка застосовані сучасні засоби, а саме: архітектура SPA, фреймворк Laravel мови програмування PHP для створення бекенду, бібліотека React мови програмування JavaScript для створення фронтенду. Проведене тестування показало, що розроблений вебзастосунок є готовим до використання. В майбутньому цей програмний продукт може розвиватися шляхом виявлення потреб у новому функціоналі, його реалізації та оновлення для всіх користувачів системи.

Список використаної літератури:

1. Online Courses – Learn Anything, On Your Schedule [Електронний ресурс]. – Режим доступу : <https://www.udemy.com/>.
2. Є-мова – Вивчай українську мову з нами! [Електронний ресурс]. – Режим доступу : <https://course.emova.language-ua.online/>.
3. React – JavaScript-бібліотека для створення користувацьких інтерфейсів [Електронний ресурс]. – Режим доступу : <https://uk.legacy.reactjs.org/>.
4. Сучасний підручник з JavaScript [Електронний ресурс]. – Режим доступу : <https://uk.javascript.info/>.
5. Laravel – The PHP Framework For Web Artisans [Електронний ресурс]. – Режим доступу : <https://laravel.com/>.

6. PHP Manual [Електронний ресурс]. – Режим доступу : <https://www.php.net/manual/en/index.php>.
7. Що таке API? Просте пояснення від Петра Газарова [Електронний ресурс]. – Режим доступу : <https://dev.ua/news/chto-takoe-api-prostym-yazykom>.
8. Lawson K. What Are Single Page Applications and Why Do People Like Them So Much? / K. Lawson [Електронний ресурс]. – Режим доступу : https://www.bloomreach.com/en/blog/2018/what-is-a-single-page-application?spz=learn_var.
9. MVC Design Pattern – GeeksforGeeks [Електронний ресурс]. – Режим доступу : <https://www.geeksforgeeks.org/mvc-design-pattern/>.
10. What is Use Case Diagram? [Електронний ресурс]. – Режим доступу : <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>.
11. What is REST - REST API Tutorial [Електронний ресурс]. – Режим доступу : <https://restfulapi.net/>.
12. What is Component Diagram? [Електронний ресурс]. – Режим доступу : <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/>.

References:

1. «Online Courses – Learn Anything, On Your Schedule», [Online], available at: <https://www.udemy.com/>
2. «Ye-mova – Vuychai ukrainsku movu z namy!», [Online], available at: <https://course.emova.language-ua.online/>
3. «React – JavaScript-biblioteka dlia stvorennia korystuvatskykh interfeisiv», [Online], available at: <https://uk.legacy.reactjs.org/>
4. «Suchasnyi pidruchnyk z JavaScript», [Online], available at: <https://uk.javascript.info/>
5. «Laravel – The PHP Framework For Web Artisans», [Online], available at: <https://laravel.com/>
6. «PHP Manual», [Online], available at: <https://www.php.net/manual/en/index.php>
7. «Shcho take API? Proste poiasnennia vid Petra Hazarova», [Online], available at: <https://dev.ua/news/chto-takoe-api-prostym-yazykom>
8. Lawson, K. (2018), «What Are Single Page Applications and Why Do People Like Them So Much?», [Online], available at: https://www.bloomreach.com/en/blog/2018/what-is-a-single-page-application?spz=learn_var
9. «MVC Design Pattern – GeeksforGeeks», [Online], available at: <https://www.geeksforgeeks.org/mvc-design-pattern/>.
10. «What is Use Case Diagram?», [Online], available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
11. «What is REST – REST API Tutorial», [Online], available at: <https://restfulapi.net/>
12. «What is Component Diagram?», [Online], available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/>

Свістельник Олександр Сергійович – магістрант Державного університету «Житомирська політехніка».

<https://orcid.org/0009-0009-2371-9695>.

Наукові інтереси:

- інформаційні системи та технології.

Локтікова Тамара Миколаївна – старший викладач кафедри інженерії програмного забезпечення Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0002-3525-0179>.

Наукові інтереси:

- цифрова обробка зображень;
- інформаційні системи та технології.

Морозов Андрій Васильович – кандидат технічних наук, доцент, проректор з науково-педагогічної роботи Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0003-3167-0683>.

Наукові інтереси:

- комбінаторна оптимізація;
- інформаційні технології.

Лисогор Юрій Іванович – старший викладач кафедри інженерії програмного забезпечення Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0003-1194-2813>.

Наукові інтереси:

- комп'ютерна графіка та дизайн;
- цифрова обробка сигналів;
- інформаційні системи та технології.

Кушнір Надія Олександрівна – старший викладач кафедри інженерії програмного забезпечення Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0002-0797-3687>.

Наукові інтереси:

- комбінаторна оптимізація;
- інформаційні технології.

Палагута Катерина Олексіївна – кандидат економічних наук, доцент, доцент кафедри інженерії програмного забезпечення та кібербезпеки Державного торговельно-економічного університету.

<https://orcid.org/0000-0003-1167-9509>.

Наукові інтереси:

- Web-програмування;
- інформаційні технології в економіці.

Svistelnyk O.S., Loktikova T.M., Morozov A.V., Lysogor Y.I., Kushnir N.O., Palahuta K.O.

Online service for taking Ukrainian language courses

In Ukraine, the problem of insufficient use of the state language remains relevant. Due to geopolitical changes, including russian aggression, the importance of the Ukrainian language for the country's self-identification has increased. This became the basis for the development of an online service for taking Ukrainian language courses. The web application is designed to improve the Ukrainian language proficiency of Ukrainian citizens by taking relevant courses with mandatory testing of the knowledge gained by completing test tasks. Each course is divided into sections. In each section, the user has the opportunity to watch a video and work through a reference summary. To build the application, a client-server architecture, using web technologies that do not require additional licensing, was chosen. The server side of the application is based on the Apache2 web server and the MySQL database management system. When designing the software application, the SPA architecture was used to provide the user with an experience similar to that of a desktop program. The client side is built according to modern standards for creating a Single-Page Application, meaning that all the necessary code is loaded along with the page or dynamically loaded as needed, usually in response to user actions. To implement the software product, a database consisting of 15 tables was designed. During the design process, it was decided to use the Laravel framework built using the PHP programming language to create a private API, i.e. the backend of the application. To implement the client side, the JavaScript programming language library React is used. At the moment, this is the most popular solution for SPA architecture. The developed application has passed both functional and non-functional testing.

Keywords: web application; Ukrainian language; SPA; API; Laravel; React.

Стаття надійшла до редакції 14.09.2023.