

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

Факультет комп'ютерних наук, фізики та математики

Кафедра інформатики, програмної інженерії та економічної кібернетики

**РОЗРОБЛЕННЯ БЕКЕНД ВЕБРЕСУРСУ ДЛЯ НАВЧАННЯ УЧНІВ НА
ЕТАПІ БАЗОВОЇ СЕРЕДНЬОЇ ОСВІТИ**

Кваліфікаційна робота (проект)

на здобуття ступеня вищої освіти «бакалавр»

Виконав: студент 4 курсу 431 групи
Спеціальності: 122 Комп'ютерні науки
Освітньо-професійної програми:
Комп'ютерні науки
Чертушкін Роман Євгенійович
Керівники: кандидат педагогічних наук,
доцент
Вінник Максим Олександрович
кандидат фізико-математичних наук,
доцент
Єрмолаєв Вадим Анатолійович
Рецензент: кандидатка педагогічних наук,
доцентка
Гончаренко Тетяна Леонідівна

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	3
ВСТУП.....	4
РОЗДІЛ 1.....	6
Постановка задачі та теорія тестування.....	6
1.1. Постановка задачі.....	6
1.2. Навчання за допомогою онлайн-тестів.....	6
1.3. Спектр застосування та основні переваги.....	9
1.4. Вимоги до проекту.....	10
Розділ 2.....	11
Теоретичні основи розробки web-сервісів.....	11
2.1 Основні поняття розробки web-сервісів.....	11
2.2 Веб-сервер.....	12
2.3 Етапи розробки.....	13
2.4 Технології розробки.....	15
2.5 Модель взаємодії веб-серверу та сайту.....	16
Розділ 3.....	19
Розробка серверної складової веб-сервісу.....	19
3.1 Програмні засоби.....	19
3.2 Архітектура додатку.....	20
3.3 Model View Controller.....	22
3.4 Контролери Арі та роутинг.....	26
3.5 Взаємодія серверу та веб-додатку.....	29
ВИСНОВКИ.....	30
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	31

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД	База даних
ПК	Персональний комп'ютер
API	Application Programming Interface
CRUD	Create Read Update Delete
FTP	File Transfer Protocol
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
MVC	Model View Controller
SQL	Structured Query Language
Бекенд	Сервер

ВСТУП

Сьогодні багато процесів в нашому житті перейшли в інформаційну мережу. Люди все більше проводять важливі заходи, роботу і навчання в інтернет. Зараз, напевно у кожного є смартфон або ноутбук, а також швидкий інтернет. Тенденція перекладу всього в онлайн на даний час розглядається як ніколи важливіше. 2020 рік показав, що епідемія коронавіруса змусила вчитися багатьох школярів вдома. Звичайно ж це спровокувало багато негативних наслідків, у вигляді неправильної організації навчального процесу.

У нашій країні, а також в деяких інших пострадянського простору, процес навчання вже прийшов в інформаційне середовище. Дивлячись на можливість контролю знань і зацікавленість дітей в Гейміфікаціях, можна з упевненістю сказати що процес навчання набагато краще використовувати в інформаційному середовищі з використанням ігрових елементів. Виходячи з цього ми вирішили розробити інструмент для вчителів і учнів, який зможе зацікавити перших отримувати нові знання.

Актуальність теми. В даний час якісний сайт формує позитивні освітні послуги установи, а також просуває платні послуги шкіл і ліцеїв. У 2020 році, коли карантин змусив вчитися мільйони школярів онлайн по будинках потреба в якійсній онлайн-освіті зросла в рази.

Мета дослідження – розробка серверної частини для сервісу здійснення педагогічного тестування.

Завдання дослідження:

1. Аналіз теорії тестування.
2. Аналіз систем розробки.
3. Складання діаграм прецедентів .
4. Розглядання сучасних підходів та технологій у сфері бекенд розробки.

Об'єкт дослідження: технологій створення Restful API (серверної частини)

Предмет дослідження: веб-додаток для проходження тестів.

Практичне значення полягає в наданні учням та викладачам шкіл можливості проходити тестування та створювати їх, у вигляді гри-тесту.

Структура дослідження. Дипломна робота складається зі вступу, списку умовних скорочень, трьох розділів, висновків, списку використаних джерел.

РОЗДІЛ 1

Постановка задачі та теорія тестування

1.1. Постановка задачі

Кожне створення продукту починається з опису проблеми. У нашому випадку завдання полягає в розробці веб-додатку для тестування дітей, який має всі відповідні функції та відповідає потребам користувачів.

На сьогоднішній день майже у всіх шкіл є комп'ютерні класи, а у учнів смартфони з виходом в інтернет, тому й проходити опитування їм буде набагато зручніше за допомогою смартфона. Найважче у роботі з дітьми зацікавити їх проходити опитування. Для цього треба розробити інтерактивні тести у вигляді гри., таким чином вірогідність проходження тесту та засвоєння нової інформації вразі зросте.

Всі випробування повинні виконуватись під час випробування вчителя. Крім того, викладач може легко скопіювати стандартизований тест, змінивши необхідні кроки. Це означає, що потрібен центр обробки даних на стороні сервера – БД.

Постановка задачі: розробити веб-додаток для проходження опитувань та тестувань у вигляді ігрових тестів. Ознайомитись з теорією тестування, аналогами та сучасними технологіями розроблення веб-додатків. Розробити серверну частину веб-сервісу.

1.2. Навчання за допомогою онлайн-тестів

Хоча тести в наш час мають погану репутацію, дослідження показують, що часте тестування допомагає учням запам'ятовувати інформацію як при традиційному, так і при дистанційному навчанні. Інноваційні методики навчання, в яких застосування онлайн-тести дозволяють перетворити напружені контрольні роботи в задоволення для всього класу.

Щотижневе тестування

Щотижневе або навіть щоденне тестування допомагає учням застосувати вивчений матеріал на практиці. Це практичне повторення пояснює важливі факти, готуючи учнів до більш серйозних перевірок, таким як стандартні тести і випускні іспити. Крім того, часте тестування робить менш напруженими такі стресові ситуації.

[18] Вчитель повинен надати учням онлайн-тести з перевіркою матеріалу, викладеного їм під час занять. Короткі та неформальні тести, змушують учнів бути уважними на заняттях, в той же час допомагаючи оцінювати рівень знань в реальному часі.

Тести та заняття у класі

[15] Коли тести стають частиною занять в класі, а не засобом вимірювання успішності або виставлення підсумкових оцінок - вони відкривають можливості для спільної роботи учнів. Крім того, використання спільних тестів в якості компонента методик навчання спонукає вчитися краще, щоб мати можливість допомогти іншим досягти успіху.

Звичайно, в кожному спільному тесті повинен бути баланс між груповою роботою і індивідуальною оцінкою. Для досягнення цього балансу спробуйте випадковим чином вибрати 2 учнів для написання «шпаргалки» на дошці перед кожним тестом. Ці учні повинні спільними зусиллями, не звертаючись до підручників і конспектів, написати шпаргалку протягом трьох хвилин або менше. Потім учні самі проходять тест, користуючись складеної ними шпаргалкою.

Корисні нотатки у тестах

Дослідження показують, що детальний зворотний зв'язок щодо виправлення помилок - важлива частина методик викладання, так як вона

сприяє кращому засвоєнню інформації. Крім того, коли викладач пояснює помилки відразу, учень з більшою ймовірністю зможе згадати його слова пізніше. На щастя, онлайн-тести дозволяють легко дати (і отримати) негайний зворотний зв'язок.

Треба надати учням змогу знайти правильну відповідь наступного разу, коли їм зустрінеться питання, доповнюючи онлайн-опитування спеціально складеними відгуками. Щоб досягти найкращого результату, треба включити у відгук як правильну відповідь, так і корисний контекст, що дозволяє цю відповідь запам'ятати. Також можна провести аналітику тесту, щоб отримати цілісне уявлення про успішність класу.

Питання різних типів

[15] У тесті слід використовувати різноманітні питання, це ключ до збереження інтересу учнів у подальшому. Ці питання також допомагають дітям ефективно застосовувати свої знання в різних ситуаціях. Підтримати інтерес учнів, можна використовуючи різні типи питань в тестах, наприклад:

Відповіді з множиною варіантів. Ці типи питань відмінно підходять для тестів початкового рівня, так як вони прості і не вимагають багато часу для відповіді.

Щоб попросити учнів описати складні поняття, використовуйте питання з відповідями в довільній формі. Хоча такі відповіді важче оцінювати, вони дають учням можливість розповісти саме те, що вони знають.

Допоможіть Вашим учням швидко дати короткі відповіді, використовуючи питання з одним текстовим полем. Ці питання годяться для швидких перевірочних тестів, так як для відповіді на них потрібно трохи часу.

Перевірте здатність Ваших учнів розставляти в потрібному порядку важливі події та іншу інформацію за допомогою запитань з ранжируванням.

Питання цього типу оптимальні для підсумкових контрольних з певної теми, такий як етапи наукового методу або історичні події.

1.3.Спектр застосування та основні переваги

Спектр застосування онлайн тестування досить широкий:

- Кадрові агентства використовують дану методику при підборі претендентів для абсолютно різних цілей.
- Роботодавці використовують онлайн тестування як альтернативний метод стандартного співбесіди з претендентом.
- Компанії середнього і малого бізнесу онлайн тестування застосовують при маркетингових дослідженнях і виборі оптимальної стратегії розвитку підприємства.
- Тестування у школах та інститутах.

Основними переважними ознаками будь-яких видів онлайн-тестування є:

- проведення віддаленого тестування (можливість проведення процедури тестування в режимі реального часу в будь-якій точці, де є доступ до Інтернету);
- миттєва обробка результатів і отримання повної звітності;
- доступ до результатів лише уповноважених осіб (вхід в систему тестування здійснюється на підставі встановлених паролів);
- збереження звітів тестування у вигляді файлових даних;
- тестування може відбуватися в будь-який час і в будь-якому порядку;
- можливість охоплення кількох тем певного питання.

Існує один явний недолік - це досить трудомістке складання самих анкет і тестів. Однак і тут створені програми, в яких дані функції виконує комп'ютеризована система.

1.4.Вимоги до проекту

Створення будь-якого проекту починається з постановки задачі. У нашому випадку - це розробка бекенд частини для веб-сервісу тестування дітей перших - дев'ятих класів.

На сьогоднішній день багато шкіл за допомогою інтернету надають дітям багато цікавих та інтерактивних завдань. При такому підході дітям зрозуміліше та легше запам'ятовувати інформацію та проходити всілякі тести.

Сутність нашого сайту полягає в наданні тестування для дітей у вигляді гри. Всі тести повинні створюватися за допомогою конструктора в якому вчитель буде писати питання і правильні відповіді. Учні ж в свою чергу повинні проходити тести у вигляді гри. Результати тестування будуть надаватися у вигляді звіту, тому веб-додатку потрібно мати серверну частину та БД.

Постановка задачі: розробити веб-додаток для проходження опитувань та тестувань дітей перших – дев'ятих класів. Мінімізувати функціонал, щоб задовольнити запити користувачів. Дізнатися про передові технології розробки бекенд складової веб-сайту. Перевірити правильність веб-додатків та серверного компоненту. У роботі над проектом використовувати знання про організацію опитувань.

Розділ 2

Теоретичні основи розробки web-сервісів

2.1 Основні поняття розробки web-сервісів

Без інтернету на сьогоднішній день не може жити переважна частина населення. Дана технологія пов'язує людей з усього світу, з різних куточків Землі. Дуже важко знайти людину, яка не користувався хоч раз Інтернетом, ще складніше знайти людину, яка про нього не чула. Все частіше в різних країнах всесвіту люди переходять на зручне онлайн-навчання. Це зумовлено тим, що такий прийом рівняється на життя людини та дає можливість здобути інформацію “Тут и зараз”.

При розробці сучасного програмного забезпечення для інтернету все більше уваги видається такий тип додатків, як web-сервіси. У розвинених країнах, інтернет вже давно використовуються в області навчання: за допомогою використання web-сервісів можна навчатися там де це було неможливо до цих пір. Такі технології використовуються зазвичай у сфері виставлення балів, тестування та проведення онлайн уроків.

У зв'язку зі сказаним, можна вважати, що розробка та застосування web-сервісів для надання освітніх послуг, є досить актуальною темою в сучасному суспільстві. Існує досить багато визначень поняття web-сервіс. Але найчастіше web-сервіс визначається як - web-ресурс зі стандартизованими інтерфейсами, який може взаємодіяти з іншими програмними модулями за допомогою повідомлень, заснованих на визначених стандартах і протоколах.

Головною відмінністю web-сервісу від веб-сайту є, те, що web-сервіс надає певні послуги. Наприклад, призначений для користувача веб-сервіс для тестування школярів перших-восьмих класів пропонує вчителю певний набір інструментів для створення власних курсів. Що стосується веб-сайт, це

ресурс на якому розташовується будь-яка інформація. Заздалегідь проаналізувавши кілька популярних інтернет-ресурсів можна зробити висновок, що дизайн і навігація відіграє важливу роль в залученні уваги користувачів.

Навігація в веб-сервісі повинна бути продуманою і зрозумілою, це позитивно позначиться на користувацькому досвіді. Існує безліч мов, на яких можна розробити веб-ресурси. Ми ж будемо використовувати Vue.js для роботи з інтерфейсом, а також PHP для роботи з базами даних.

Сам по собі веб-сервіс використовує ідею модульності, при розробці програмного продукту використовують модулі, з яких складається майбутній користувацький інтерфейс з певними функціями.

2.2 Веб-сервер

З точки зору апаратної частини веб сервер-це комп'ютер, призначений для зберігання даних сайтів. Також він призначений для доставки файлів у веб-браузер користувачів. Веб сервер може бути доступний через певне доменне ім'я.

Якщо дивитися з точки зору програмного забезпечення, то веб сервер не може обійтися без компонентів, призначених для контролю доступу користувачів до даних, які знаходяться на сервері. В якості таких компонентів виступають HTTP-сервер (як частина програмного забезпечення) і HTTP-протоколу за допомогою якого спілкуються web-сервер і браузер.

На самому першому рівні, коли браузеру користувача потрібно якийсь певний файл, який знаходиться на сервері, він запитує його за допомогою HTTP-протоколу. Після того, як запит досягне певного сервера, сервер HTTP зможе прийняти запит, потім знайде потрібний користувачеві документ, після чого відправить його назад. Якщо документ не буде знайдений, сервер

повідомить про помилку під номером 404. Якщо передача файлів завершилася надходить повідомлення з кодом 200 ОК, що означає успішний запит до серверу.

Цих кодів дуже багато, основні з них:

- 200 ОК - успішно;
- 201 Created - створено;
- 400 Bad Request – невірний запит;
- 401 – користувач не авторизований;
- 402 Payment Required – необхідна плата;
- 403 Forbidden – немає рівня доступу;
- 500 Internal Server Error – внутрішня помилка серверу

Веб сервер може обслуговувати один або кілька сайтів. Кожен окремий сайт в термінах веб сервера називається «віртуал хост». Сайт може бути доступний як по одному доменному імені так і по декількох. Тоді одне доменне ім'я називається основним, а решта синонімами. Наприклад один веб-сервер віртуального хостингу розміщує до 300-400 незалежних сайтів.

2.3 Етапи розробки

Виділяють кілька етапів розробки web-сервісу:

1. Планування.
2. Реалізація.
3. Тестування.
4. Публікація.

5. Рекламування.

6. Супровід.

Розглянемо ці етапи детальніше:

Планування. На цьому етапі повинні бути отримані відповіді на наступні питання: мета створення web-сервісу, аудиторія, на яку він розрахований, яка інформація буде розміщена. Доцільно познайомитися з сайтами, тематика яких близька до тематики створюваного web-сервісу, з тим, щоб уникнути помилок або повчитися на позитивних прикладах. Визначаються категорії, підрозділи сайту, кількість сторінок в кожному розділі, підрозділі. Розробляється структурна (навігаційна) схема сайту.

Реалізація. На цьому етапі проводиться підготовка текстового та графічного матеріалу. Матеріал розбивається по файлах відповідно до структури. Організовувати посилання між web-сторінками сайту. Розробляється дизайн. На цьому етапі визначаються: фон, на якому буде розташована інформація, колір тексту, колір гіпертекстових посилань, знаки навігації, виділення розділів графікою або текстовими посиланнями, представлення фотографічного матеріал.

Тестування. Після того як web-сервіс створений, необхідно перевірити правильність переходів зі сторінки на сторінку. У зв'язку з тим, що браузері різних фірм іноді по-різному інтерпретують команди розмітки HTML документу, а браузері застарілих версій "не розуміють" описів, що відповідають новим стандартам. Бажано переглянути web-сервіс в браузерах різних виробників і різних версій, при різних дозволах екрану.

Публікація. Розміщення сайту в Internet. Для цього необхідно розмістити всі файли web-сервісу на web-сервері, що надає такі послуги.

Рекламування. На цьому етапі передбачається взаємний обмін посиланнями з іншими сайтами, схожими за тематикою, включення адреси

сайту в листи, візитні картки, брошури, буклети і т. д., реєстрація сайту в популярних пошукових серверах;

Супровід. Після створення web-сервісу потрібно не забувати оновлювати інформацію, розширювати матеріал, покращувати дизайн, в іншому випадку велика ймовірність того, що потенційні відвідувачі web-сервісу не будуть до нього повертатися.

2.4 Технології розробки

При роботі з сайтом, веб-майстру необхідно знати кілька мов програмування і дві мови розмітки. Мовами розмітки є:

1. HTML (від англ. HyperText Markup Language - «мова гіпертекстової розмітки»);
2. CSS (англ. Cascading Style Sheets - каскадні таблиці стилів).

Мова HTML необхідна для роботи з інтерфейсом та його проектування. CSS необхідний для роботи з зовнішнім виглядом веб-документів. На відміну від мов розмітки, мови програмування можна використовувати різні. Вибір мови програмування, часто залежить від налаштувань сервера. Процес розробки сайту ділиться на три складові:

1. контент (content);
2. фронтенд (Front-End);
3. бекенд (Backend);

MVC - це шаблон проектування програми, при якому контролер служить виконуючою ланкою бекенда, модель служить виконуючою ланкою фронтенда, а вид служить виконуючою ланкою контенту (Рис. 4).

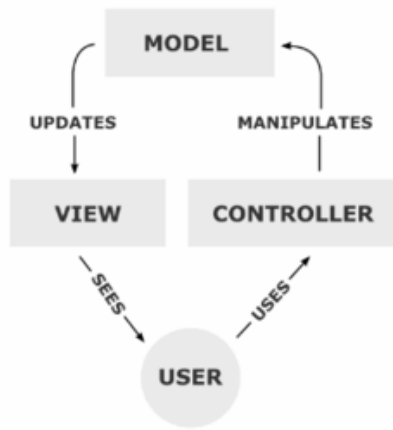


Рис. 1. Model-view-controller

Для того, щоб прискорити розробку CMS існують різні Фреймворки (англ. Framework - каркас, структура) Які вживають різні шаблони проектування. Фреймворки поділяють вид від уявлення і контролера, тим самим сильно полегшують завдання розробників. Фреймворки також бувають орієнтовані на бекенд і фронтенд. Це значить вони можуть полегшити завдання, як фронтент розробнику так і бекенда розробнику.

2.5 Модель взаємодії веб-серверу та сайту

Сайт повинен взаємодіяти з сервером за допомогою rest-запитів. Взаємодія з сервером зводиться до 4 операцій (необхідний і достатній мінімум):

- Отримання даних (JSON або XML)
- Додавання нових даних
- Модифікація даних
- Видалення даних

[11] Отримання даних не змінює стану сервера. Кожен тип операції користується своїм НТТР методом:

- GET-отримання.

- POST-додавання.
- PUT-Модифікація.
- DELETE-видалення.

Як приклад візьмемо стандартне API для списку користувачів. Ми запитуємо список користувачів, отримуємо інформацію про Користувача, додаємо нового користувача, змінюємо інформацію і видаляємо користувача. Нам знадобляться наступні запити:

- GET / api / users - отримати список користувачів.
- GET / api / users / 7- отримати інформацію про Користувача з ID=7.
- POST / api / users - додати нового користувача.
- PUT / api / users / 7 - змінити інформацію про Користувача з ID=7..
- DELETE / api / users / 7 - видалити користувача з ID=7.

При зверненні на будь-який з цих шляхів нам прийде відповідь у вигляді файлу json (пояснення нижче) з набором запитаних даних.

[7] JSON - це текстовий формат обміну даними, заснований на мультіпарадигмальній мові програмування. Його основне призначення полягає в зберіганні і передачі структурованого потоку інформації.

Синтаксис дуже простий, де key – це назва змінної, а value – значення:

```
{ 'key': 'value', ... }
```

Звертаючи увагу на все вище сказане, веб-сервіс (фонтенд) повинен бути розроблений на швидкій у дії бібліотеці. Завдання полягає у вибрі інструменту з переліку: React, Angular, Vue. Мы вибрали Vue, це JavaScript-

фреймворк з відкритим вихідним кодом для створення користувацьких інтерфейсів.

Веб-сервер (бэкенд) повинен бути розробленим на технології php+sql. Php це поширена мова програмування загального призначення з відкритим вихідним кодом. PHP спеціально сконструйований для веб-розробок і його код може впроваджуватися безпосередньо в HTML.

[14] Sql - декларативна мова програмування, що застосовується для створення, модифікації та управління даними в реляційній базі даних, керованої відповідною системою управління базами даних.

Розділ 3.

Розробка серверної складової веб-сервісу

3.1 Програмні засоби

Перед розробкою програми були сформульовані наступні вимоги до системи:

- **Масштабованість.** Можливість підвищувати пропускну здатність сервісу і додавати новий функціонал без необхідності сильно змінювати існуючий код.
- **Безпека.** В основі технологічного стека повинні лежати продукти, які мають вбудований захист. Навколо цих продуктів повинні бути живі спільноти розробників, що підтримують продукт, щоб нові «дірки» в безпеці могли бути в найкоротші сороки залатані.

З всіх характеристик мов програмування нас цікавить, чи мова інтерпретується або компілюється. За визначенням, компільовані мови програмування завдання виконують швидше, ніж інтерпретуються, і є багато рішень на базі компільовано-інтерпретованих мов: Java і C #.

З інтерпретуючими мовами ситуація простіше. Для того, щоб протестувати зміни достатньо всього лише оновити сторінку в браузері, тим більше, що цей процес можна автоматизувати, підключивши інструмент, який при зміні файлів розробки, сам оновлює сторінку. Також через відсутність досвіду роботи з бекенд фреймворками, необхідний був продукт з низьким порогом входу, доступною документацією і навчальним співтовариством.

Для розробки веб-сервера було вирішено використовувати платформу laravel і мова програмування PHP. Ця платформа з'явилася на ринку відносно давно, в IT-співтоваристві прижилася і полюбилася за головні переваги - строгу типізацію і легкість в проектування складних систем баз даних.

Як IDE була обрана програма Visual Studio Code від компанії Microsoft. Це обумовлено тим, що додаток на PHP легше інтегрується в Visual Studio Code за допомогою плагінів. В якості СУБД була обрана MS SQL, це бібліотека добре працює у зв'язці з платформою laravel і дозволяє повністю структурувати базу даних для веб-сервісу.

3.2 Архітектура додатку

Laravel визначають як full-stack фреймворк, оскільки він може працювати з усіма можливими частинами веб-додатку. До таких частин можна віднести: веб-сервіси, управління базами даних, генерація HTML, маршрутизація, кешування, аутентифікація, локалізація, управління групами користувачів, збірка css і js файлів, інкапсуляція ресурсів і багато іншого. Таким чином вертикально інтегроване середовище веб-розробки забезпечує кращий досвід для розробника. З Laravel розробник може взаємодіяти як через вбудовану утиліту командного рядка, яка управляє середовищем проекту Laravel, так і просто за допомогою настройки конфігураційних файлів вручну. Як правило обидва ці способи використовуються разом.

Laravel накладає деякі досить серйозні обмеження на структуру веб-додатків. Але, як не дивно, ці обмеження спрощують процес розробки. Laravel відрізняється від інших вертикально інтегрованих середовищ тим, що вона конфігурацій воліє конвенції. У той час як в деяких веб-фреймворк, написаних на Java, Python або PHP, часто потрібно редагування дуже великого обсягу файлів з конфігураціями, то для початку роботи з Laravel ніяких налаштувань практично не потрібно, хіба що крім декількох рядків на PHP.

Проекти під Laravel мають практично однакову файлову структуру - таку, в якій кожен функціональний блок має своє визначене місце. Це має спростити досвід роботи з цим фреймворком в цілому. Стандартна файлова

структура складається з великої кількості каталогів. Нижче наведено короткий виклад функцій кожного з цих файлів і папок:

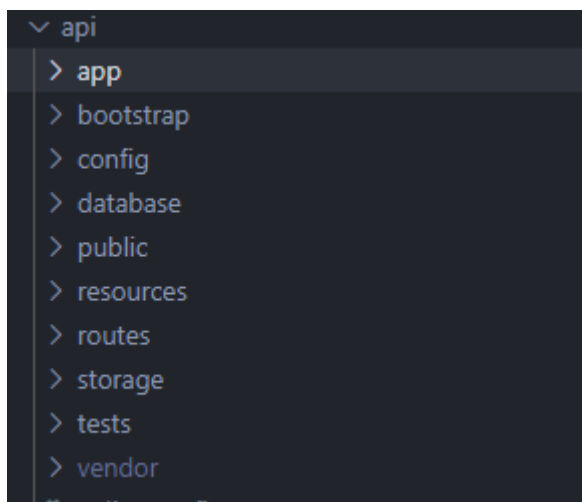


Рис. 1. - Файлова структура проекту

На рис. (1) ми бачимо перелік папок проекту. У кожній директорії своє призначення. У таблиці 1 нижче я покажу призначення декількох важливих:

/app	Тут знаходяться MVC та ORM моделі. Також конструктори запитів до SQL
/bootstrap	Містить в собі файли, які необхідні для запуску і роботи ядра Laravel.
/config	Місце знаходження всіх файлів конфігурацій, починаючи з бази даних і кешування, закінчуючи модулем відправки листів і файловою системою.

/database	Зберігаються файли міграцій і генерації баз даних з моделей.
/routes	Тут зберігаються файли для конфігурації модуля маршрутизації, а саме api.php
/vendor	Папка з бібліотеками
.env	Цей файл служить зв'язуючи вузлом між Laravel та БД.

Таблиця 1. Файлова структура

3.3 Model View Controller

[4] MVC - скорочення від «Model View Controller», представляє архітектуру, яку розробники приймають під час створення додатків. В архітектурі MVC ми розглядаємо структуру програми щодо того, як працює потік даних у середині.

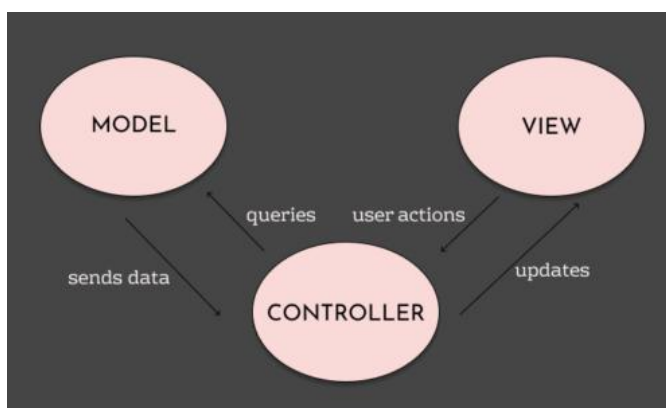


Рис. 2. - Model View Controller

Розглянемо Рис. (2). Модель - це представлення реального екземпляра або об'єкта в нашій базі коду. Перегляд представляє інтерфейс, за допомогою

якого користувач взаємодіє з нашим додатком. Коли користувач виконує дію, контролер обробляє дію та при необхідності оновлює Модель.

Якщо ви перейдете на веб-сайт, різні сторінки, які ви бачите, надаються шаром View. Коли ви натискаєте на посилання, щоб переглянути більше, шар Controller обробляє дії користувача. Це може передбачати отримання даних із джерела за допомогою рівня Model. Далі дані об'єднуються, розташовуються в рівні View і відображаються користувачеві.

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Game extends Model
{
    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'score', 'title', 'description', 'image'
    ];

    public function game()
    {
        return $this->belongsToMany('App\game', 'games_questions', 'game_id');
    }

    public function questions()
    {
        return $this->belongsToMany('App\Models\Question', 'games_questions', 'question_id');
    }
}
```

Рис. 2.1. Модель гри-тесту

На рис. (2.1) представлена головна модель, яка описує данні потрібні для взаємодії користувача та тесту у вигляді гри на стороні фронтенду. Змінна \$fillable зберігає у собі представлення полів у БД MySQL. Тобто модель зв'язує таблицю у БД та програму на PHP.

+ Параметры												
← T →												
			id	score	title	description	created_at	updated_at				
<input type="checkbox"/>		Изменить		Копировать		Удалить	3	10	game1	Descr for game 1	NULL	2021-03-29 11:23:13
<input type="checkbox"/>		Изменить		Копировать		Удалить	7	NULL	NULL	NULL	2021-03-29 10:54:09	2021-03-29 10:54:09
<input type="checkbox"/>		Изменить		Копировать		Удалить	8	NULL	NULL	NULL	2021-03-29 11:09:25	2021-03-29 11:09:25

↑ Отметить все С отмеченными: Изменить Копировать Удалить Экспорт

Рис. 2.2. Таблица гри-тесту

На рис. (2.2) ми бачимо представлення моделі Game у вигляді SQL таблиці. Усі поля зі змінної **\$fillable** перенеслися у вигляд таблиці. Щоб таблиця згенерувалася програмі на Laravel потрібно використовувати міграції таблиць.

У нашому проекті можна виділити декілька моделей та зав'язків між ними:

1. **Game та Question.** Зв'язуються за допомогою змінної **game_id** та **question_id** у суміжній таблиці **games_questions**.
2. **User та Role.** Зв'язуються за допомогою змінних **role_id** та **user_id** у суміжній таблиці **user_role**.

Часто під час розробки додатків та життєвого циклу до нього можуть бути додані нові вимоги та функціональність. У міру зміни запитів на програми можуть змінюватися вимоги до структури бази даних. Структуру бази даних можна змінювати вручну. Одна ненавмисна помилка може коштувати десятки годин і сотні зниклих користувачів. Редагувати міграції треба з розумом.

Звичайно, нам потрібен інструмент, який надає вам доступ до системи баз даних, пов'язаної з моделлю програми. Такий інструмент називається перехідним. Міграція - це в основному контроль версій бази даних. Усі великі реалізації SQL мають вбудовані системи передачі даних: Oracle, MySQL, MSSQL тощо. Laravel забезпечує інтерфейс для обробки передач для всіх механізмів баз даних.

Таким чином, підтримується метод вільного зчеплення: передачі пов'язані з логічною моделлю програми, і можна змінити механізми баз даних, не втрачаючи ці переходи.

Міграції здійснюються за допомогою вбудованого консольного інтерфейсу Laravel під назвою Artisan, а саме **php artisan make: Migration**, після чого параметри можна налаштовувати за допомогою аргументів.

Наприклад, `-create = $name` створює файл передачі для створення таблиці, де `$name` - це бажане ім'я таблиці.

```
class CreateGamesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('games', function (Blueprint $table) {
            $table->bigIncrements('id')->unsigned();
            $table->integer('score')->nullable();
            $table->text('title')->nullable();
            $table->text('description')->nullable();
            $table->timestamps();
        });
    }
}
```

Рис. 3. Міграція таблиці Games

На рис. (3) представлена міграція таблиці Games. В цьому файлі ми будемо вигляд таблиці, а саме генеруємо поля: `id`, `game`, `score`, `title`, `description` тощо. Після створення файлів міграцій у консолі проекту запускається команда `php artisan migrate`, після чого у SQL генеруються таблиці. Наприклад, таблиця міграції для Game моделі (див мал. 2.2) буде виглядати наступним чином.

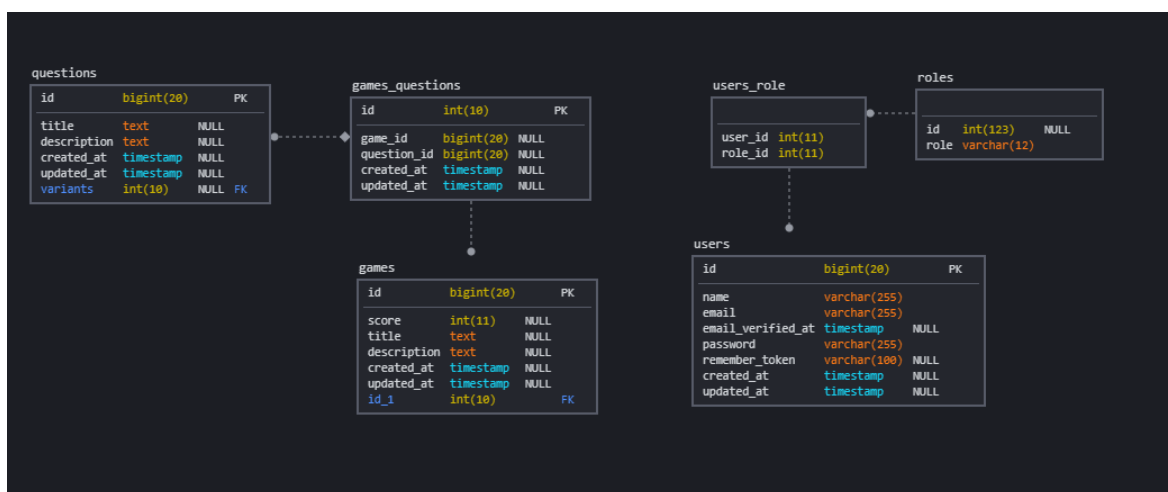


Рис. 3.1 Представлення в БД

У нашому серверному додатку база даних виглядає наступним чином, як на малюнку 3.1. Є дві головні таблиці `users_roles` і `games_questions`. Перша відповідає за впорядкування ігор та питань, а друга за користувачів і їх ролі. Дані таблиці були згенеровані за допомогою моделей розглянутих вище.

3.4 Контролери Арі та роутинг

У системі MVC буква «С» означає «Контролер». Він працює як листоноша між таблицями та моделями. Логіка запиту обробки даних у вигляді відповіді, організовується у файлах маршрутів – роутинг системі. Веб-сервіс шле запит на обробку даних за допомоги `POST`, `GET`, `DELETE` функцій. Бекенд в свою чергу обробляє запит за допомогою класів контролера.

```
class GameController extends BaseController
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $games = game::with(['game'])->with(['questions'])->get();

        return $this->sendResponse($games->toArray(), 'Games retrieved successfully.');
```

Рис. 3.2. Games контролер

Розглянемо рисунок (3.2). Як можемо побачити в коді представлена функція з назвою `index()`. Вона відповідає за обробку інформації надісланої з фронтенду. На її основі конструюється запит в SQL систему:

```
$games = game::with(['game'])->with(['questions'])->get();
```

Після цього відповідь записується у вигляді JSON та відправляється до місця призначення. Даний запис еквівалентна SQL запиту:

Для того щоб створити файл-контролер у системі треба відкрити командний рядок або термінал на основі операційної системи, яку ви використовуєте, і ввести **`php artisan make: driver <ім'я драйвера>`**. Це створює звичайний конструктор, коли ми передаємо аргумент. Якщо ви не

хочете створювати звичайний конструктор, ви можете просто проігнорувати аргумент. Створений конструктор відображається в дерикторії **app/Http/Controllers**. Створений драйвер можна викликати з **route.php**.

```
Route::apiResource('/games', 'App\Http\Controllers\Api\GamesController');

Route::get('/games', [GamesController::class, 'index']);
Route::get('/games/{id}', [GamesController::class, 'show']);
Route::post('/games', [GamesController::class, 'store']);
Route::put('/games/{id}', [GamesController::class, 'update']);
Route::delete('/games/{id}', [GamesController::class, 'destroy']);

Route::apiResource('/questions', 'App\Http\Controllers\Api\QuestionController');

Route::get('/questions', [QuestionController::class, 'index']);
Route::get('/questions/{id}', [QuestionController::class, 'show']);
Route::post('/questions', [QuestionController::class, 'store']);
Route::put('/questions/{id}', [QuestionController::class, 'update']);
Route::delete('/questions/{id}', [QuestionController::class, 'destroy']);
```

Рис. 3.3 Роутінг система

Щоб визначити маршрути, доступні в API треба зареєструйте контролери у файлі **routes/api.php**. Розглянемо рисунок (3.3). На ньому представлено роутінгова система. У списку знаходяться запити і контролери, які опрацьовують звернення по http. Модель поведінки дуже проста, веб-додаток шле запит по http за допомогою Get, Post, Put, Delete запитів, після чого сервер відповідає у вигляді масиву даних. Наприклад роут **\games** використовує **GamesController** і функцію **index**, яка була розглянута вище. Також тут чітко видно, як описуються шляхи для створення і видалення ігор-тестів та питань для них.

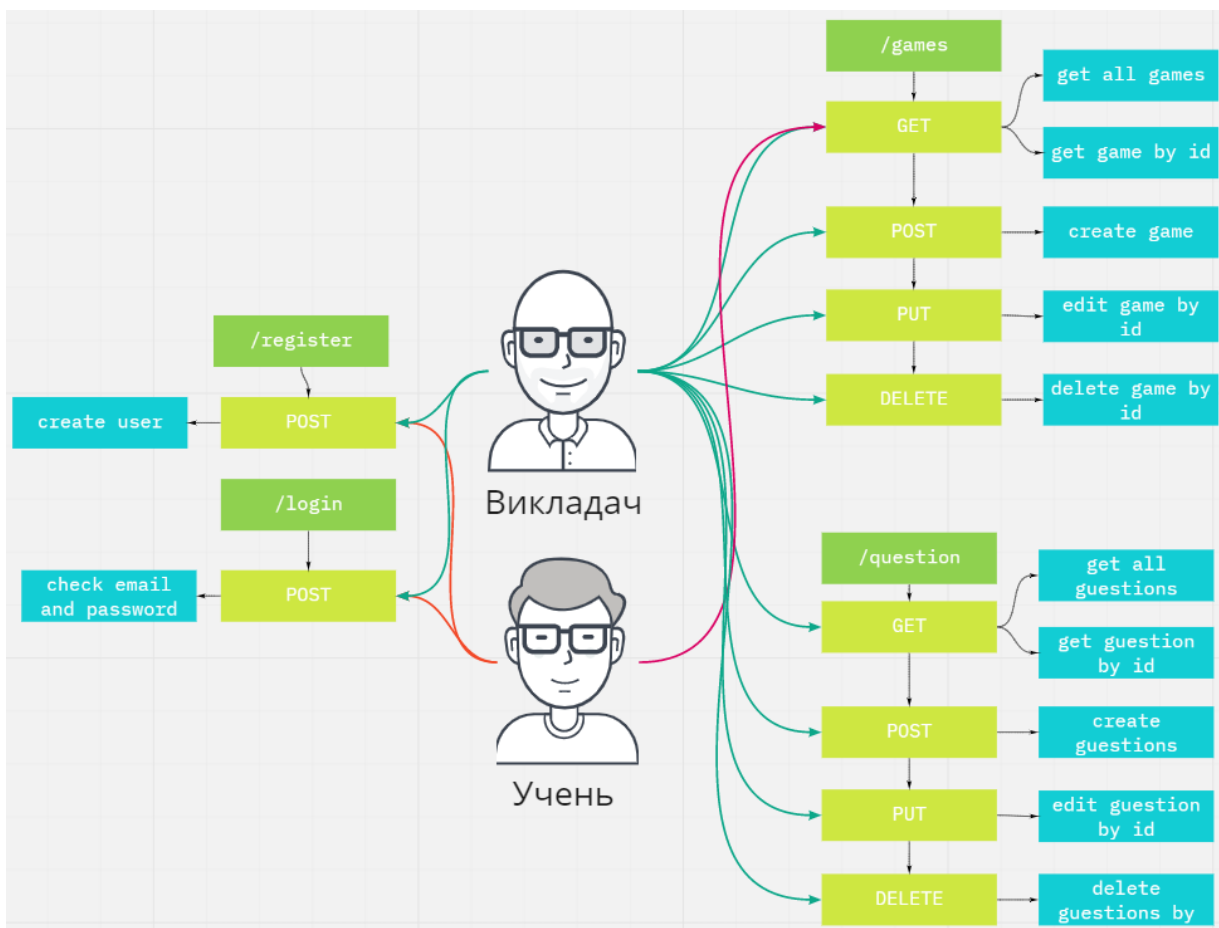


Рис. 3.4 UseCase діаграма

На рис. (3.4) представлена UseCase діаграма роутінгової системи. На ній чітко видно які дії може виконувати викладач та учень. Викладачу доступні наступні шляхи:

1. **/game**. Дії – POST (створити гру-тест), GET (перегляд ігор), PUT (редагування), DELETE (видалення).
2. **/questions**. Дії – POST (створити питання), GET (перегляд питань), PUT (редагування), DELETE (видалення).
3. **/register** та **/login** – реєстрації та вхід.

Учень в свою чергу має доступ до GET запиту **/games** та функціоналу реєстрації.

3.5 Взаємодія серверу та веб-додатку

Як було сказано неодноразово, веб-додаток отримує данні з веб-серверу за допомогою http запитів. У нашому додатку сервер поставляв основні ігри на головну сторінку, сторінку самої гри та праймє у зв'язці з конструктором тестів.

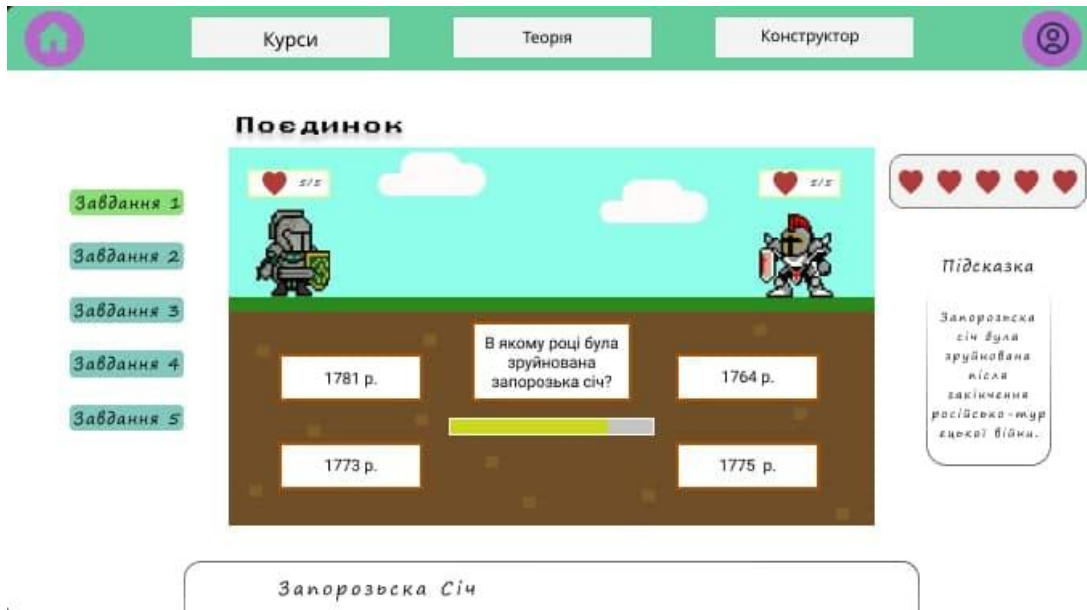


Рис. 3.5 Гра-тест 'Поєдинок'

Наприклад, розглянувши рис. (3.5) ми бачимо гру у якій треба відповідати на питання та при правильній відповіді лицар вступе у поєдинок, або лишається сердечка при неправильній. Веб-сервіс посілає запит на /game/{id}, де {id} – це номер гри у системі. При успішному отриманні даних виведеться вся інформація на екрані користувача, наприклад питання та відповіді. Таким чином веб-сервіс взаємодіє з сервером та приймає потрібні данні.

ВИСНОВКИ

У ході дипломної роботи була поставлена задача спроектувати і розробити серверну частину веб-додатку для навчання школярів начальних класів.

Цю мету я досяг шляхом вирішення наступних питань:

1. Аналіз теорії тестування.
2. Аналіз систем розробки.
3. Розглядання сучасних підходів та технологій у сфері бекенд розробки.
4. Складання діаграм прецедентів.
5. Розробка API системи.

Розроблений сервер відповідає всім перерахованим в тексті вимогам та готовий до експлуатації фронтенд частиною. Проект має перспективи подальшого розвитку. Таким чином завдання вирішені у повному обсязі, мета досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Брусов А. С., Тарасов С. О. Использование фреймворка Laravel 5. 0 для разработки web-приложений //Современные материалы, техника и технологии. – 2015. – №. 3 (3).
2. Гардейчик С. М., Шербаф А. И. PHP-фреймворк laravel с использованием архитектурной модели MVC //Перспективные направления развития отечественных информационных технологий. – 2017. – С. 133-135.
3. Грузин Н. А. СРАВНЕНИЕ ФРЕЙМВОРКОВ ДЛЯ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ: LARAVEL, DJANGO И FLASK //Modern Science. – 2020. – №. 2-1. – С. 359-364.
4. Таскынбаев А. Б. СОЗДАНИЕ ВЕБ-ПРИЛОЖЕНИЙ ПРИ ПОМОЩИ СРЕДЫ РАЗРАБОТКИ LARAVEL //СБОРНИК МАТЕРИАЛОВ X ВСЕРОССИЙСКОЙ, НАУЧНО-ПРАКТИЧЕСКОЙ КОНФЕРЕНЦИИ МОЛОДЫХ УЧЕНЫХ С МЕЖДУНАРОДНЫМ УЧАСТИЕМ" РОССИЯ МОЛОДАЯ". – 2018. – С. 31564.1-31564.3.
5. Тен Т. Л., Шинекенев Н. А., Когай Г. Д. ПОСТРОЕНИЕ REST API НА ОСНОВЕ ВЕБ-ФРЕЙМВОРКА LARAVEL //Перспективы развития строительного комплекса. – 2018. – №. 12. – С. 369-372.
6. Zhou W. et al. REST API design patterns for SDN northbound API //2014 28th international conference on advanced information networking and applications workshops. – IEEE, 2014. – С. 358-365.
7. Формат JSON, метод toJSON [Электронный ресурс] - Режим доступа: <https://learn.javascript.ru/json>
8. Работа с JSON [Электронный ресурс] - Режим доступа: <https://developer.mozilla.org/ru/docs/Learn/JavaScript/Объекты/JSON>
9. Создание REST API с аутентификацией [Электронный ресурс] - Режим доступа: <https://laravel.demiart.ru/create-rest-api-with-authentication/>

10. Руководства по Ролям и Правам в Laravel [Электронный ресурс] - Режим доступа: <https://laravel.demiart.ru/guide-to-roles-and-permissions/>
11. Как создать RESTful API Laravel [Электронный ресурс] - Режим доступа: <https://softobzor.com.ua/laravel-restful-api/>
12. Что такое MVC: рассказываем простыми словами [Электронный ресурс] - Режим доступа: <https://ru.hexlet.io/blog/posts/chto-takoe-mvc-rasskazyvaem-prostymi-slovami>
13. Концепция MVC для чайников [Электронный ресурс] - Режим доступа: <https://ruseller.com/lessons.php?id=666>
14. Модель проектирования MVC — что это? PHP и MVC [Электронный ресурс] - Режим доступа: <https://otus.ru/nest/post/1847/>
15. 5 методик обучения с использованием онлайн-тестов [Электронный ресурс] - Режим доступа: <https://www.surveymonkey.ru/mp/5-teaching-techniques-for-using-online-quizzes/>
16. Чопсиева Г. М., Шаповалова Л. И. Использование онлайн-тестов при обучении грамматической стороне иноязычной речи учащихся 9 классов //Информационные технологии в исследовательском пространстве разноструктурных языков. – 2017. – С. 148-150.
17. Маркова Г. А. ИСПОЛЬЗОВАНИЕ МАССОВЫХ ОТКРЫТЫХ ОНЛАЙН-КУРСОВ В ПРОЦЕССЕ ОБУЧЕНИЯ //Эпоха науки. – 2020. – №. 21.
18. Лысова Л. К. Онлайн-технологии в процессе обучения //Электронное обучение в непрерывном образовании 2018. – 2018. – С. 40-43.
19. Апальков В. Г. Использование современных средств обучения и интернет технологий в обучении иностранным языкам //Открытое образование. – 2013. – №. 6.
20. Строганов Б. Г. Обучение через WEB. – 2013.