МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**КУРСОВИЙ ПРОЕКТ**

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Web-програмування, ч.2»

на тему: **««Курси іноземних мов»**

Студента VI курсу групи ІПЗ-19-1

спеціальності 121 «Інженерія програмного забезпечення»

Држевецького Микити Олеговича

(прізвище, ім’я та по-батькові)

Керівник ст. викл. каф. ІПЗ Чижмотря О.Г.

Дата захисту: " \_\_\_ " \_\_\_\_\_\_\_\_\_\_\_\_ 203 р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. В. Кузьменко

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чижмотря О.Г.

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сергутін С.О.

(підпис) (прізвище та ініціали)

Житомир – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп’ютерних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень: бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Зав. кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2023р.

ЗАВДАННЯ

НА КУРСОВИЙ ПРОЕКТ СТУДЕНТУ

Држевецькому Микиті Олеговичу

1. Тема роботи: «курси іноземних мов»

керівник курсового проекту: старший викладач кафедри інженерії програмного забезпечення Чижмотря О.Г. .

1. Строк подання студентом: "\_\_"  2023р.
2. Вихідні дані до роботи: Розробити сайт з курсами іноземних мов.
3. Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
   * + 1. Технічне завдання
       2. Аналіз аналогічних розробок
       3. Алгоритми роботи програми

4. Опис роботи програми

5. Програмне дослідження

1. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)

Посилання на репозиторій:https://github.com/NikitaDrzhevetskyi/courses

1. Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посади консультанта | Підпис, дата | |
| завдання  видав | завдання прийняв |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання " " 2023 р.

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів курсового проекту | Строк виконання етапів проекту | Примітки |
| 1 | Постановка задачі |  |  |
| 2 | Пошук, огляд та аналіз аналогічних розробок |  |  |
| 3 | Формулювання технічного завдання |  |  |
| 4 | Опрацювання літературних джерел |  |  |
| 5 | Проектування структури |  |  |
| 6 | Написання програмного коду |  |  |
| 7 | Відлагодження |  |  |
| 8 | Написання пояснювальної записки |  |  |
| 9 | Захист |  |  |

**КАЛЕНДАРНИЙ ПЛАН**

**Студент** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Држевецький М.О

(підпис) (прізвище та ініціали)

**Керівник проекту** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чижмотря О.Г.

(підпис) (прізвище та ініціали)

**РЕФЕРАТ**

Завданням курсового проекту була розробка WEB-додатку інформаційної системи, що дозволяє користувачеві переглядати різноманітну інформацію, а адміністраторові створювати, та редагувати дані на сайті.

Пояснювальна записка до курсового проекту на тему «курси іноземних мов» складається з вступу, трьох розділів, списку викорисаної літератури та додатків.

Текстова частина викладена на 32 сторінках друкованого тексту.

Пояснювальна записка має 2 сторінки додатків. Список використаних джерел містить 10 найменувань і займає 1 сторінку. В роботі наведено 15 рисунків. Загальний обсяг роботи – 35 сторінки.

Ключові слова: ІНТЕРФЕЙС, КУРСИ, РОЛІ, ІНФОРМАЦІЙНА СИСТЕМА, БАЗА ДАНИХ, ІНФОРМАЦІЯ, КОРИСТУВАЧ, АДМІНІСТРАТОР.

**ЗМІСТ**

[ВСТУП **.**](#_Toc41527598)

[РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ 9](#_Toc41527599)

[1.1 Аналіз задачі, засобів та методів її вирішення 9](#_Toc41527600)

[1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту 12](#_Toc41527601)

[1.3 Технічне завдання на курсовий проект 15](#_Toc41527602)

[2.1 Проектування загального алгоритму роботи програми 17](#_Toc41527603)

[2.2 Розробка функціональних алгоритмів роботи програми](#_Toc41527604)

2.3 Розробка програмного забезпечення [РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ](#_Toc41527606) **[.](#_Toc41527606)**

[3.1 Опис роботи з програмним додатком **.**](#_Toc41527607)

[3.2 Тестування роботи програмного забезпечення **.**](#_Toc41527608)

[ВИСНОВКИ **.**](#_Toc41527609)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **О**](#_Toc41527610)

# *Додаток А*

# ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

Sql -Structured Query Language

PS – PhpStorm

БД – база даних

ІМ – іноземні мови

ІС – інформаційна система

# ВСТУП

Дана курсова робота відображає процес створення сайту за петерном MVC на тему «Управління курсами іноземних мов».

Зацікавленість до освіти дорослих зростає в усьому світі. Це пояснюється тим, що дорослі люди усвідомлюють потребу, або невідповідність між своїм професійним, соціальними і культурними рівнями. Здатність набувати додаткові навички, «освіжати» свої знання й розвивати прихований потенціал набуває все більшої значимості в XXI столітті.

Актуальним варіантом набуття нових знань та розвитку серед людей середнього віку є вивчення іноземних мов. Як показує практика: знання мови дає більше шансів знайти хорошу роботу чи просунутися в кар’єрі. Впевнені навички комунікації з представниками інших країн - є невід'ємною складовою професійної компетенції сучасного фахівця. Окрім цього, вивчення другої мови також може дати більші уявлення про інші культури. Люди зі знаннями ІМ більш впевнено подорожують світом.

Аналізи досліджень показують, що курси ІМ є невід’ємною частиною будь-яких освітніх процесів. В останні десятиліття спостерігається значне зростання кількості відвідувачів таких курсів. Зі зростанням кількості учнів – зростає кількість інформації (особиста інформація користувача, інформація про курси, оцінки тощо), яку потрібно зберігати. Виникає потреба у структуруванні і зберіганні великої кількості даних і, відповідно, в отримуванні оперативної відповіді навіть на дуже складні запити пов’язаних з ними.

**Актуальність теми -** обумовлена популярністю курсів іноземних мов, а також потребою у структуруванні великої кількості інформації.

**Об’єкт дослідження курсової роботи** – створення додатку задопомогою паттерну MVC, розробка БД за тематикою даного проекту.

**Предмет** **дослідження** - «Курси іноземних мов», реалізація основних можливостей, які мають бути присутні на типових представниках таких проектів.

**Метою роботи** – закріплення знань, отриманих при вивченні PHP, HTML, JavaScript, Sql, Bootstrap програмного продукту, демонстрація практичного застосування сайту “Іноземних мов”, побудова застосунку за допомогою архітектурного шаблону MVC

.

# РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

## 1.1 Аналіз задачі, засобів та методів її вирішення

Вимогами до веб-сайту будуть:

* інтуїтивно зрозумілий інтерфейс користувача;
* надійність: веб-сайт має працювати безперебійно;
* ефективність: додаток повинний мати мінімальні вимоги до апаратного забезпечення, для використання цього програмного засобу потрібен тільки браузер та доступ в Інтернет;
* адаптивність: можливість безпроблемного використання сайту на різних платформах – комп’ютери, планшети, мобільні пристрої
* модифікація: веб-сайт, в разі потрібності, має бути легко модифікований (додавання функціоналу).

Щодо функціональних вимог, програмний продукт «продаж курсів» повинен містити:

* купити курс;
* пошук за конкретними параметрами;
* перегляд данних про курс;
* спроектовану базу даних, яка ефективно зберігає інформацію;
* розділ веб-сайту для авторизації або реєстрації користувача;
* можливість зміни паролю;
* сторінку профілю користувача;
* можливість редагування, видалення курсів та категорій адміністратором.

Для ефективної роботи, зручності та безпеки, буде вводитися система аутентифікації. У будь-який час користувач повинен мати змогу увійти в систему під своїм обліковим записом. У разі його відсутності, створити новий.

Під час створення програмного продукту фігуруватиме серверна мова програмування PHP, розробка повинна проходити дотримуючись об’єктно-орієнтованого підходу та використовуючи патерн MVC – схема розділення даних додатку, користувацького інтерфейсу й управляючої логіки на три окремі компоненти: модель, представлення і контролер так, щоб була можливість модифікації кожного компонента незалежно від інших.

Дотримуючись даних правил і буде здійснюватиметься розроблення програми.

Створення графічних елементів здійснюватиметься за допомогою Bootsrap - безкоштовного набору інструментів з відкритим кодом, призначених для створення веб-сайтів та веб-додатків, які містять шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу.

Спираючись на досвід роботи, та сучасні тенденції, для реалізації інтерфейсу управління інформаційною системою, що розробляється, було обрано мову програмування PHP, тому доцільно було використати СУБД MySQL. Великою перевагою є те, що для роботи з MySQL не потрібно нічого додатково встановлювати та налаштовувати; все необхідне вже доступне разом зі стандартним постачанням PHP. Адміністрування БД буде відбуватися з використанням веб-інтерфейсу PhpMyAdmin.

Спеціальні вбудовані функції для роботи з MySQL дозволяють просто та ефективно працювати з цією СУБД (виконувати запити, обробляти помилки, читати та записувати дані).

Переваги MySQL:

* Відкритий вихідний код;
* Простота у використанні;
* Функціонал (включає в себе весь необхідний набір інструментів);
* Безпека (багато систем безпеки вже вбудовані і працюють по замовчуванню);
* Масштабованість;
* Швидкість;

Табл. 1

В першому розділі було визначено основне завдання, поставленно цілі та задачі. Крім цього було створено загальний план роботи і визначено потреби, які мають бути виконані.

## 1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту

На сьогоднішній день курси іноземних мов в Інтернеті набувають все більшої популярності, і вже існує значна кількість веб-сайтів, що пропонують такі курси. Однак, для досягнення успіху та задоволення потреб користувачів, важливо аналізувати існуюче програмне забезпечення з такою тематикою.

Розглянемо декілька популярних курсів іноземних мов:

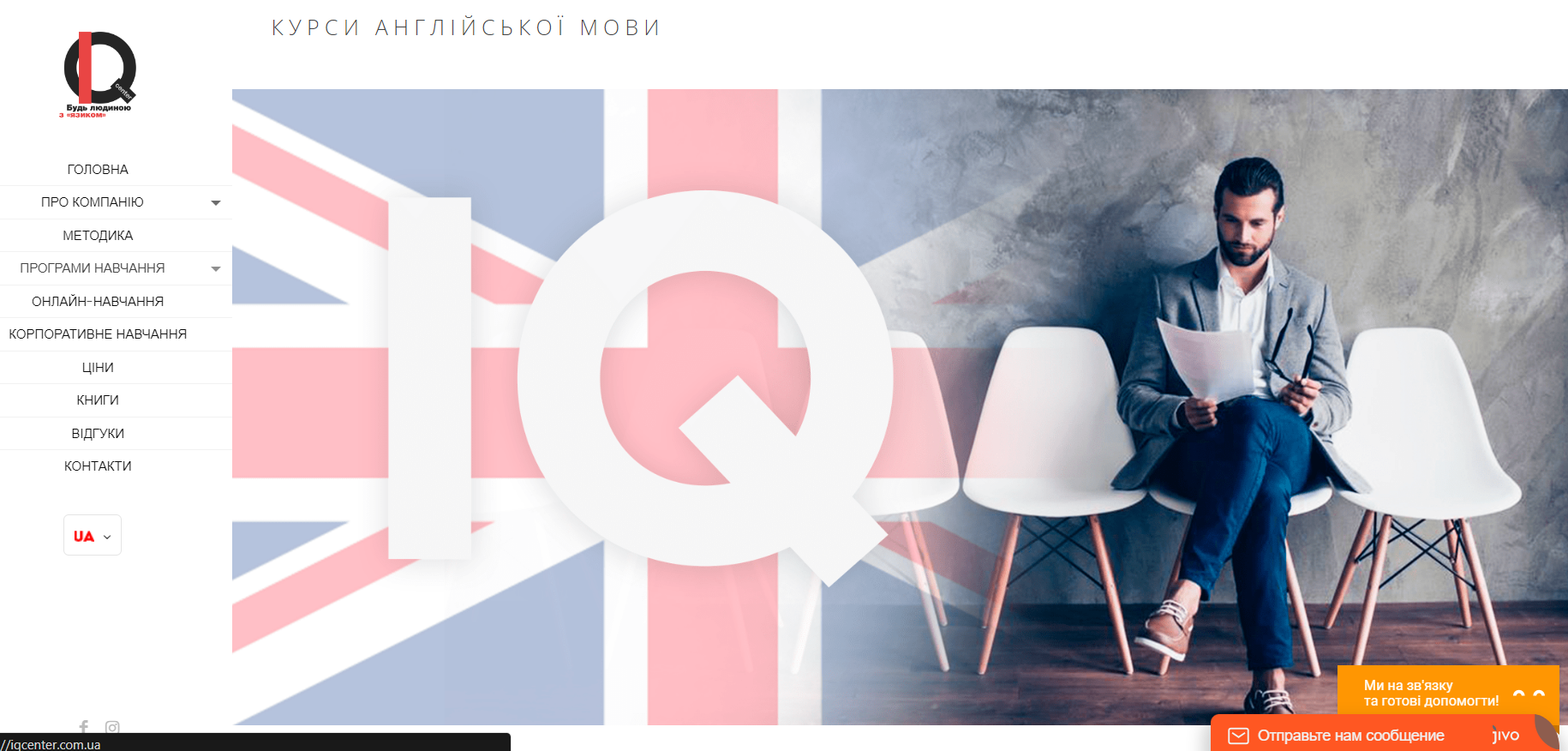


Рис. 1.1. Інтерфейс “ <https://iqcenter.com.ua/>”

Розділ: Аналіз існуючого програмного забезпечення для проекту "Курси іноземних мов"

Плюси:

1. Широкий вибір курсів: Сайт надає студентам можливість вибору великий вибір курсів, що охоплюють різні рівні вивчення та мови. Це дозволяє студентам вибрати саме ті курси, які найкраще відповідають їхнім потребам та метам.

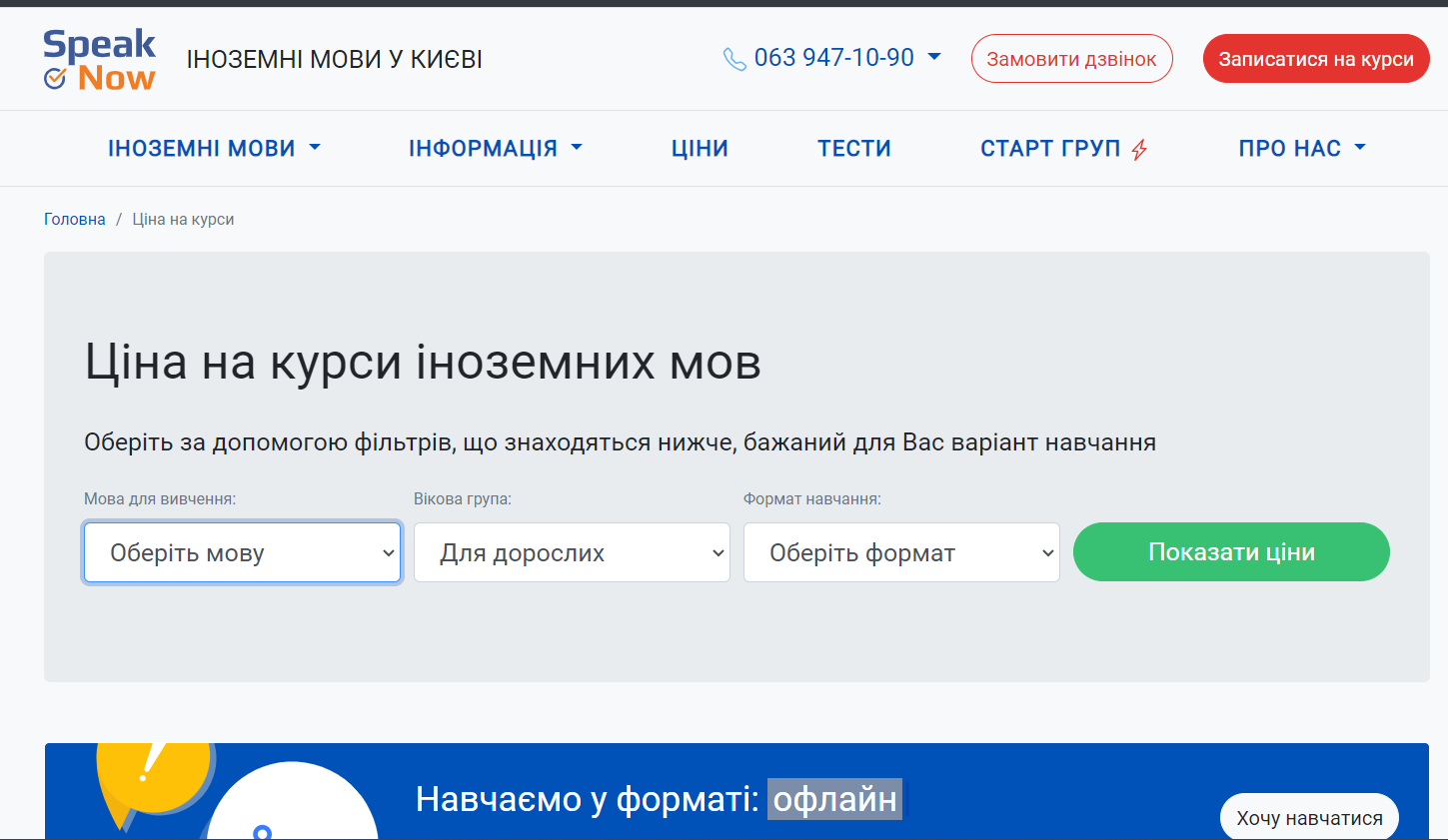
2. Чат-бот: Наявність чат-бота на сайті дозволяє студентам отримувати швидкі та зручні відповіді на свої запитання. Це забезпечує оперативну підтримку та допомогу, що покращує взаємодію між студентами та сайтом.

Мінуси:

1. Відсутність зручної навігації сайту: Сайт має проблему зі зручністю навігації, що може призвести до складнощів у знаходженні необхідної інформації. Брак логічної структури та зрозумілих посилань може вплинути на загальний досвід користувачів.

2. Відсутність адаптивності: На мобільних пристроях іконки та текст на сайті відображаються дуже великими, що ускладнює читання та взаємодію з сайтом. Відсутність адаптивного дизайну може знизити зручність використання сайту на різних пристроях.

3. Присутність російської мови на українському сайті.

Рис. 1.2. Інтерфейс “ https://speaknow.ua/prices ”

Плюси:

1. Інтуїтивно зрозумілий інтерфейс: Сайт має інтуїтивно зрозумілий та зручний інтерфейс, що дозволяє студентам легко орієнтуватися та знаходити необхідну інформацію. Це покращує загальний досвід користувачів та сприяє ефективному навчанню.

2. Наявність перевірки рівня знань: Сайт надає можливість студентам перевірити свій рівень знань іноземних мов за допомогою тестів або оцінки. Це дозволяє студентам зрозуміти свої сильні та слабкі сторони та зосередитися на відповідних аспектах навчання.

3. Можливість обрати різні формати навчання: Сайт пропонує різні формати навчання, такі як відеоуроки, аудіо матеріали, інтерактивні вправи тощо. Це дає студентам можливість вибрати той формат, який найкраще відповідає їхньому стилю навчання та сприяє більш ефективному засвоєнню матеріалу.

Мінуси:

1. Повільна швидкість завантаження сторінок: Сайт може мати проблеми з повільною швидкістю завантаження сторінок, що може призвести до незручностей та втрати часу для користувачів. Це може вплинути на їх загальний досвід та зацікавленість використання сайту.

**1.3 Технічне завдання на курсовий проекту**

Технічне завдання на курсовий проект "Курси іноземних мов"

Технічне завдання повинно містити наступну інформацію:

1. Опис функціональності проекту:

- Реєстрація користувачів: Користувачі зможуть створювати облікові записи, вводячи необхідну інформацію, таку як ім'я, електронна адреса, пароль тощо.

- Авторизація користувачів: Користувачі зможуть увійти до свого облікового запису за допомогою введення вірних облікових даних.

- Виведення списку курсів: Список доступних курсів іноземних мов буде відображатися користувачам, включаючи назву курсу, опис та інші деталі.

- Запис на курс: Користувачі зможуть записатися на обраний курс іноземної мови, надавши необхідну інформацію.

2. Архітектурні принципи та паттерн MVC:- Проект буде реалізований з використанням паттерна MVC (Model-View-Controller). Модель (Model) буде відповідати за роботу з базою даних, контролер (Controller) - за обробку запитів і керування логікою додатку, а представлення (View) - за відображення даних користувачам.

- Модульність: Проект повинен бути розділений на логічні модулі, щоб забезпечити чітку структуру та можливість розширення функціоналу в майбутньому.

3. Використання бази даних MySQL:

- Система повинна використовувати базу даних MySQL для зберігання інформації про користувачів, курси іноземних мов, записи на курси та журнали відвідуваності.

- Структура бази даних повинна бути розроблена таким чином, щоб забезпечити ефективність та зручний доступ до даних.

4. Інтерфейс користувача:

- Веб-інтерфейс: Користувачам буде доступний зручний веб-інтерфейс, де вони зможуть взаємодіяти з системою.

- Форми вводу: Система повинна мати відповідні форми вводу для реєстрації, авторизації, запису на курс та відміни запису.

# 

# РОЗДІЛ 2 ПРЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## **2.1 Проектування загального алгоритму роботи програми**

Після отримання достатнього об’єму інформації відносно потрібної галузі розробки переходимо до етапу проектування WEB-додатку. Це надасть нам можливість оцінити алгоритм роботи з точки зору користувача, а також адміністратора.

На цьому етапі розробки програмного забезпечення наведемо загальний алгоритм роботи WEB-додатку, що зобразимо на рисунку 2.1..

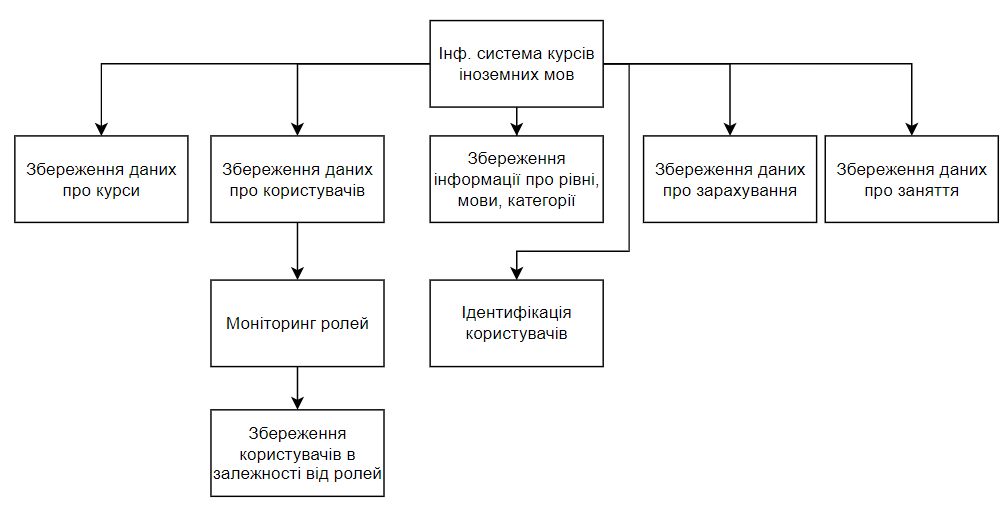


Рис.2.1 Схема загальної інформаційної системи



Рисунок 2.2. Загальний алгоритм роботи веб додатку

Загальний алгоритм для обробки запитів до веб-сайту буде наступний:

1. Користувач вводить адресу або натискає на посилання;
2. Викликається відповідний контролер;
3. Метод контролера звертається до потрібної моделі;
4. У моделі відбувається запити до бази даних для отримання або занесення інформації;
5. Модель повертає результати в метод контролера;
6. Контролер передає дані в представлення;
7. Представлення з відповідними даними передаються користувачеві, як відповідь на його запит.

Наступним кроком буде проектування бази даних. З попереднього підрозділу можна виокремити такі функціональні таблиці: для зберігання даних користувачів, для зберігання новин, для зберігання відомостей про оголошення та автомобіль. Також, для зберігання довідкових даних (список областей і відповідних міст України, список марок та моделей автомобілів), потрібні окремі таблиці. Зважаючи на це, усі дані будуть зберігатись у базі даних з наступними таблицями:

Основну діаграму бази даних з відображенням усіх зв’язків та полів наведено в Додатку А.

Для збереження ролей, доступних в інформаційній системі було створено табличку Roles. В данній таблиці знаходяться перелік основних ролей, які будуть надаватися користувачу при реєстрації. Структура таблиці наведена нижче:

Структура таблиці "Roles"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип | ПК | ЗК | Опис |
| id | int | + | - | Код ролі |
| name | varchar(255) | - | - | Назва ролі |
| description | Varchar(255) | - | - | Опис ролі |

Для збереження користувачів, було створено табличку Users. В данній таблиці знаходяться уся потрібна інформація для аутентифікації в системі, а саме логін (email) та зашифрований пароль. Хеш паролю зберігаються для того, аби гарантувати безпеку даних. Також ця таблиця зв’язана з таблицею Roles. Структура таблиці наведена нижче:

Структура таблиці "Users"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип | ПК | ЗК | Опис |
| id | int | + | - | Код |
| First\_name | varchar(255) | - | - | Ім’я |
| Last\_name | Varchar(255) | - | - | Прізвище |
| genger | Varchar(5) | - | - | Стать |
| date\_birth | date | - | - | Дата народження користувача |
| email | varchar(255) | - | - | Email користувача |
| telephone | varchar(30) | - | - | Телефон |
| password\_hash | text | - | - | Зашифрований пароль |
| role | int | - | + | Зв’язок з табличкою roles (поле roles.id) |
| create\_at | datetime | - | - | Дата створення |
| update\_at | datetime | - | - | Дата останнього оновлення |

Для збереження студентів, було створено табличку Students. В данній таблиці знаходяться усі потрібна інформація про студентів у системі в системі. Дана таблиця пов’язана з табличкою Users. Дані в цю таблицю вносятся завдяки роботі тригера. Структура таблиці наведена нижче:

Структура таблиці " Students "

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип | ПК | ЗК | Опис |
| id | int | + | - | Код |
| user\_id | int | - | + | Зв’язок з табличкою Users (поле Users.id) |
| is\_active | int | - | - | Опис ролі (0 – не активний, 1 – активний) |
| enrollmentDate | datetime | - | - | Дата зарахування |
| update\_at |  | - | - | Дата останнього оновлення |

Для збереження викладачів, було створено табличку Teachers. В данній таблиці знаходяться усі потрібна інформація про викладачів у системі. Дані з цієї таблиці використовуються для запису викладачів у таблицю заняття. Дана таблиця пов’язана з табличкою Users. Дані в цю таблицю вносятся завдяки роботі тригера. Структура таблиці наведена нижче:

Структура таблиці " Teachers"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип | ПК | ЗК | Опис |
| id | int | + | - | Код |
| user\_id | int | - | + | Зв’язок з табличкою Users (поле Users.id) |
| is\_active | int | - | - | Опис ролі (0 – не активний, 1 – активний) |
| hireDate | datetime | - | - | Дата зарахування |
| update\_at |  | - | - | Дата останнього оновлення |

Для збереження мов, було створено табличку languages. Структура таблиці наведена нижче:

Структура таблиці "languages"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип | ПК | ЗК | Опис |
| id | int | + | - | Код |
| name | varchar(255) | - | - | Назва мови |
| code | varchar(3) | - | - | Код мови |

Для збереження рівнів, було створено табличку levels. Дану табличку редагувати неможливо. Структура таблиці наведена нижче:

Структура таблиці " levels "

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип | ПК | ЗК | Опис |
| id | int | + | - | Код |
| name | varchar(255) | - | - | Назва рівня |
| code | varchar(3) | - | - | Код рівня |
| description | text | - | - | Опис ріня |

Для збереження категорії, було створено табличку categories. Структура таблиці наведена нижче:

Структура таблиці "categories"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип | ПК | ЗК | Опис |
| id | int | + | - | Код |
| name | varchar(255) | - | - | Назва категорії |
| description | text | - | - | Опис категорії |

Для збереження курсів, було створено табличку courses. В даній табличці зберігається інформація про усі курси в ІС. Дана таблиця пов’язана з таблицями categories, levels, languages. Структура таблиці наведена нижче:

Структура таблиці "courses"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип | ПК | ЗК | Опис |
| id | int | + | - | Код |
| name | varchar(255) | - | - | Назва курсу |
| language | text | - | - | Опис курсу |
| level | int | - | + | Рівень. Зв’язок з табличкою level (поле level.id) |
| category | int | - | + | Категорія. Зв’язок з табличкою category (поле category.id) |
| lessons | int | - | + | Заняття. Зв’язок з табличкою category (поле lessons.id) |
| price | float | - | - | Ціна |
| description | text | - | - | Опис |
| is\_active | varchar(1) | - | - | Активний курс (0 – ще не почався, 1 – триває, 2 – завершився, 3 – приховано) |
| start | date | - | - | Початок курсу |
| finish | date | - | - | Закінчення курсу |

Для збереження інформації про зарахування студентів на курс, оплату, зарахування/відрахування студента, було створено табличку enrollments. В даній табличці зберігається інформація про зарахування студента на курс. Дана таблиця пов’язана з таблицею enrollments. Крім того вона пов’язана з таблицею courses. Структура таблиці наведена нижче:

Структура таблиці " enrollments"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип | ПК | ЗК | Опис |
| id | int | + | - | Код |
| id\_student | int | - | - | Код студента. Зв’язок з табличкою category (поле Students.id) |
| payment | float | - | - | Код мови |
| date\_payment | datetime | - | - | Дата оплати |
| type\_payment | int | - | - | Тип оплати |
| id\_course | int | - | + | Код курсу. Зв’язок з табличкою courses (поле courses.id) |
| success | int | - | - | Активний курс (0 – відрахований, 1 – зарохований, 2 – чекає підтвердження) |
| created\_at | datetime | - | - | Дата створення |
| updated\_at | datetime | - | - | Дата останнього оновлення |

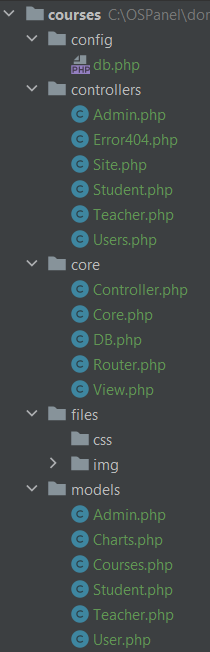
Для збереження інформації про заняття на курсі (розклад) було створено табличку lessons. Дана таблиця пов’язана з таблицею lessons. Крім того вона пов’язана з таблицею courses. Структура таблиці наведена нижче:

Структура таблиці " lessons"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип | ПК | ЗК | Опис |
| id | int | + | - | Код |
| name | varchar(255) | - | - | Назва мови |
| id\_course | int | - | + | Код курсу. Зв’язок з табличкою courses (поле courses.id) |
| id\_teacher | int | - | + | Код викладача. Зв’язок з табличкою courses (поле courses.id) |
| start | datetime | - | - | Кінець |
| finish | datetime | - | - | Початок |
| type | varchar(255) | - | - | Тип уроку |
| link | varchar(255) | - | - | Посилання |
| created\_at | datetime | - | - | Дата створення |
| updated\_at | datetime | - | - | Дата останнього оновлення |
| description | varchar(255) | - | - | опис |

На даному етапі відбувається розробка спроектованого Web-д. За допомогою середовища програмування Php Storm створюємо рішення та проєкт courses. У програмі міститься 47 файлів типу php (Рис2.4),

## 2.3 Розробка програмного забезпечення



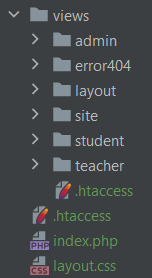


Рис2.3 Структура php

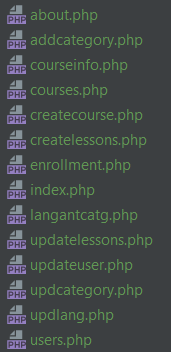


Рис2.4 admin

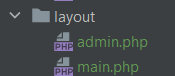


Рис2.5 layout

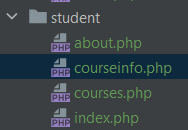


Рис2.6 student



Рис2.7 Структура css та .htaccess

Для реалізації основного інтерфейсу було розроблено веб-сайт з основною сторінкою і панеллю керування. Основна сторінка включає кнопки навігації (авторизація, реєстрація, курси, кабінет, вихід). Кнопки змінюються в залежності від входу/виходу користувача. На сторінці з курсами розміщено панель для сортування курсів (рис.5). Курси можна сортувати за ціною, категоріями, мовами, кількістю занять. Також, по мірі надходження інформації з БД формується пагінація.

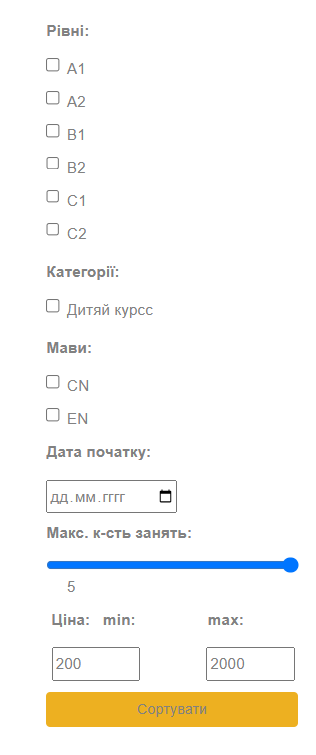


Рис2.8 Структура css та .htaccess

Курси виводять у вигляді блоків з основною інформацією (рис. 6.). Кнопка дозволяє зарахувати користувача на курс. Реєструватися на курси можуть лише студенти.

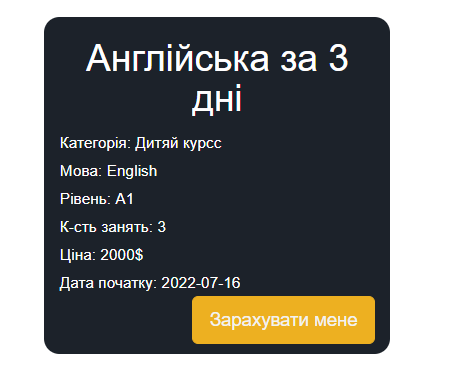


Рис.2.9 курс

Панель адміністратора включає в себе кнопки в залежності від ролі користувача. Наприклад для адміністратора відкрито можливість для редагування і оновлення будь яких даних – відповідно, кількість кнопок для переходу, порівняно зі студентом буде відрізнятися. (рис. 7)



Рис.2.10. Кнопки адміністратора.

На головній сторінці розміщено статистичні дані, які допомагають аналізувати інформацію з бд.  
Статистика формується для кожного користувача, чи то студент, чи адмін, за різними категоріями і параметрами (рис. 8).

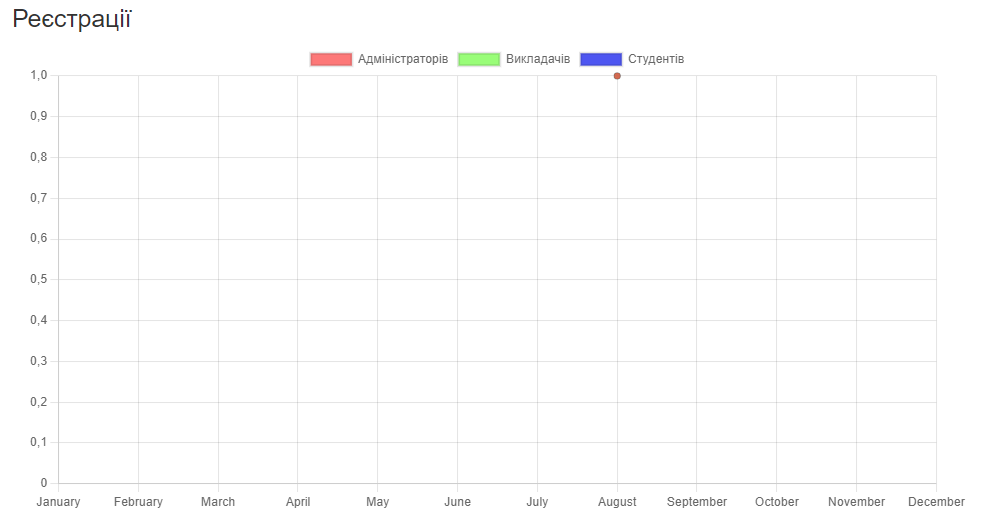


Рис.2.11. Приклад графіка аналізу даних для адміністратора.

Для створення і оновлення даних передбачені форми. Якщо це оновлення, то форма заповнюється відповідними даними з БД. Якщо створення – форма буде пустою. (рис. 9)

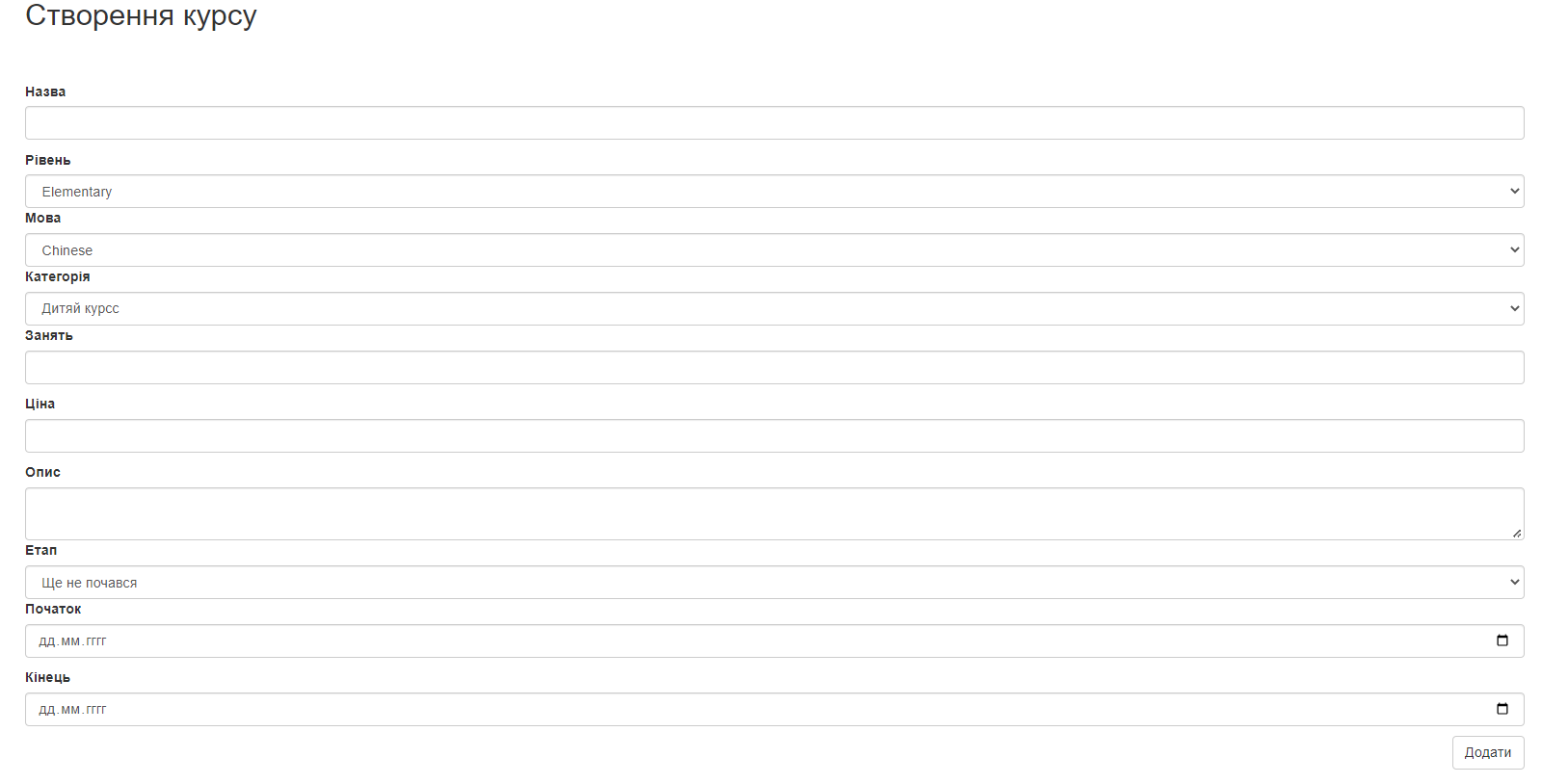


Рис. 2.12. Форма створення курсу.

Загалом данні відображаються у таблицях, в яких їх можна сортувати за всіма параметрами. Сортування забезпечують поля над виведенням даних (рис. 10).

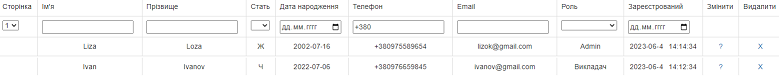


Рис. 2.13.. Таблиця з даними.

## *3.2 Реалізація операцій обробки даних в БД*

Підключення до бази даних відбувалося завдяки розширенню PDO (PHP Data Objects), яке забезпечує універсальний інтерфейс для доступу до різноманітних БД.

public function \_\_construct($host, $dbname, $user, $password)  
{  
 $this->pdo = new \PDO("mysql:host={$host};dbname={$dbname};charset=utf8",$user ,$password);  
}

Було створено клас DB, в якому було визначено основні операції для роботи з даними, а саме CRUD операції. Для того, аби уберегтися від SQL ін’єкцій було розроблено ряд перевірок для даних, що надходять від користувача, та особливий спосіб формування запитів.

Для вибору інформації із БД було створено метод Select, що приймав параметри (назву таблиці, поля, ліміт, сортування) тощо – і повертав запит із БД, якщо такий було знайдено. (Додаток Б. Select)

Для оновлення даних було розроблено метод Update, який приймав назву таблиці, поля і дозволяв оновлювати інформацію відповідного поля у таблиці, або всієї таблиці. Якщо оновлення проходило успішно – повертався параметр true, інакше – false. (Додаток Б. Update)

Для створення, додавання записів у БД було визначено метод Insert. Даний метод приймає в себе назву таблички і поля, які буде занесено у БД. За допомогою циклу ми формуємо запит і передаємо його у БД. Якщо додавання пройшло успішно – нам повернеться ідентифікатор цього запису. (Додаток Б. Insert)

Для видалення даних було визначено метод Delete. (Додаток Б. Delete).

Дані тригери регулювали ролі для користувачів у БД. А саме:

change\_role – при зміні ролі користувача, активність в протилежній таблиці (студент/викладач) змінюється на 0 або 1.

BEGIN

if(new.role like 3) THEN

IF(EXISTS(SELECT \* FROM students WHERE id = old.id)) THEN

UPDATE students SET is\_active = 1 WHERE id = old.id;

ELSE

INSERT INTO students (user\_id, is\_active, enrollmentDate, updated\_at) VALUES (new.id, 1, CURRENT\_TIMESTAMP(), CURRENT\_TIMESTAMP());

END IF;

IF(EXISTS(SELECT \* FROM teachers WHERE id = old.id)) THEN

UPDATE teachers SET is\_active = 0 WHERE id = old.id;

END IF;

ELSEIF(new.role like 2) THEN

IF(EXISTS(SELECT \* FROM teachers WHERE id = old.id)) THEN

UPDATE teachers SET is\_active = 1 WHERE id = old.id;

ELSE

INSERT INTO teachers (user\_id, is\_active, hireDate, updated\_at) VALUES (new.id, 1, CURRENT\_TIMESTAMP(), CURRENT\_TIMESTAMP());

END IF;

IF(EXISTS(SELECT \* FROM students WHERE id = old.id)) THEN

UPDATE students SET is\_active = 0 WHERE id = old.id;

END IF;

ELSE

IF(EXISTS(SELECT \* FROM students WHERE id = old.id)) THEN

UPDATE students SET is\_active = 0 WHERE id = old.id;

END IF;

IF(EXISTS(SELECT \* FROM teachers WHERE id = old.id)) THEN

UPDATE teachers SET is\_active = 0 WHERE id = old.id;

END IF;

END IF;

END  
 new\_user – при додаванні користувача – від автоматично додається в табличку з викладачами або студентами, залежно від ролі.

BEGIN

if(NEW.role = 3) THEN

INSERT INTO students (user\_id, is\_active, enrollmentDate, updated\_at) VALUES (NEW.id, 1, CURRENT\_TIMESTAMP(), CURRENT\_TIMESTAMP());

ELSEIF(NEW.role = 2) THEN

INSERT INTO teachers (user\_id, is\_active, hireDate, updated\_at) VALUES (NEW.id, 1, CURRENT\_TIMESTAMP(), CURRENT\_TIMESTAMP());

END IF;

END

## *3.3 Тестування*

Тестування роботи системи проводилося безліч разів, щоб виявити помилка та інші дифекти з метою їх усунення.

Тестування проводилося в декілька етапів:

1. Відкрити сайт, зареєструватися та авторизуватися.
2. Перейти у каталог, виконати сортування курсів.
3. Перевірка пагінації
4. Перехід в кабінет, редагування даних.
5. Авторизація як викладач.
6. Перегляд курсів, студентів, редагування занять.
7. Авторизація як адміністратор.
8. Додавання, оновлення видалення курсів, мов, користувачів тощо.
9. Перевірити «вихід» користувача з системи.

На першому етапі тестування було перевірено зручність використання та зрозумілість ІС, перевірено основні функції зчитування передавання, запису даних, налагоджено роботу з БД. Внесено зміни до БД.

На другому етапі було перевірено роботу всієї системи, проведено пошук багів та помилок. Перевірено роботу усіх кнопок, перевірено перевірку введення даних. Виявлення можливих перебоїв програми, виключеннь тощо.

На третьому етапі було перевірено відповідність усім вимогам відповідно до поставленої задачі, удосконалено інтерфейс.

## *Висновки до третього розділу*

В даному розділі було розглянуто роботу ІС, протестовано, перевірено взаємодію БД та інтерфейсного додатку, розглянуто основні можливості у роботі з сайтом, ознайомлено з основними виключеннями та похибками.

Помилки в роботі програми можуть виникати лише при неправильному її використанні або за непередбачуваних обставин.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

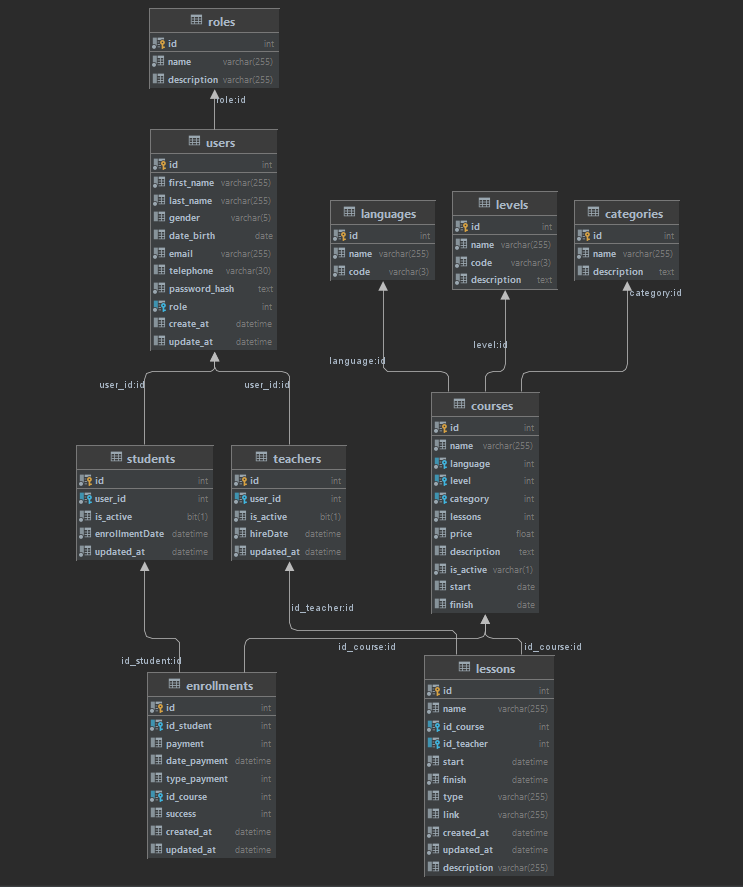
1. Офіційна документація PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/>.
2. Довідкова інформація [Електронний ресурс] – Режим доступу до ресурсу: https://metanit.com/php/tutorial/.
3. Порівняння БД [Електронний ресурс] – Режим доступу до ресурсу: [https://tproger.ru/translations/sqlite-mysql-postgresql-comparison/.](https://tproger.ru/translations/sqlite-mysql-postgresql-comparison/)
4. Довідкова інформація по MVC [Електронний ресурс] – Режим доступу до ресурсу:<https://ru.hexlet.io/blog/posts/chto-takoe-mvc-rasskazyvaem-prostymi-slovami>.
5. Реалізація MVC [Електронний ресурс] – Режим доступу до ресурсу:

<https://www.youtube.com/watch?v=3_UJ3nu_580&list=PLqKuDFs5Nd3L5hVkFH05vLYIgWeDTimDb>

1. Робота з CMS системою [Електронний ресурс] – Режим доступу до ресурсу:<https://freehost.com.ua/faq/wiki/cms-chto-eto-takoe/?gclid=Cj0KCQiAuP-OBhDqARIsAD4XHpcna9xuJfTlnDVSll7i-okR0tGWQ98KLPUTvQdJipu3P2HUGerklaQaAuioEALw_wcB>.
2. Порівняння існуючих БД [Електронний ресурс] <https://www.neftegas.info/upload/iblock/388/38851baaa72bd1449907e46f62776187.pdf>
3. Адміністрування БД [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/hub/db_admins/>
4. Робота з MySql [Електронний ресурс] – Режим доступу до ресурсу: <https://htmlacademy.ru/tutorial/php/mysql>
5. Довідкова інформація по Bootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://bootstrap-4.ru/docs/5.1/getting-started/introduction/>

ДОДАТКИ

## *Додаток А*

******

**Додаток Б**

**Select**

public function Select($table, $fields = '\*', $join = null, $where = null, $orderBy = null, $offset = null, $limit = null, $having = null, $group=null)  
{  
 $selectStr = self::*checkFieldsAndWhere*($where, $fields);  
  
 $sql = "SELECT {$selectStr['fields']} FROM {$table} {$join} {$selectStr['where']}";  
  
 //ORDER  
 if (is\_array($orderBy)) {  
 $orderByParts = [];  
 foreach ($orderBy as $key => $value) {  
 $orderByParts[] = "{$key} {$value}";  
 }  
 $sql .= ' ORDER BY ' . implode(', ', $orderByParts);  
 } else if (is\_string($orderBy)) {  
 $sql .= ' ORDER BY ' . $orderBy;  
 }  
  
 //LIMIT  
 if (!empty($limit)) {  
 if (!empty($offset)) {  
 $sql .= " LIMIT {$offset}, {$limit} ";  
 } else {  
 $sql .= " LIMIT {$limit} ";  
 }  
 }  
  
 if (!empty($having)) {  
 $sql .= " HAVING {$having} ";  
 }  
 if (!empty($group)) {  
 $sql .= " GROUP BY {$group} ";  
 }  
  
 $sth = $this->pdo->prepare($sql);  
 $sth->execute($selectStr['values']);  
 return $sth->fetchAll();  
}

**Insert**

public function Insert($table, $fields = null)  
{  
 if(is\_array($fields)){  
 $resultStr['fields'] = implode(', ', array\_keys($fields));  
 $values = [];  
 foreach ($fields as $key => $value) {  
 $values[] = "?";  
 }  
 $resultStr['values'] = implode(', ', array\_values($values));  
 }else {  
 $resultStr['fields'] = $fields;  
 }  
 $sql = "INSERT INTO {$table} ({$resultStr['fields']}) VALUES ({$resultStr['values']})";  
 $sth = $this->pdo->prepare($sql);  
 $sth->execute(array\_values($fields));  
 return $this->pdo->lastInsertId();  
}

**Update**

public function Update($table, $fields, $where = null)  
{  
 $resultArr = [];  
 if (is\_array($fields) and !empty($fields)) {  
 $fieldsParts = [];  
 foreach ($fields as $key => $value) {  
 $fieldsParts[] = "{$key} = ?";  
 $resultArr[] = $value;  
 }  
 $resultStr['fields'] = "SET " . implode(', ', $fieldsParts);  
 } else if (is\_string($fields) and strlen($fields) > 0) {  
 $resultStr['fields'] = $fields;  
 }  
 //$where  
 if (is\_array($where) and !empty($where)) {  
 $whereParts = [];  
 foreach ($where as $key => $value) {  
 $whereParts[] = "{$key} = ? ";  
 $resultArr[] = $value;  
 }  
 $resultStr['where'] = "WHERE " . implode(' AND ', $whereParts);  
 $resultStr['wherevalues'] = array\_values($where);  
 } else  
 if (is\_string($where) and strlen($where) > 0) {  
 $resultStr['where'] = "WHERE " . $where;  
 }  
  
 try {  
 $sql = "UPDATE {$table} {$resultStr['fields']} {$resultStr['where']}";  
 $sth = $this->pdo->prepare($sql);  
 $sth->execute($resultArr);  
  
 return true;  
 }catch (PDOException $e) {  
 return false;  
 }  
}

**Delete**

public function Delete($table, $where = null)  
{  
 $selectStr = self::*checkFieldsAndWhere*($where);  
  
 $sql = "DELETE FROM {$table} {$selectStr['where']}";  
  
 $sth = $this->pdo->prepare($sql);  
 if($where === null)  
 $sth->execute();  
 else  
 $sth->execute($selectStr['values']);  
  
}