Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

\_Навчально-науковий інститут інформаційних технологій та робототехніки

(повна назва факультету)

\_Кафедра комп’ютерних та інформаційних технологій і систем\_

(повна назва кафедри)

**Пояснювальна записка**

**до дипломного проєкту (роботи)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_магістра\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(освітньо-кваліфікаційний рівень)

на тему

\_Розроблення інтелектуальної інформаційної системи електронного навчання\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Виконав: студент \_6\_ курсу, групи \_601-ТН\_\_

спеціальності

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_122 Комп’ютерні науки\_\_\_\_\_\_\_\_\_\_\_\_\_

(шифр і назва напряму)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Хлопонін О. С.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_\_\_\_\_\_\_\_\_Двірна О. А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ .\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Полтава – 2022 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП’ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

**спеціальність 122 «Комп’ютерні науки»**

**на тему**

**«Розроблення інтелектуальної інформаційної системи електронного навчання»**

**Студента групи 601-ТН Хлопоніна Олега Сергійовича**

Керівник роботи

кандидат технічних наук,

доцент Двірна О.А.

Завідувач кафедри

кандидат технічних наук,

доцент Головко Г.В.

Полтава – 2022

**РЕФЕРАТ**

Кваліфікаційна робота магістра: 78с., 29 малюнків, 4 таблиці, 2 додатки, 27 джерел.

**Об’єкт дослідження:** функціонал інтелектуальної інформаційної системи електронного навчання.

**Мета роботи:** розроблення програмних модулів інтелектуальної автоматизованої системи управління навчальним процесом у школі, а саме функціоналу адміністратора та учня.

**Методи:** створення бази даних для системи інтелектуального електронного навчання, реалізація механізмів збору та аналізу статистичних даних за допомогою елементів штучного інтелекту, розробка користувацького інтерфейсу, застосування системи контролю версій Git в розробці програмних модулів, функціональне тестування програмного забезпечення методом «білого ящика».

**Ключові слова:** автоматизована інформаційна система, інтелектуальна інформаційна система, елементи штучного інтелекту, освітній інтернет-ресурс, електронне навчання, інтерфейс користувача, база даних, програмний код, модуль.

**ANNOTATION**

Qualification work of magister’s degree: 78p., 29 pictures, 4 tables, 2 appendices, 27 sources.

**Object of study:** functionality of the intelligent e-learning information system**.**

**The goal of the work:** development of software modules of the intelligent automated control system of the educational process in the school, the functionality of the administrator and the student.

**Methods:** creation of a database for an intelligent e-learning system, implementation of statistical data collection and analysis mechanisms using elements of artificial intelligence, development of a user interface, application of the Git version control system in the development of software modules, functional testing of the software product by the method of "white box".

**Keywords:** automated information system, intelligent educational Internet resource, elements of artificial intelligence, e-learning, user interface, database, program code, module.

ЗМІСТ

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ 6](#_Toc121956543)

[ВСТУП 7](#_Toc121956544)

[РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 9](#_Toc121956545)

[1.1 Огляд поняття та методології електронного навчання 9](#_Toc121956546)

[1.2 Аналіз основних вимог та акцентів для електронного навчального ресурсу 12](#_Toc121956547)

[1.3 Огляд наявних засобів та інструментарію електронного та дистанційного навчання 15](#_Toc121956548)

[1.4 Штучний інтелект в електронному навчанні 18](#_Toc121956549)

[РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ 22](#_Toc121956550)

[2.1 Визначення функцій системи 22](#_Toc121956551)

[2.2 Архітектура інформаційної системи 32](#_Toc121956552)

[2.3 Проектування бази даних 32](#_Toc121956553)

[РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ 36](#_Toc121956554)

[3.1. Вибір технологій 36](#_Toc121956555)

[3.2 Розробка бази даних 39](#_Toc121956556)

[3.3 Розробка користувацького інтерфейсу учня 46](#_Toc121956557)

[3.4 Розробка користувацького інтерфейсу адміністратора 54](#_Toc121956558)

[РОЗДІЛ 4 ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ 58](#_Toc121956559)

[4.1 Вибір способу тестування 58](#_Toc121956560)

[4.2 Тест план 58](#_Toc121956561)

[ВИСНОВОК 64](#_Toc121956562)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 65](#_Toc121956563)

[ДОДАТОК А ЛІСТИНГ КОДУ ДЛЯ РЕАЛІЗАЦІЇ ТЕСТУВАННЯ 68](#_Toc121956564)

[ДОДАТОК Б ЛІСТИНГ КОДУ ДЛЯ РЕАЛІЗАЦІЇ ПЕРЕГЛЯДУ НАВЧАЛЬНИХ МАТЕРІАЛІВ 75](#_Toc121956565)

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

**ІТ** – інформаційні технології.

**ІС** – інформаційна система.

**БД** – база даних.

**ПЗ** – програмне забезпечення.

**СУБД** – система управління базами даних.

**ЕОР** – електронно освітній ресурс.

**SQL** – Structured Query Language.

**HTTP** – HyperText Transfer Protocol.

**UML** – Unified Modeling Language.

**ISO** – International Organization for Standardization.

**SCORM** – Sharable Content Object Reference.

**AJAX** – Asynchronous JavaScript And XML.

**HTML** – HyperText Markup Language.

**XML** – Extensible Markup Language.

**PHP** – Hypertext Preprocessor.

**CSV** – Comma-Separated Values.

**CSS** – Cascading Style Sheets.

# ВСТУП

Ми живемо в час, коли інформаційно-комунікаційні технології проникають практично в усі сфери людської діяльності і стають звичайною частиною нашого життя. Звичайно, сфера освіти також реагує на цей розвиток. Спочатку комп’ютери стали предметом навчання, поступово вони також почали брати участь у підтримці освіти, а останнім часом, завдяки набутому необхідному технічному рівню, їм пропонують взяти на себе дуже фундаментальну роль у процесі навчання та здобуття знань. Так зване електронне навчання переживає перші роки свого існування, що принесло багато нових можливостей і викликів.

Сьогодні дуже важко оцінити, куди піде розвиток інтеграції інформаційно-комунікаційних технологій. У порівняно недавній час, коли було створено електронне навчання в тому вигляді, в якому ми його розуміємо сьогодні, воно сповнювало багатьох надії. Вона здавалася майже безмежною і очікувалося, що її впровадження в практику буде дуже швидким і викличе революційні зміни в усіх сферах освіти. Однак це очікування не виправдалося повністю.

Електронне навчання виявилося дуже специфічним підходом до навчання, якого недостатньо традиційним способам і методам навчання. Електронне навчання породило потребу в нових методах навчання, нових способах його оцінки та вимірювання якості та ефективності освіти. Після початкового великого буму настає період критичної оцінки та переоцінки підходу до електронного навчання, який може дати новий імпульс до відкриття нових можливостей застосування. Електронне навчання сьогодні є дуже стимулюючою темою, повною запитань без відповідей і непроторених шляхів, крім того, це дуже жива та динамічна тема.

При впровадженні будь-яких інновацій у педагогічну практику важливу роль відіграє користь цього кроку для освітньої галузі. Електронне навчання пропонує дійсно широкий спектр можливостей застосування та стосується майже всіх сфер освітньої практики. Водночас, однак, це несе з собою і ряд проблемних моментів, ігнорування яких може зіпсувати весь конкретний навчальний процес. Тому необхідно знати сильні та слабкі сторони електронного навчання та навчитися використовувати сильні сторони та належним чином усувати слабкі сторони. На певний напрямок розвитку такої інтеграції електронного навчання в практику вказують тенденції останніх років:

* дистанційне навчання става все більше поширене, уже незалишилось жодного навчального закладу, у якому би воно не застосовувалось;
* електронне навчання стає все більш оруєнтоване на дискретність у часі;
* навчання є орієнтоване не на швидке бездумне запам’ятовування матеріалу, а на довготривале його засвоєння та повторення.

Тема дипломної роботи присвячена розробці функціональної частини учня інтелектуальної інформаційної системи електронної освіти.

Результати виконаної роботи надалі можуть бути використанні загальноосвітніми навчальними закладами при веденні дистанційної або змішаної форми освіти, для створення звітів про успішність класів, забезпечення спілкування викладачів та учнів і перевірки завдань.

# РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## **1.1 Огляд поняття та методології електронного навчання**

Електроння навчання– такий різновид навчання, за яким учасники і організатори навчального процесу здійснюють переважно індивідуалізовану взаємодію як синхронно або асинхронно у часі, використовуючи в більшості електронні транспортні системи доставки засобів навчання та інших інформаційних об’єктів, комп’ютерні мережі Інтернет, медіа навчальні [1].

Підвищення попиту на освітні послуги та високий рівень розвитку інформаційно-комунікаційних технологій сприя швидкому розвитку електронного навчання у світі. Найбільшого поширення електронне навчання набуло в США та Канаді. Найрозвиненішими країнами Європи в сфері електронного навчання є Великобританія, Німеччина, Італія та Франція [1].

В сучасному інформаційному суспільстві електронний спросіб отримання знань є повсякденною складовою життя кожного школяра.

При складанні визначення терміну e-learning, тобто, по суті, електронної освіти, пропонуються два основних поняття. Перший з них заснований на американському середовищі, де електронне забезпечення навчання масово з'являється задовго до розвитку Інтернету, ще до масового поширення комп'ютерних технологій. В Америці вже давно використовується термін Технологічне навчання (Technology-based Learning), який включає все навчання з використанням таких технологій, як радіо, телебачення, супутникові приймачі, комп’ютери тощо.

В даний час загальне позначення E- використовується для такого навчання в американському середовищі навчання. Водночас навчання за допомогою інформаційно-комунікаційних технологій з використанням комп’ютерних мереж є лише його частиною і називається навчанням через Web. Навпаки, європейський підхід до цих понять є скоріше протилежним. Усе навчання, яке підтримується деякими технологіями, називається навчанням на основі технологій, а навчання через комп’ютерні мережі – електронним навчанням. Зокрема, Європейський Союз визначає електронне навчання як: «використання нових мультимедійних технологій та Інтернету для покращення якості освіти шляхом полегшення доступу до ресурсів і послуг, а також сприяння взаємній співпраці та обміну знаннями, навичками, інформацією. Парадоксально подібне визначення пропонує засновник американської індустрії електронного навчання Елліот Массі, який визначає його як: «інструмент, що використовує мережеві технології для створення, розповсюдження, вибору, адміністрування та постійного оновлення навчальних матеріалів».

Електронне навчання – є процес формальної та неформальної освітньої та навчальної діяльності, процесів, спільнот та подій, що забезпечуються з використанням електронних засобів масової інформації, таких як Інтернет, інтранет, телебачення, телефони, персональні комп’ютери. тощо. Проте більшість професійної громадськості погоджується, що сутність електронного навчання – це інформаційно-комунікаційні технології.

Електронне навчання використовує різні інструменти, а також має багато форм, його слід розуміти більше як «процес навчання з використанням інформаційно-комунікаційних технологій». І однією з його фундаментальних характеристик є використання великих або локальних комп’ютерних мереж – тому його можна розглядати як онлайн-освіту. Можливості його застосування дуже широкі, від підтримки очного навчання до повністю дистанційних форм навчання.

Електронна форма навчання може надати учням найшвидший доступ до нетрадиційних для класичної школи джерел інформації. Електронне навчання також надає змогу активізувати творчу діяльність школярів та підвищити результативність їх самостійної роботи, а вчителі, своєю чергою, можуть використовувати у своїй роботі абсолютно нові і беззаперечно ефективні методи навчання із застосуванням концептуального та математичного моделювання явищ та процесів [2].

Рівень якості електронної освіти нічим не менший ніж традиційної оскільки, для підготовки засобів такої форми освіти залучаються найсучасніші інформаційні технології та навчально-методичні матеріали, також передбачається спеціалізований контроль якості освіти та відповідність її всім стандартам [3].

Завдяки підключенню електронних систем спрощується відповідне адміністрування, автоматизація також входить в адміністрування результатів навчання та оцінювання навчання. Навчальний заклад отримує детальну інформацію про те, наскільки окремі студенти активні у навчанні, чи відповідають вони встановленим умовам навчання. Величезною перевагою також є значно нижчі вимоги до простору для аудиторій, лабораторій, навчальних кабінетів і бібліотек. Водночас суттєвої економії можна досягти й для студентів денної форми навчання за умови відповідної організації навчання, перейшовши на змішане навчання та скоротивши кількість очних годин лише до частки колишнього значення.

Для самих студентів електронне навчання також дає багато можливостей. Перш за все, це забезпечує високу ступінь індивідуалізації дослідження. Йдеться головним чином про те, що навчання більше не пов'язане зовнішніми впливами, такими як місце викладання, темп інших учнів, сувора послідовність дій і сам тип діяльності. Електронне навчання дає змогу навчатися навіть вдома, де студент може створити справді приємне та стимулююче середовище, зрозуміло, переваги для людей з обмеженими можливостями або людей із дуже віддалених місць.

Коли мова йде про асинхронну форму електронного навчання, то також відкриваються величезні можливості для індивідуального підходу кожного студента до навчання. Справа не тільки в тому, що студент може вибрати, в який час дня він буде вчитися, але також і в якому темпі або в багатьох випадках навіть який порядок тем він вибере. Крім того, електронне навчання дає змогу пропонувати студентам альтернативні засоби навчання (текст, звук, зображення, відео, анімація, симуляція), які дозволяють їм вибрати матеріал, який найкраще відповідає індивідуальному стилю навчання. Навчальний заклад, доступний 24 години на добу, 7 днів на тиждень, безумовно, є мрією кожного вчителя та учня.

Ще одна велика перевага електронних навчальних посібників полягає в їх легкому розповсюдженні та доступності. Той факт, що нова інформація може бути негайно врахована в навчальних опорах, призводить до високої актуальності навчального контенту, який може посилатися на останні відкриття, події та тенденції. Крім того, тепер самі студенти можуть дуже легко брати участь у створенні та доповненні змісту курсу, які можуть оцінювати навчальні матеріали, коментувати їх, доповнювати, додавати власні. Самі дискусії та інтерактивні електронні семінари стають окремим і часто дуже насиченим джерелом інформації.

## **1.2 Аналіз основних вимог та акцентів для електронного навчального ресурсу**

Освітня сфера використовує інформаційні технології з метою підвищити індивідуалізацію навчальної діяльності, забезпечити зв’язок між вчителем та учнем. Також для забезпечення високого рівня навчального процесу актуальною задачею є створення та запровадження систем, що забезпечують реалізацію різних форм навчальних комунікацій [4].

В даний час існує незліченна кількість формулювань визначень електронного навчання. Це завжди залежить від часу, в якому створюється визначення, і головним чином від контексту можливостей і динаміки розвитку мультимедійних та інформаційних технологій.

Якщо ми хочемо включити всі освітні можливості, пов’язані зі сферою електронного навчання, визначення має бути ширшим за своєю природою. Значення словосполучення e-learning можна сприймати як електронну освіту. Електронна освіта – це, з точки зору вчителя, реалізація освітнього процесу електронними засобами, наразі точніше засобами інформації та комунікації. Електронне навчання учня, у свою чергу, означає реалізацію процесу власного навчання цими засобами. Проте останніми роками проблема всієї концепції електронного навчання поглибилася, а інформаційна грамотність серед освітян і населення значно покращилася.

Таким чином, можна сказати, що електронне навчання включає не лише теорію, а й дослідження та будь-який навчальний процес, який відповідно до етичних принципів використовує інформаційні та комунікаційні технології, робота з даними в електронному вигляді. Спосіб їх використання та доступність навчальних матеріалів залежать головним чином від цілей і змісту освіти, характеру освітнього середовища, потреб і можливостей усіх учасників освітнього процесу.

Електронний освітній ресурс – такий освітній ресурс, що представлений в електронно-цифровій формі та містить програмні компоненти, освітній контент та метадані [5].

Додатковими матеріалами ресурсу можуть бути: хрестоматія, глосарій, анімаційні послідовності, довідники, анотована бібліографія, відеофрагменти, додаткова література, матеріали з суміжних курсів [6].

Головними вимоги до ЕОР є:

* наявність методичних рекомендацій по використанню ЕОР у роботі вчителя;
* послідовність та систематичність подання навчального матеріалу;
* надійність – учень повинний отримати бали, що відповідають показаним результатам;
* дружній інтерфейс;
* адаптивність – пристосованість програмного засобу до індивідуальних особливостей учня: учень самостійно може обрати собі темп навчання; або форму навчальних матеріалів;
* безперервність та повнота і безперервність циклу навчання з програмним засобом;
* відповідність санітарним нормам, максимальний час роботи за комп’ютером є 15-30 хвилин на уроці;
* використання сучасних методів обробки інформації;
* інтерактивне навчання – реалізація діалогу і зворотного зв'язку;
* розвиток інтелектуального потенціалу;
* відповідність законів України щодо авторського права [7].

Як показують проведені дослідження, при розробці зовнішнього вигляду електронних навчальних ресурсів, необхідно користуватися підходом керування мотивацією, що полягає в тому, аби кожен елемент дизайну інтерфейсу брав участь в активізації потреби в пізнанні й освіті. Данна навчальна тактика також вміщує планування та вибір видів навчальної діяльності, представлення та підбір навчальних матеріалів [8].

Основною вимогою до навчальних сайтів є те, що вони не мають відвертати увагу школярів на речі, що не відносяться до навчання, їх інтерфейс повинен бути простим та лаконічним, а найголовніше інтуїтивно зрозумілим для простого користувача [9].

Програмне забезпечення повинно бути створене таким чином, аби кожний користувач міг отримати до нього доступ, тобто такий програмний продукт повинен мати такий атрибут якості як адаптивність (адаптованим під різноманітні формати екрану комп’ютеру) [10].

Інформаційна система повинна бути доступною для звичайного користувача, аби викладачі та школярі довго не звикали до використання навчального ресурсу [11].

Одним з найголовніших завдань електронних навчальних ресурсів є надання викладачам та учням зручний функціонал для найкращої комунікації між собою [12].

## **1.3 Огляд наявних засобів та інструментарію електронного та дистанційного навчання**

Після огляду існуючих інформаційних систем електронного навчання, було обрано три для більш детального аналізу та описано основні їх функції, властивості та характеристики: Atutor [13], Docedo [14], iSpring Learn [15].

**ATutor** – веборієнтована система керування навчанням, вироблена в Канаді, яка написана мовою програмування РНР, має легкий у використанні інструментарій для вчителів, що дає змогу просто створювати та переносити навчальні матеріали, а також запускати власне онлайн навчання.

У системі визначено декілька типів користувачів, кожен з яких володіє своїми власними можливостями:

* вчителі;
* учні;
* адміністратори.

ATutor вміщує вбудований редактор курсів, який має вельми гнучкі налаштування та надає можливість викладачам персоналізувати навчальні курси.

Одним з найбільших недоліків ресурсу є те, що функціонал платформи залежить від встановлених модулів, так як старі модулі можуть не бути сумісними з новими версіями ATutor, а підтримуваних до сьогодні модулів наявно мало. Основним функціоналом системи є проведення курсів та тестування, збір статистики.

**Docebo** – модульна навчальна платформа, італійського виробництва, створена з використанням мови програмування РНР. Система є платною.

Модулі, що входять в систему:

1. Learn – модуль, який дозволяє проводити тестування, завантажувати до інформаційної системи навчальні матеріали, створювати програми навчання, додавати користувачів і призначати їм навчальні курси, формувати детальні звіти по матеріалах, завданнях та користувачах.

2. Coach & Share – навчальний модуль, що забезпечує простір для спілкування, надає можливість учням публікувати свої пости, оцінювати та коментувати пости інших, а також задавати викладачам пов’язанні з навчанням питання.

3. Extended Enterprise – модуль, який надає можливість налаштувати права доступу для різних організацій і користувачів.

4. Perform – модуль управління навичками, що дозволяє виявляти прогалини в навичках учнів і порекомендувати навчальні матеріали для їх усунення.

Система має вбудований редактор тестів та опитувань. Платформа має декілька типів питань, а також можливість встановивши час на одне питання або на цілий тест. Але система не надає можливості створювати тести з розгалуженням.

Навчальна платформа надає можливість вручну створювати користувачів. При необхідності додати доступ до порталу великій кількості людей за один раз, можна завантажити список у форматі CSV або XLS, а також користувачів можна додати за допомогою інтеграції CRM-системи SalesForce.

У системі Docebo кожен користувач може мати одну з наступних ролей, які у кожний момент можна змінити:

* адміністратор – має права на управління визначеною організацією, певними курсами чи конкретними групами;
* головний адміністратор – має права на переміщення користувачів з однієї організації в іншу, зміну налаштування, керування всіма матеріалами, видалення облікових записів, а також редагування користувачів, груп та організацій;
* звичайний користувач – має доступ лише до призначених тільки для нього навчальних матеріалів, та не може управляти іншими користувачами і своїм обліковим записом;

Платформа надає можливість об'єднувати учнів у групи за певним критерієм, наприклад, клас в якому навчається учень чи група по деякому предмету, до якої він належить. Курси також можуть бути призначенні для відразу цілої групи користувачів, або, у разі необхідності, сервіс дозволяє призначати курси для окремого учня.

У Docebo є можливість слідкувати за статистикою навчання, використовуючи вбудовані типи звітів, з можливістю їх налаштування.

Звіти, що були створенні, можуть бути переглянуті, відображенні чи прихованні для окремих користувачів, відредаговані, експортовані до форматів CSV, XLS або HTML [17].

**iSpring Learn** – онлайн платформа, що використовується для вебінарів, онлайн-тестування та дистанційного навчання. Інформаційна система є платною з наявністю пробного періоду для ознайомлення.

Освітня платформа iSpring Learn надає наступні можливості:

* створювати та редагувати навчальні курси;
* використовувати в навчальних курсах елементи гейміфікації;
* додавати та зберігати тести, презентації, курси, й інтерактивні діалоги, а також документи в різноманітних форматах презентації (.PPT), аудіо (.MP3), відео (.FLV) чи довідкові документи (.PDF, .DOC, .XLS);
* проводити та записувати вебінари;
* додавати користувачів до системи, об'єднувати їх у групи після чого надавати їм доступ до навчальних матеріалів;
* навчатися у офлайн режимі за допомогою мобільного додатку;
* створювати та переглядати статистичні звіти по заходах, учнях чи курсах [18].

Усі користувачі системи можуть бути об'єднані в групи, а також отримати одну з ролей в системі:

* адміністратор – володіє повним доступ до всіх налаштувань облікового запису, але не може підписуватись на курси;
* адміністратор організації – володіє повним доступом до керування користувачами та групами користувачів у організації, до якої належить;
* власник облікового запису – володіє повним контролем над створеним ним обліковим записом;
* користувач - має доступ лише до призначених йому курсів та історії своєї активності;
* автор – володіє контролем над створеним ним курсом.

Платформа iSpring Learn дає змогу призначити курси не лише окремим користувачам, а й групам. Кожен курс може бути призначений до якогось дедлайну або на певний термін [19].

## **1.4 Штучний інтелект в електронному навчанні**

Штучний інтелект є широким поняттям, що охоплює будь-які технології, які повторюють людське мислення і такі навички, як розуміння складної інформації, независле підбиття висновків та здатність вести зв’язний та осмислений діалог [20].

Машинне навчання – це підвид штучного інтелекту, що являє собою процес, у якому ПЗ навчається за тим самим принципом, що й людина. Протягом цього навчання програма аналізує великі обсяги даних та відшуковує закономірності задля класифікації інформації чи створення прогнозів. Циклічність даного алгоритму дій дає програмі змогу «вчитися», змінюючи свої наміри залежно від попередніх висновків [20].

Системи штучного інтелекту можуть обробляти набагато більше інформації, ніж може сприйняти будь-яка людина. А отже й швидко та набагато точніше розв’язувати поставленні перед ними завдання. Розробники навчального програмного забезпечення використовують переваги штучного інтелекту для створення програм, які можуть підлаштовуватися до індивідуальних особливостей кожного учня.

Штучний інтелект дає змогу створювати адаптивні освітні платформи, до яких використовуючи машинне навчання можна додати персоніфіковані навчальні плани, що будуть пришвидшувати опанування знань для кожного учня.

Штучний інтелект може сприяти персоналізованому підходу, він здатний надати той навчальний план, якого учень потребує в цю мить.

Незважаючи на великий потенціал, у використанні штучного інтелекту також існують деякі обмеження. Одним з них є те, що машинне навчання може найкраще працювати тільки тоді, коли було використано значний обсяг прикладів. А в такій ризиковій галузі, як освіта, може бути важко отримати багато негативних прикладів, оскільки вчитель не може собі дозволити часто помилятись.

Для навчання штучний інтелект повинні також використовуватися лише правильні дані, аби він дійшов коректних висновків. Результати навчання вийдуть хибними, якщо до загального числа даних потрапить неправдива чи неточна інформація.

Також використання подібних технологій може дати під загрозу захист персональних даних, тому при комп’ютерному навчанні слід дбати про відповідальне використання інформації, а не лише про її точність та якість.

У сьогоденні штучний інтелект має величезний вплив на стан освіти у світі, що лишає по собі великі наслідки.

Машинне навчання може в майбутньому розширити можливості учнів та вчителів, перетворити функціонування системи освіти та збільшити конкурентоспроможність навчальних закладів.

Середньо освітні навчальні заклади можуть використовувати штучний інтелект, для збільшення персоналізації навчання та подолати найбільше обмеження сучасної моделі освіти.

До основних переваг штучного інтелекту для учнів в освіті належить:

* віртуальні ментори;
* можливість отримати освіту у будь-який час;
* наявність різних варіантів в залежності від потреб учнів;

До основних переваг штучного інтелекту для вчителів навчальних закладів в освіті належить:

* можливість побачити слабкі місця;
* краще залучення;
* автоматичне створення навчальних програм;
* можливість знайти хорошого вчителя [21].

Використовувати штучний інтелект в освіті можна через:

* індивідуальне навчання;
* голосові помічники;
* розумний контент;
* глобальне навчання.

Штучний інтелект пропонує багато можливостей для обміну знаннями у всьому світі. Використовуючи рішення штучного інтелекту, студенти можуть вивчати різні курси та навчальні програми. Є багато платформ з інтерактивними навчальними матеріалами від кращих репетиторів [21].

Штучний інтелект також надає можливості студентам, які розмовляють різними мовами або мають проблеми із зором або слухом.

Наприклад, Presentation Translator – це рішення на основі штучного інтелекту, яке створює субтитри в режимі реального часу. Використовуючи штучний інтелект -розпізнавання мови, учні можуть чути чи читати рідною мовою [22].

Технологія штучного інтелекту вже успішно впроваджується і використовується в багатьох сферах нашого життя. Освіта вважається одним з найбільш консервативних секторів, де кожна інновація зустрічається вкрай прохолодно. Але можливості штучного інтелекту здатні розв’язувати непрості проблеми, з якими стикаються учні та викладачі і підвищити якість освіти. Уже з’явилися працюючі технології і методики, що активно застосовуються великими освітніми центрами [22].

Одним із найчастіше використовуваних методів штучного інтелекту є статистичний аналіз. Від допомагає зробити із великого об’єму неструктурованих даних змістовні висновки.

Статистичний аналіз це процес збирання та аналізу даних на основі яких формується деякий результат. Результат зазвичай представляє оброблені дані та виявлені в ньому деякі патерни та тенденції, на основі яких можна робити висновки та подальші рекомендації.

До основних переваг такого виду аналізу можна віднести:

* чітке відображення проміжків даних за деякий період часу, що допомагає точніше виявляти тенденції та на його основі робити прогнозування;
* допомагає виразніше відображати недоліки та переваги аналізованої області даних;
* допомагає приймати обґрунтовані та науково підкріплені рішення.

Зазвичай в процесі статистичного аналізу використовуються такі кроки:

* первинне визначення проблеми. На цьому етапі зазвичай формують цілі та очікуванні результати від аналізу;
* накопичення даних. Актуальні та точні дані є важливішим аспектом аналізу від яких залежить все дослідження. Зазвичай дані беруться з перевірених джерел або розробляються власні;
* аналіз даних. Найчастіше використовується дослідницький метод для розкриття даних за допомогою арифметичних операцій та графіків чи описів для візуального відображення результатів [23];
* підведення підсумків. За результатами аналізу розробляються висновки, рекомендації та ймовірне прогнозування йодо подальшого розвитку системи. Правильні висновки з аналізу призводять до ефективного корегування системи та її удосконалення.

Отже, статистичний аналіз даних організовує із різноманітних цифр єдине джерело інформації. При цьому дослідник має бути кваліфікованим для надання адекватної оцінки результатів та планування подальшої діяльності [24].

# РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ

## 

## **2.1 Визначення функцій системи**

Основним призначенням системи являється автоматизація навчального процесу у загальноосвітніх навчальних закладах. На рисунку 2.1 зображена розроблена концептуальна схема інтелектуальної інформаційної системи електронного навчання.



Рисунок 2.1 – Загальна структура системи

Система повинна імітувати навчальний процес, забезпечувати інтерактивну взаємодію учнів та вчителів, а також мати систему оцінки знань. В ході навчання учні повинні мати доступ до навчальних матеріалів в будь-який час та мати змогу самостійно з ним працювати.

Основні користувачі системи представлено в таблиці 2.1 у вигляді акторів та їх опис, як вони взаємодіють з системою та які завдання виконують у ході роботи.

Таблиця 2.1 – Актори ІС електронного навчання

|  |  |
| --- | --- |
| **Актор** | **Короткий опис** |
| *Адміністратор* | Адміністратор, що виконує основні налаштування системи, підтримує її, а також у разі виникнення будь-яких проблем допомагає іншим користувачам. |
| *Вчитель* | Користувач, основною задачею якого є створення сторінок курсів, наповнення їх навчальним матеріалом та завданнями для учнів, а також перевірка домашніх завдань, робіт та тестів. |
| *Учень* | Користувач, який використовує навчальні матеріали на сторінках курсів, здає домашні завдання, а також проходить тестування |

Користувачі системі мають такі функції:

* актор “*Адміністратор*” проводить налаштування системи та підтримує її функціонування, а також допомагає іншим акторам при виникненні проблем;
* актор “*Учень”* переглядає сторінки курсів, використовує навчальні матеріали, складає тести та надсилає виконані домашні завдання;
* актор “*Вчитель*” наповнює сторінки курсів навчальним матеріалом, створює домашні завдання та тести, переглядає результати тестування і виконані роботи та виставляє оцінки учням.

Адміністратор системи налаштовує її та надає права користувачам, але прав на редагування матеріалів курсів не має (рис. 2.2).

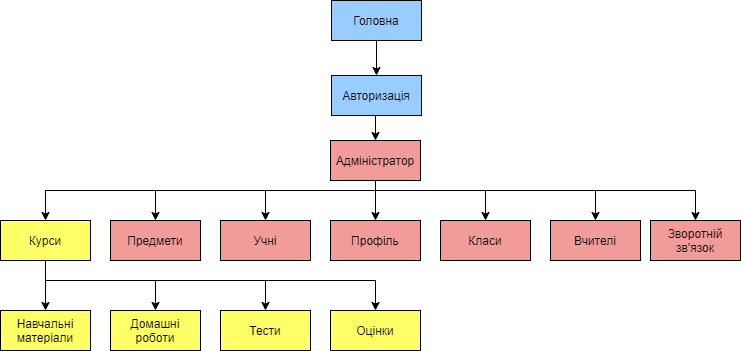


Рисунок 2.2 – Структура системи з погляду адміністратора

Вчитель має права на зміну доданих їм навчальних матеріалів. Учень може лише переглядати сторінки курсів, але не редагувати їх (рис. 2.3).

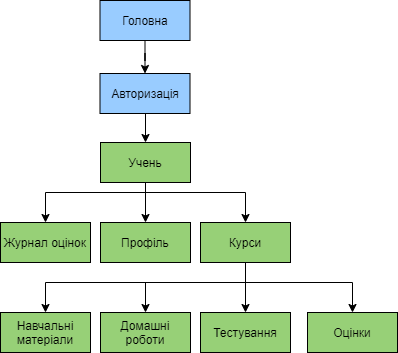


Рисунок 2.3 – Структура з погляду учня

У таблиці 2.2 наведено загальні варіанти використання системи, які повинні бути виконані в модельованій системі для актора “*Учнем”*.

Таблиця 2.2 – Сценарії ІС електронного навчання

|  |  |  |
| --- | --- | --- |
| **Основний актор** | **Найменування** | **Формулювання** |
| *Вчитель, Учень* | Перегляд навчальних матеріалів | Дозволяє *Вчителю* або *Учню* переглядати навчальні матеріали в системі. |
| *Учень* | Складення тесту | Дозволяє *Учню* складати тести в системі. |
| *Учень* | Виставлення оцінки за тест | Система автоматично підраховує оцінку за складений тест. |
| *Вчитель, Учень* | Перегляд результатів тестування | Дозволяє *Учню* переглянути результати складеного ним тесту. Дозволяє *Вчителю* переглядати результати тестування всіх учнів. |
| *Учень* | Керування домашнім завданням | Дозволяє *Учню* керувати домашніми завданнями в системі, а саме переглядати, редагувати та додавати нові. |
| *Учень* | Додавання домашнього завдання | Дозволяє *Учню* додавати домашнє завдання на сайт. |
| *Учень* | Редагування навчальних матеріалів | Дозволяє *Учню* редагувати додані ним на сайт домашні завдання. |
| *Вчитель, Учень* | Перегляд домашнього завдання | Дозволяє *Учню* переглянути додане ним домашнє завдання. Дозволяє *Вчителю* переглядати домашнє завдання його учнів. |
| *Учень* | Перегляд оцінок | Дозволяє *Учню* переглянути свої оцінки. |

На Use Case діаграмі (рис. 2.4) відображено зв’язки між акторами системи та різними варіантами її використання.

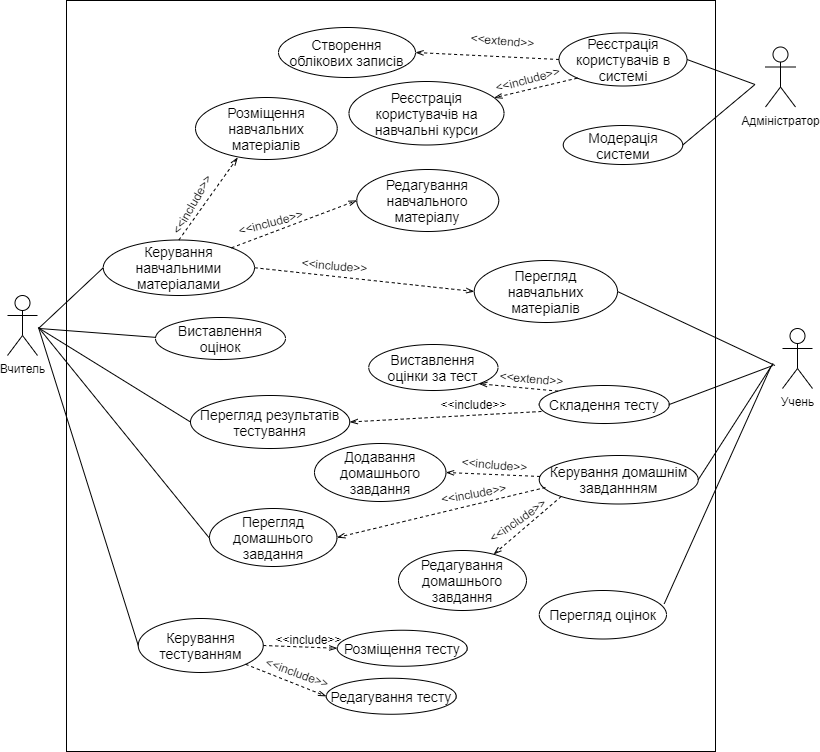


Рисунок 2.4 – Use Case діаграма системи

Більш наочно логіку сценаріїв використання системи учнем для розміщення домашнього завдання відображено на рисунках 2.5 та 2.6.

Ініціює цей сценарій актор «Учень». А беруть участь у цій взаємодії обєкти:

* клас «Student» (Учень);
* клас «HomeworkOptions» (Параметри роботи із домашнім завданням), який відповідає за вибір дії із домашнім завданням в поточному прецеденті;
* клас «AddNewHomework» (Додавання нового домашнього завдання), який за додавання домашнього завдання в систему;
* клас «Homework» (Домашнє завдання);
* клас «Subject» (Предмет).

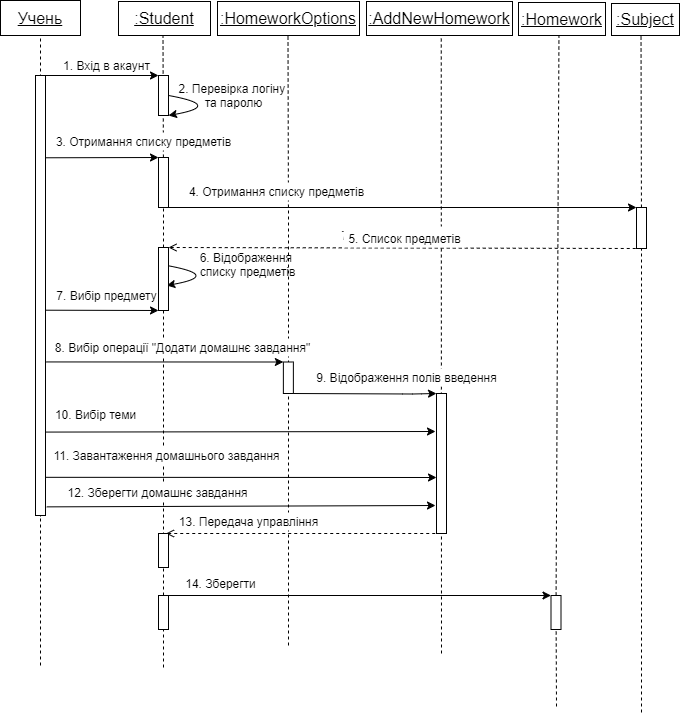


Рисунок 2.5 – Діаграма послідовностей

В ході виконання даного сценарію «Учень» входить в свій профіль у системі, а система відображає всі курси до яких він підключений. Актор обирає потрібний йому предмет та вибирає операцію «Додати домашнє завдання». Далі обирає тему та вносить матеріал в систему, або додає файл, або вводить текстову відповідь. На фінальному етапі учень відправляє свою відповідь, а система зберігає його.

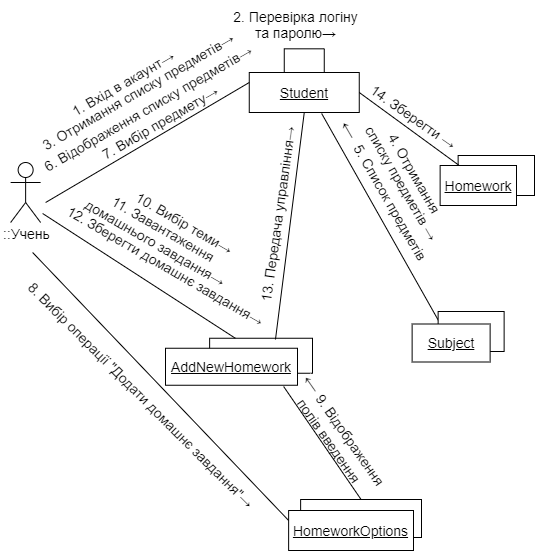


Рисунок 2.6– Діаграма кооперації

Вхідні сигнали та імпульси можуть змінювати стан системи як зображено на рисунку 2.7.



Рисунок 2.7 – Діаграма станів для користувача системи – Адміністратор

При перевірці системи адміністратором вона має стан «Перевірена система», якщо при цьому були виявлені одна чи декілька помилок, то система переходить у стан «Непрацездатна», після чого робота системи призупиняється та отримує стан «Тимчасово не працездатна». Якщо всі помилки були виправлені та система пройшла додаткові випробування вона переходить у стан «Виправлена» та вводиться в робочий стан.

У разі якщо при першій перевірці помилок не було знайдено, то система одразу отримує статус «Працездатна» та перевірка завершуються, а система переходить у кінцевий стан.

Процес додавання виконаного домашнього завдання учнем представлено на діаграмі (рис. 2.8).

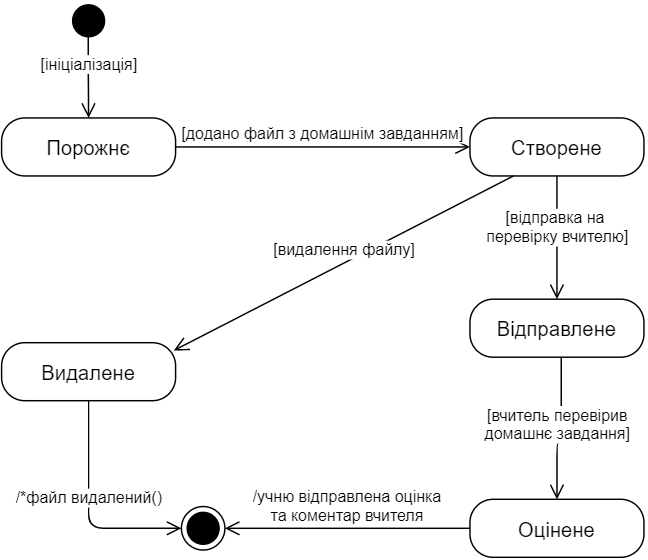


Рисунок 2.8 – Діаграма станів для користувача системи – Учень

Одразу після створення вчителем домашнього завдання воно має стан «Порожнє». Після події додавання відповіді у вигляді файлу система переходить в стан «Створене». Після цього вона може при видаленні файлу отримати стан «Видалене» або при відправленні відповіді учнем та збереженні файлу у системі «Відправлене». Після стану «Видалене» система переходить в кінцевий стан. Зі стану «Відправлене» після настання події перевірки домашнього завдання вчителем система отримує стан «Оцінене». Одразу після цього учень отримує свою оцінку та коментар вчителя, а домашнє завдання переходить у свій кінцевий стан.

Мовою UML представлено процес додавання виконаної домашньої роботи у вигляді діаграми поведінки моделі інформаційної системи (рис. 2.9).

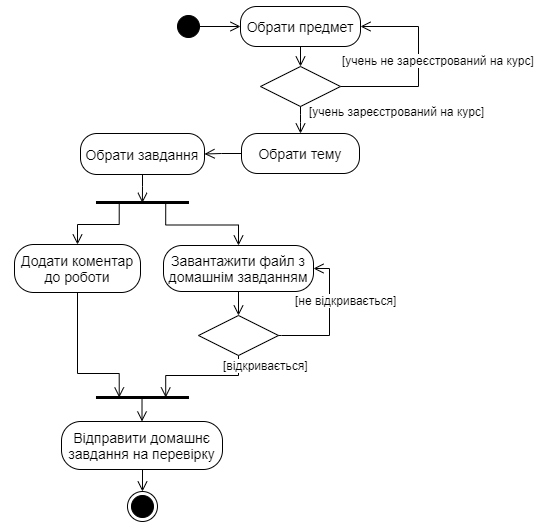


Рисунок 2.9 – Діаграма діяльності

Зі списку учень вибирає потрібний предмет, знаходить тему та відкриває завдання, далі пише коментар та додає файл до системи. Потім учень відправляє роботу вчителю на перевірку, система зберігає його відповідь та користувач завершує роботу з системою.

Якщо учень не має доступу до предмету, то система повертає його назад до вибору предметів зі списку.

## **2.2 Архітектура інформаційної системи**

Користувач використовує систему шляхом управління інтерфейсом вебсторінок інформаційної системи електронного навчання. При завантаженні сторінки браузер відправляє запит на web-сервер та отримує відповідь у вигляді користувацького інтерфейсу. Сервер формує сторінку базуючись на наявній логікі та взаємодії з базою даних. В подальшому користувач шляхом взаємодії з інтерфейсом може відправляти запити для доступу до даних, їх модифікації або видалення (рис. 2.10).

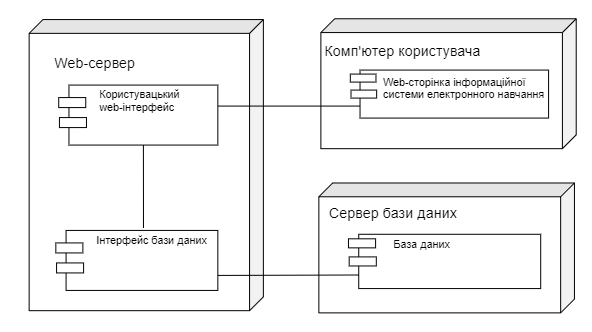


Рисунок 2.10 – Діаграма розгортання інформаційної системи

## **2.3 Проектування бази даних**

Система повинна зберігати облікові записи вчителів та учнів, перелік навчальних класів, списки предметів, містити навчальні матеріали, а також зберігати результати виконання завдань та проходження тестів учнями.

Структура бази даних базується на основі наступних правил:

* в одному класі може бути багато учнів, але учень може навчатися лише в одному класі;
* кожен учень може проходити декілька тестів та виконувати декілька завдань, але лише один результат проходження або виконання може належати учню;
* дані успішності можуть належати лише одному предмету та з одного класу, але для кожного предмету та класу можуть бути декілька видів даних успішності;
* кожен предмет може мати багато результатів тестування, але результат проходження тесту повинен належати лише одному учню, аналогічна логіка застосовується до домашніх робіт.

Головними сутностями системи є:

* «Вчитель», в якій знаходиться інформація про вчителів системи та їх загальну інформацію;
* «Учень», в якій знаходиться інформація про учнів та їх облікові записи;
* «Адміністратор», в якій знаходиться інформація про адміністратора ресурсу;
* «Тема», в якій знаходиться інформація про теми предмету на сторінці курсу;
* «Файл», в якій знаходиться інформація про файли які знаходяться на сторінках курсів;
* «Запис», в якій знаходиться інформація про текстові записи сторінок курсів;
* «Завдання», в якій знаходиться інформація про розміщені в системі завдання для учнів;
* «Клас», в якій знаходиться інформація про класи;
* «Предмет», в якій знаходиться інформація про предмети;
* «Тест», в якій знаходиться інформація про тести;
* «Питання», в якій знаходиться інформація про питання до тесту;
* «Відповідь», в якій знаходиться інформація про відповіді на запитання до тесту;
* «Виконане домашнє завдання», в якій знаходиться інформація про виконані роботи учнем;
* «Оцінка», в якій знаходиться інформація загальні оцінки учнів по предметам;
* «Результат тесту», в якій знаходиться інформація з результатами тестування учнів;
* «Сповіщення», в якій знаходиться інформація про сповіщення для користувачів.

За навединими вище сутностями та правилами створено схему бази даних за допомогою СУБД MySQL та вебінтерфейсу phpMyAdmin, яка містить19 таблиць, зображену на рисунку 2.11.

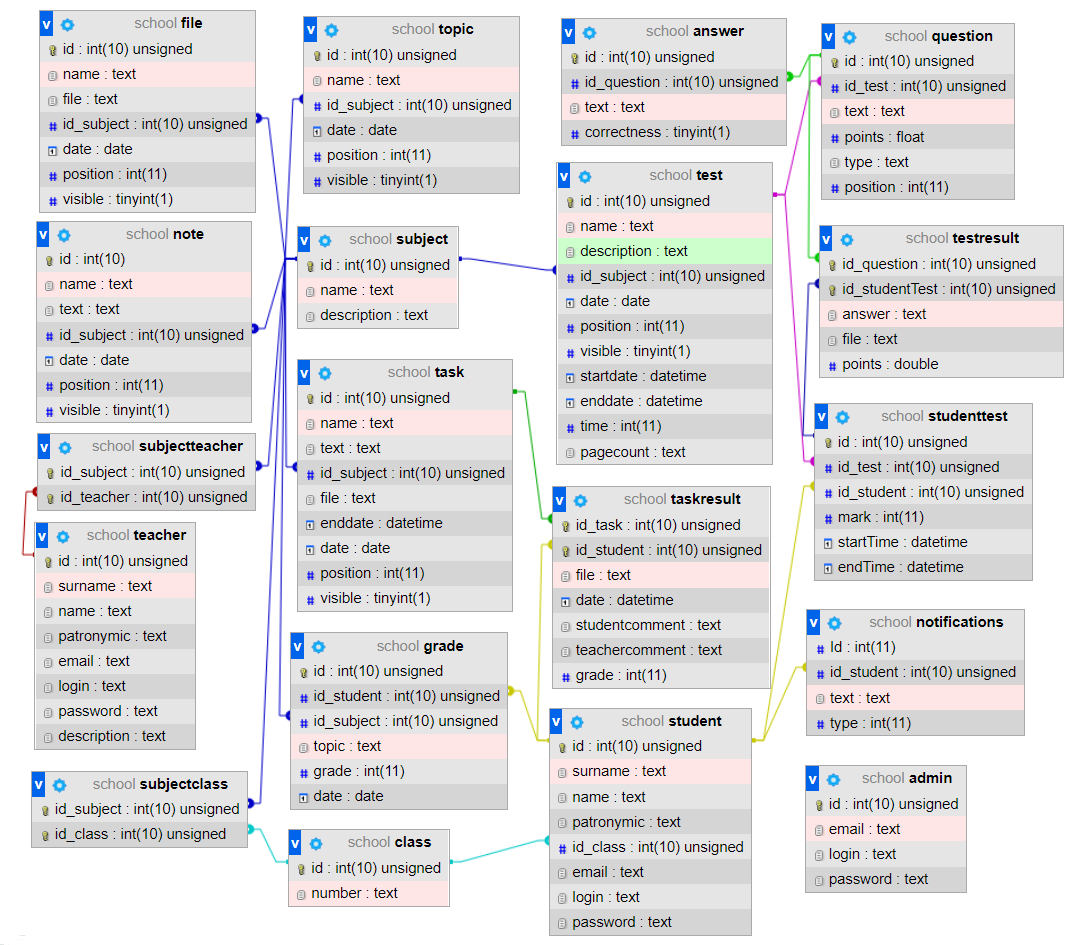


Рисунок 2.11 – Схема даних ІС

# РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ

## **3.1. Вибір технологій**

Після аналізу всіх наявних інструментів для розробки програмного забезпечення, для розробки інформаційної системи електронного навчання було вибрано серверну мову програмування PHP, основною задачею якої є надання безперервного та ефективного зв’язку серверу з інформаційної системи і базою даних.

Написаний на PHP код має виконання два обов’язкові завдання:

* можливість інтерактивності та динаміки, що забезпечує інтегрована в html php-частина;
* відображення інформації у коректному зовнішньому вигляду, що забезпечує html-частина [25].

Програмний код написаний на PHP й створені за його допомогою програмні продукти є ефективними, гнучкими, багатофункціональними та легкими у використанні, а також зручними в обслуговування, редагуванні та адмініструванні.

Головними перевагами PHP можна виділити:

* наявність широкого спектру можливостей та багатозадачності, ця мова програмування може бути використана для створення різних додатків,
* зручність для роботи з електронною поштою, pdf-документами, базами даних, редиректом або заголовками;
* простота мови в освоєння та простий для зрозуміння синтаксис;
* платформонезалежність та висока сумісність, тобто код, що був написаний за допомогою РНР буде працювати однаково добре на різних платформах;
* велика швидкість роботи мови та загальна продуктивність ресурсів;
* написання та редагування коду може бути здійснене за допомогою будь-якого текстового або html-редактору;
* велика гнучкість до потреб розробника;
* функціональність та місткість, РНР є мовою програмування, що вбудовується в код, програму на цій мові можна складати окремо від розробки вебсторінки, без прив'язки, після чого поєднати;

Мова програмування PHP уже багато років залишається на вершині популярності, незважаючи на велику кількість інших мов для написання різних вебдодатків, сервісів, сайтів тощо. За допомогою PHP можна досить легко створити різноманітні програми або динамічні сайти. Крім того, ця серверна мова програмування чудово інтегрується з майже усіма платформами для розробки сайтів та операційними системами.

При створенні програмного продукту також було використано технологію JavaScriptразом з Ajax для управління динамічним вмістом сторінок вебдодатку. Дана технологія допомогла уникнути перезавантаження сторінок після редагування її вмісту.

AJAX є набір інструментів веброзробки, що дають змогу інформаційним системам працювати в асинхронному режиму – обробляти декілька запити до сервера у фоновому режимі. JavaScript, і XML працюють асинхронно в AJAX. В результаті вебдодаток, який користується AJAX, може отримувати та відправляти дані на сервер без необхідності завантаження знову всього вмісту веб сторінки.

Технологія AJAX дозволяє користувацькому інтерфейсу та обміну даними працювати одночасно, не перериваючи один одного, а також значно полегшує роботу багатозадачності, за допомогою цієї технології може бути забезпечена одночасна робота декількох операцій [26].

У роботи з базою даних перевагу надано реляційній системі управління базами даних MySQL та інтерфейс для нього phpMyAdmin.

Головною перевагою баз даних MySQL є те, що код використовуваний MySQL, є відкритим, що дає змогу додавати новий функціонал без великих перешкод.

Також великим плюсом MySQL є її дуже швидка робота. Високу швидкість роботи бази даних забезпечує реалізована в ній багатопоточність, але не зважаючи на це, обрана СУБД є цілком безпечною.

Web-панель управління базами даних MyAdmin написана мовою програмування PHP. На сьогоднішній день являється найбільш актуальною панеллю для керування MySQL базами даних, нею користуються практично всі Hosting провайдери.

Головними перевагами є швидке та зручне редагування даних в базі, а також можливість швидкого копіювання БД, що є беззаперечним плюсом у роботі з великими обсягами даних та для відновлення даних після порушення цілісності системи.

phpMyAdmin забезпечує великі можливості в керуванні базами даних:

* вбудований пошук у певній базі даних;
* утиліта моніторингу поточного стану MySQL;
* панель для запиту за заданим шаблоном;
* інструменти імпорту та експорту;
* створення MySQL бази даних;
* панель для введення SQL команд;
* додавання до ДБ певних таблиць та подальше їх коректування, наповнення даними або виправлення;
* додавання користувачів до наявних баз даних;
* менеджер структури бази даних, що дозволяє візуальний моніторинг і коригування таблиць MySQL;
* дані про серверні змінних і налаштування.

## **3.2 Розробка бази даних**

Створена для комфортного зберігання інформації БД вміщує в собі наступні сутності (рис. 3.1):

Сутність «Завдання» має поля:

* іd – унікальний ідентифікатор завдання в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* назва – назва завдання, не може бути NULL, має тип text;
* текст – текст із завданням, може бути NULL, має тип text;
* іd предмету – ідентифікатор предмету, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* файл – прикріпленний файл, може бути NULL, має тип text;
* дата здачі – дата та час крайньої здачі завдання, не може бути NULL, має тип datetime;
* дата – дата створення завдання, заповнюється автоматично, не може бути NULL, має тип date;
* позиція – номер позиції на сторінці курсу, не може бути NULL, має тип int(11);
* видимість – видимість теми для учнів на сторінці курсу, не може бути NULL, має тип tinyint(1).

Сутність «Вчитель» має поля:

* іd – унікальний ідентифікатор вчителя в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* прізвище – прізвище вчителя, не може бути NULL, має тип text;
* ім’я – ім’я вчителя, не може бути NULL, має тип text;
* по батькові – по батькові, може бути NULL, має тип text;
* опис – загальна інформація про вчителя, його освітній ступінь та досягнення, не може бути NULL, має тип text;
* е-mail – електронна пошта вчителя для зв’язку з ним, не може бути NULL, має тип text;
* логін – логін вчителя для авторизації в системі, не може бути NULL, має тип text;
* пароль – пароль вчителя для авторизації в системі, не може бути NULL, має тип text.

Сутність «Предмет-Вчитель» має поля:

* іd предмету – ідентифікатор предмету, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* іd вчителя – ідентифікатор вчителя, не може бути NULL, є вторинним ключем, має тип int(10) unsigned.

Сутність «Тема» має поля:

* іd – унікальний ідентифікатор теми в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* назва – назва теми, не може бути NULL, має тип text;
* іd предмету – ідентифікатор предмета, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* дата – дата створення теми, заповнюється автоматично, не може бути NULL, має тип date;
* позиція – номер позиції на сторінці курсу, не може бути NULL, має тип int(11);
* видимість – видимість теми для учнів на сторінці курсу, не може бути NULL, має тип tinyint(1).

Сутність «Адміністратор» має поля:

* іd – унікальний ідентифікатор адміністратора в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* е-mail – електронна пошта адміністратора, не може бути NULL, має тип text;
* логін – логін адміністратора для авторизації в системі, не може бути NULL, має тип text;
* пароль – пароль адміністратора для авторизації в системі, не може бути NULL, має тип text.

Сутність «Клас» має поля:

* іd – унікальний ідентифікатор класу в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* номер – назва або номер класу, не може бути NULL, має тип text.

Сутність «Предмет-Клас» має поля:

* іd предмету – ідентифікатор предмету, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* іd класу – ідентифікатор класу, не може бути NULL, є вторинним ключем, має тип int(10) unsigned.

Сутність «Учень» має поля:

* іd – унікальний ідентифікатор учня в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* прізвище – прізвище учня, не може бути NULL, має тип text;
* ім’я – ім’я учня, не може бути NULL, має тип text;
* по батькові – по батькові учня, може бути NULL, має тип text;
* іd класу – ідентифікатор класу, не може бути NULL, є вторинним ключем, має тип int(10) unsigned.

Сутність «Предмет» має поля:

* іd – унікальний ідентифікатор предмета в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* назва – назва предмету, не може бути NULL, має тип text;
* опис – опис предмету, може бути NULL, має тип text.

Сутність «Запис» має поля:

* іd – унікальний ідентифікатор запису в системі, не може бути NULL, є первинним ключем , має тип int(10) unsigned;
* назва – назва запису, не може бути NULL, має тип text;
* текст – текст вмісту запису, не може бути NULL, має тип text;
* іd предмету – ідентифікатор предмета, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* дата – дата створення запису, заповнюється автоматично, не може бути NULL, має тип date;
* позиція – номер позиції на сторінці курсу, не може бути NULL, має тип int(11);
* видимість – видимість теми для учнів на сторінці курсу, не може бути NULL, має тип tinyint(1).

Сутність «Файл» має поля:

* іd – унікальний ідентифікатор файлу в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* назва – назва файлу, не може бути NULL, має тип text;
* файл – завантаженний файл, не може бути NULL, має тип text;
* іd предмету – ідентифікатор предмету, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* дата – дата створення файлу, заповнюється автоматично, не може бути NULL, має тип date;
* позиція – номер позиції на сторінці курсу, не може бути NULL, має тип int(11);
* видимість – видимість теми для учнів на сторінці курсу, не може бути NULL, має тип tinyint(1).

Сутність «Тест» має поля:

* іd – унікальний ідентифікатор тесту в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* назва – назва тесту, не може бути NULL, має тип text;
* опис – опис до тесту, може бути NULL, має тип text;
* іd предмету – ідентифікатор предмету, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* дата – дата створення тесту, заповнюється автоматично, не може бути NULL, має тип date;
* позиція – номер позиції на сторінці курсу, не може бути NULL, має тип int(11);
* видимість – видимість теми для учнів на сторінці курсу, не може бути NULL, має тип tinyint(1);
* дата початку – дата та час відкриття тестування, не може бути NULL, має тип datetime;
* дата кінця – дата та час закриття тесту, не може бути NULL, має тип datetime;
* час – час на проходження тесту, не може бути NULL, має тип int(11);
* питань на сторінку – кількість питань на одній сторінці, може бути NULL, має тип text;

Сутність «Оцінка» має поля:

* іd – унікальний ідентифікатор оцінки в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* іd учня – ідентифікатор учня, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* іd предмету – ідентифікатор предмету, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* тема – тема, з якої виставлено оцінку, не може бути NULL, має тип text;
* оцінка – значення оцінки, не може бути NULL, має тип int(11);
* дата – дата виставлення оцінки, заповнюється автоматично, не може бути NULL, має тип date.

Сутність «Виконане завдання» має поля:

* іd завдання – унікальний ідентифікатор завдання, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* іd студента – ідентифікатор студента, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* файл – прикріпленний файл з виконаним завданням, може бути NULL, має тип text;
* дата – дата створення вчителем, заповнюється автоматично, не може бути NULL, має тип date;
* коментар учня – коментар учня до виконаного завдання, може бути NULL, має тип text;
* коментар вчителя – коментар вчителя виконаного завдання учнем, може бути NULL, має тип text;
* оцінка – оцінка вчителя за завдання, не може бути NULL, має тип int(11).

Сутність «Результат тесту» має поля:

* іd питання – ідентифікатор питання, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* іd тестування учня – ідентифікатор тестування учня, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* відповідь – текст наданої відповіді, може бути NULL, має тип text;
* файл – прикріплений файл, може бути NULL, має тип text;
* бали – кількість балів за це запитання, не може бути NULL, має тип int(11).

Сутність «Відповідь» має поля:

* іd – унікальний ідентифікатор відповіді в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* текст – відповідь на питання, не може бути NULL, має тип text;
* іd питання – ідентифікатор питання, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* правильність – правильна чи неправильна відповідь, не може бути NULL, має тип tinyint(1).

Сутність «Учень-Тест» має поля:

* іd – унікальний ідентифікатор тестування учня, не може бути NULL, є первинним ключем, тип int(10) unsigned;
* іd тесту – ідентифікатор тесту, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* іd учня – ідентифікатор учня, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* бали – сумарна кількість балів отриманих за тест учнем, не може бути NULL, має тип int(11);
* час початку – дата та час коли учень почав проходити тест, не може бути NULL, має тип datetime;
* час кінця – дата та час коли учень завершив проходження тесту, може бути NULL, має тип datetime.

Сутність «Питання» має поля:

* іd – унікальний ідентифікатор питання в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* іd тесту – ідентифікатор тесту, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* текст – текст питання, не може бути NULL, має тип text;
* бали – кількість балів за це питання, не може бути NULL, має тип float;
* тип – тип питання, не може бути NULL, має тип text;
* позиція – номер позиції на сторінці курсу, не може бути NULL, має тип int(11).

Сутність «Сповіщення» має атрибути:

* іd сповіщення – ідентифікатор сповіщення, має тип int(11), є первинним ключем, не може бути NULL;
* іd користувача – ідентифікатор користувача, має тип int(10)unsigned, є вторинним ключем, не може бути NULL;
* текст – текст сповіщення, має тип text, може бути NULL;
* тип – тип сповіщення, яка може бути трьох типів: застереження, нагадування та застереження, а також відображається відповідним кольором в інтерфейсі, має тип int(11), не може бути NULL.

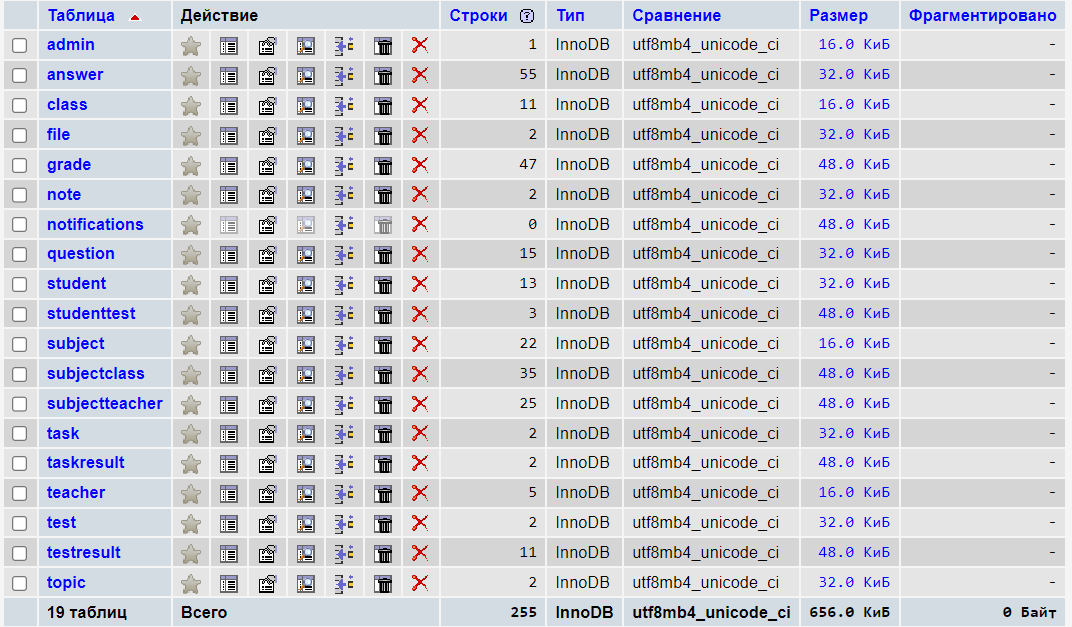


Рисунок 3.1 – Архітектура БД вебсервісу

## **3.3 Розробка користувацького інтерфейсу учня**

При розробці інтелектуальної інформаційної системи електронного навчання було створено програмний модуль, що реалізує функціонал учня. При першому вході на сайт, учень стикається з вікном входу в систему, що зображено на рисунку 3.2.

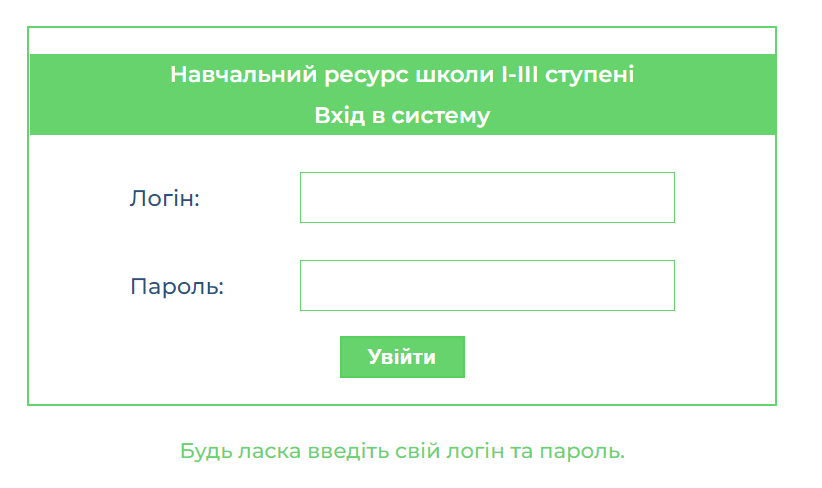


Рисунок 3.2 – Вікно входу в обліковий запис

Після вводу облікових даних користувачем система звіряє їх з даними БД і якщо відповідність знайдена відкриває користувачу його сторінку з обліковим записом. Якщо даних користувач не зареєстрований у системі або дані введено невірно система пропоную повторний ввід даних.

Зліва присутнє меню, в якому наведено перелік предметів доступних для учня, натиснувши на який відкривається сторінка курсу (рис. 3.3). На ній присутні всі доступні навчальні матеріали, які поділені за темами (рис. 3.4), а також можна надсилати виконані домашні завдання (рис. 3.5) чи проходити тести (рис. 3.6).

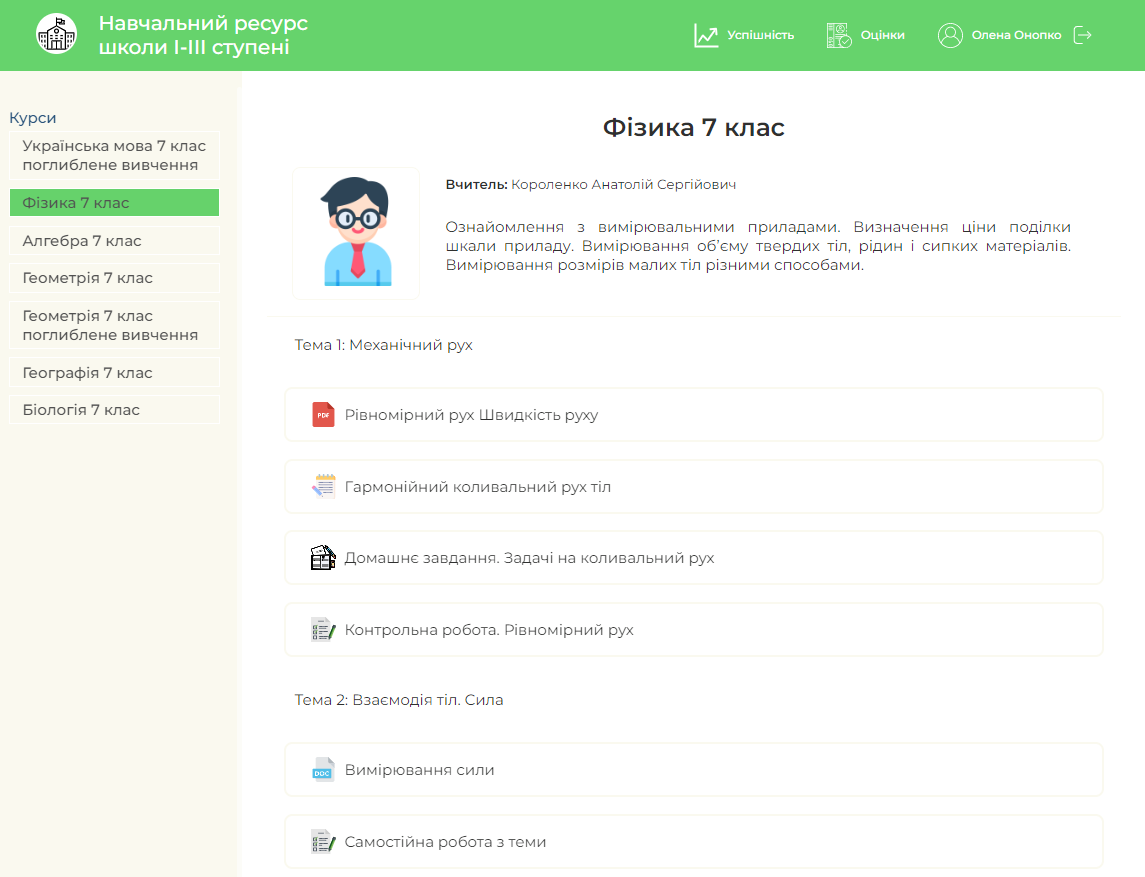


Рисунок 3.3 – Сторінка предмету

Система надає різні способи надання навчальних матеріалів, таких як текстові записи, якщо навчальний матеріал для учня налічує лише текст. Запис також може використовуватись для комунікації вчителя з учнем та надання загальної інформації.

Файл преставляє собою більш гнучкий спосіб передачі інформації, він буде завантажений на пристрій користувача після натискання та може мати різні формати. Дивлячись який тип навчального матеріалу система відображає відповідну іконку.



Рисунок 3.4 – Сторінка перегляду запису навчального матеріалу

Сторінка з домашнім завданням містить текст задач, таблицю з загальною інформацією стосовно завдання а також поля вводу відповіді. Статус завдання має три типи: «Порожнє», «Відправлене», «Оцінене».

Відповідь учень може додати у вигляді коментаря та прикріпленого файлу, а після натискання «Додати» система надсилає відповідь на оцінку вчителю.

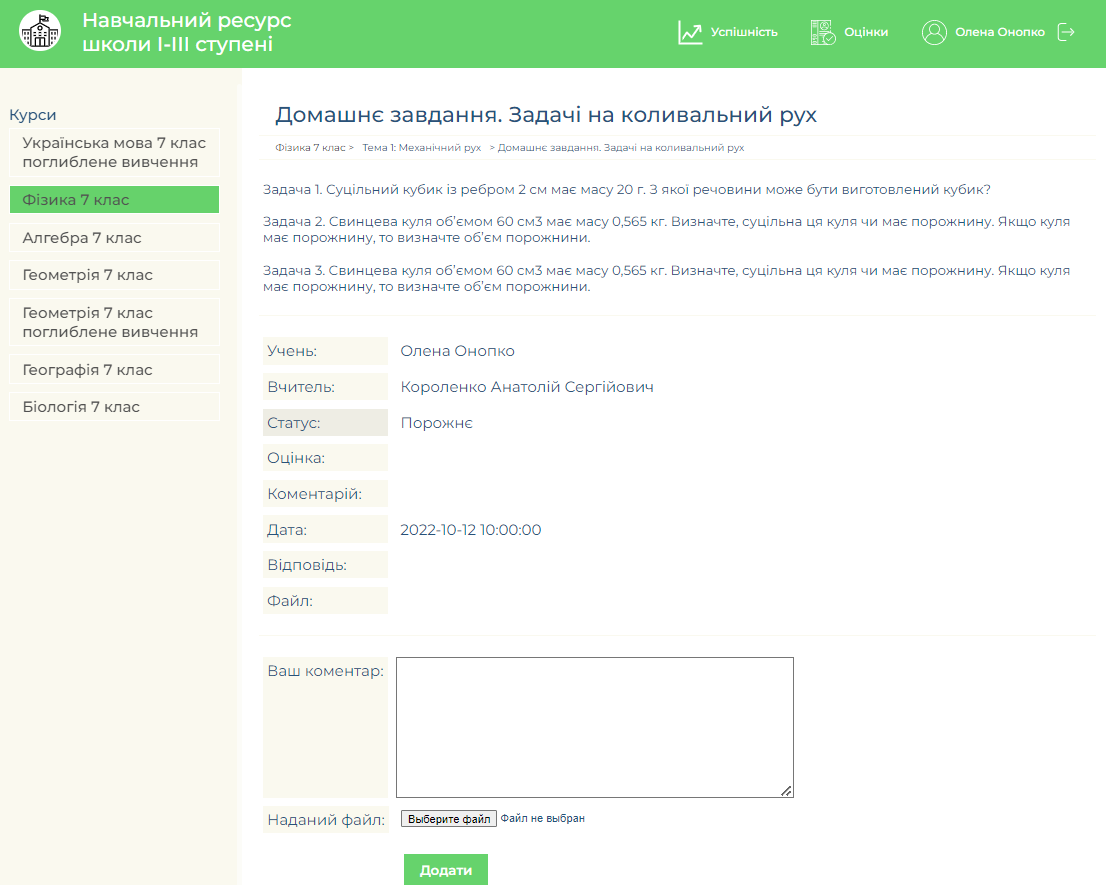


Рисунок 3.5 – Сторінка додавання домашнього завдання

Obsah obrázku text

Popis byl vytvořen automaticky

Рисунок 3.6 – Сторінка тесту

Відкривши сторінку тесту учень може або розпочати тестування натиснувши відповідну кнопку (рис. 3.7), або переглянути результати, якщо тест вже пройдено (рис. 3.8).

Під час виконання тесту, у випадку, якщо користувач має технічні негаразди, вже наявні відповіді та час зберігаються, а учень може продовжити проходження до закінчення часу.

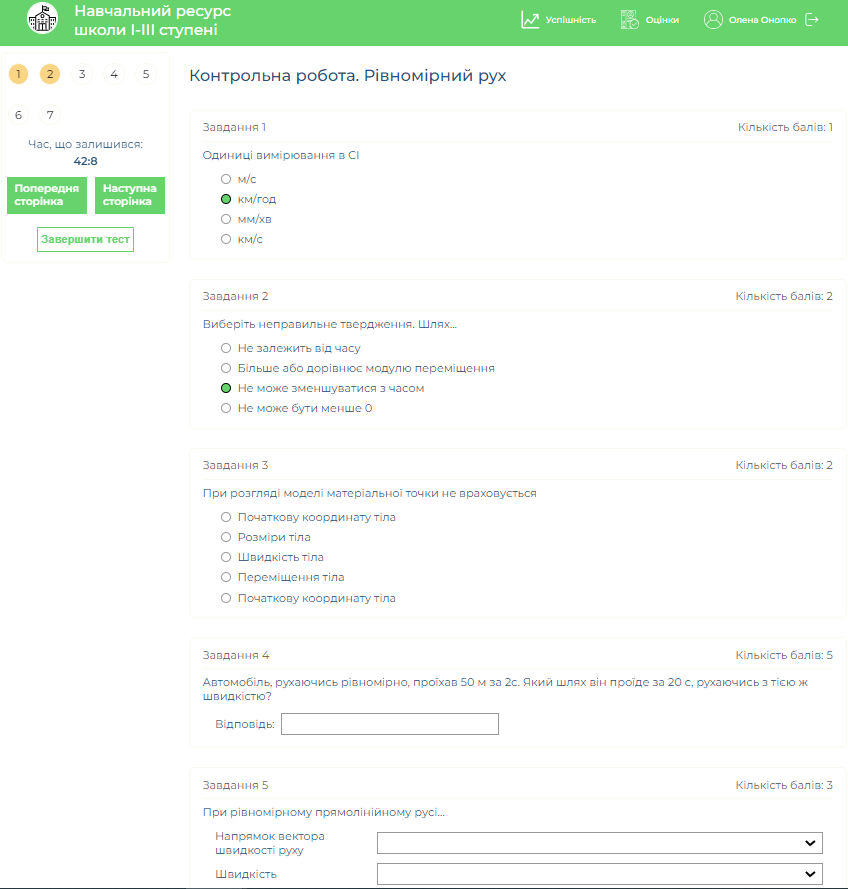


Рисунок 3.7 – Проходження тесту

Під час проходження тесту бокове меню зліва приховане, а на його місці з’являється перелік питань, натиснувши на які можна швидко перейти до потрібного питання, і підсвічені жовтим кольором якщо відповідь вже дана, а білим якщо ні. Також присутні кнопки «Наступне питання» та «Попереднє питання» для зручного переходу між сторінками тесту. Разом з цим тут відобрадається час, який залишився на проходження тесту і кнопка дострокового завершення.

Всі питання прономеровано та для кожного відведена своя кількість балів. В системі є такі види питань:

* одиничного вибору – має лише одну правильну відповідь та використовує input типу radio;
* множинного вибору – може мати декілька правильних відповідей та використовує input типу checkbox;
* на відповідність – для кожного виразу є одна правильна пара та використовує select і option;
* коротка відповідь – не має варіантів відповіді та використовує input типу text;
* есе – може мати відповідь у виді великого тексту або прикрипленого файлу та використовує textarea і input типу file.

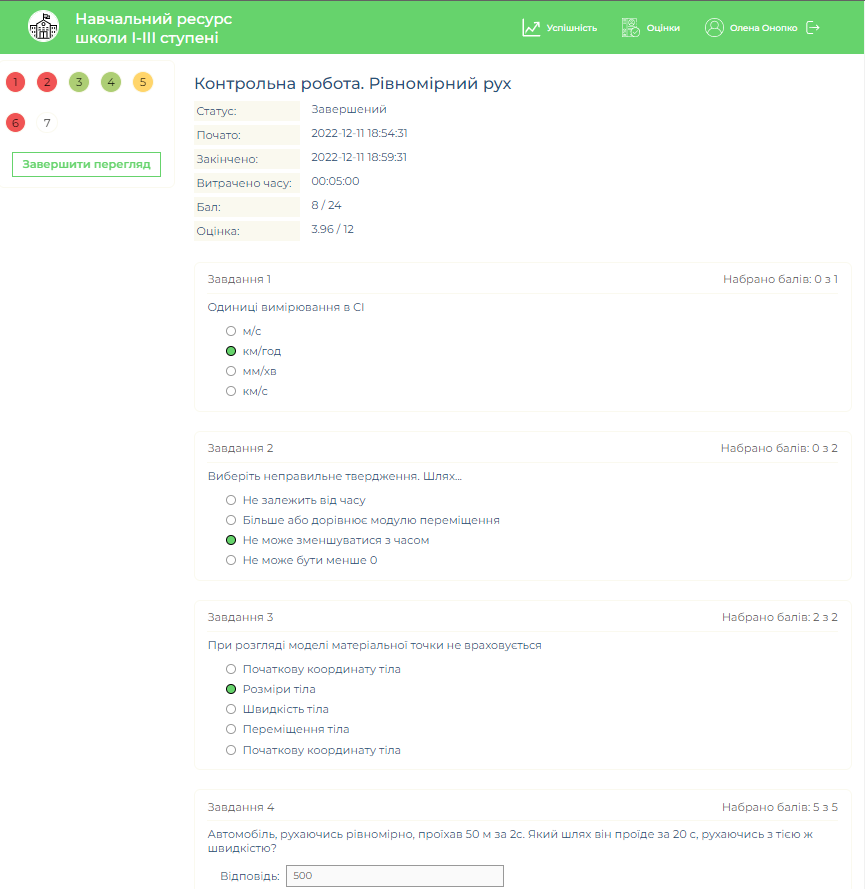


Рисунок 3.8 – Перегляд результатів тесту

При перегляді результату тесту система відображає загальну інформацію тесту про витрачений час, бал, оцінку, час початку та завершення. Також відображаються всі питання з вибраною раніше учнем відповіддю а також кількість набраних за нього балів та максимально можливу кількість. В лівому меню відображаються всі питання з позначеним кольором: з максимальною кількістю балів – зеленим кольором, частково правильні – жовтим, мінімальною – червоним, а білим – питання без відповіді.

Зверху сторінки присутнє меню, на якому натиснувши на пункт «Оцінки» учень може переглянути свої оцінки по предметам у вигляді таблиці (рис. 3.9).

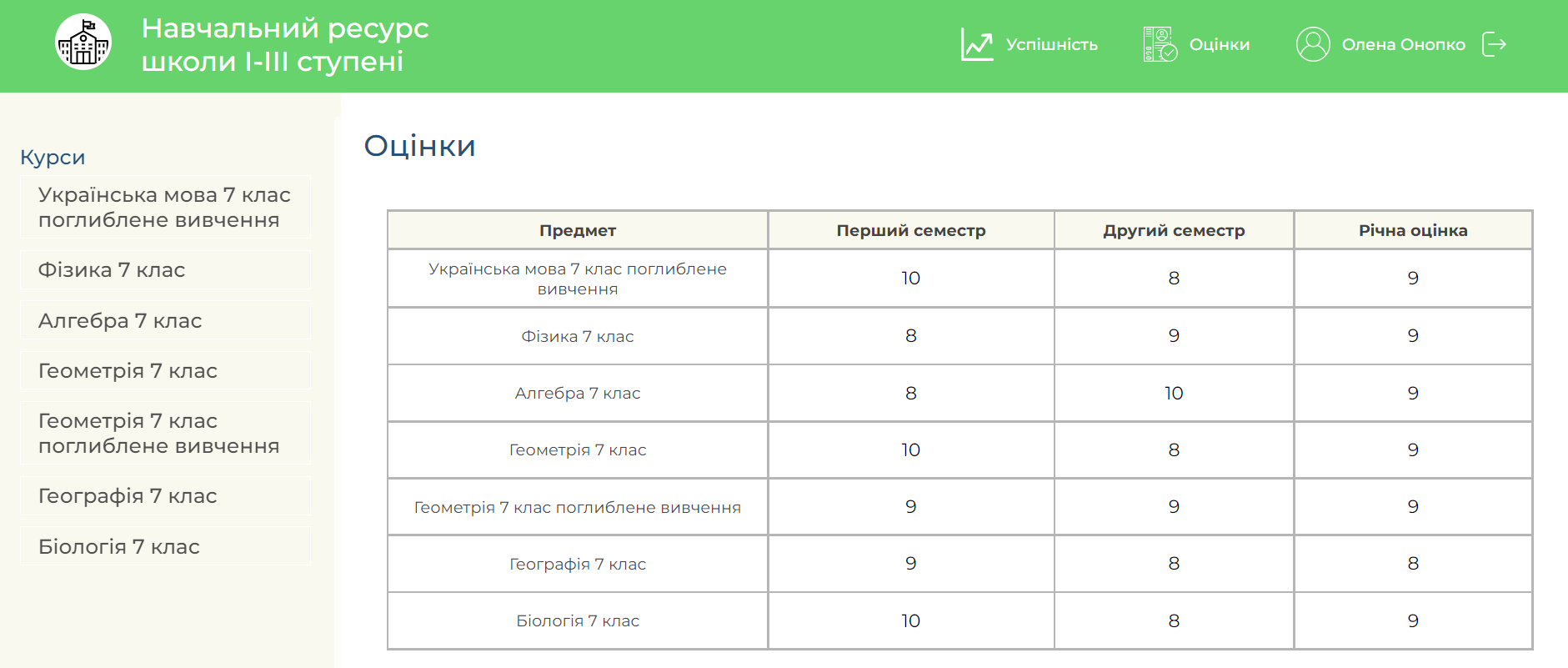


Рисунок 3.9 – Перегляд оцінок

У верхноьому меню є також посилання на сторінку «Успішність» завляки якій учень може відстежити свої оцінки у вигляді графіків та отримати корисні рекомендації на їх основі (рис. 3.10).

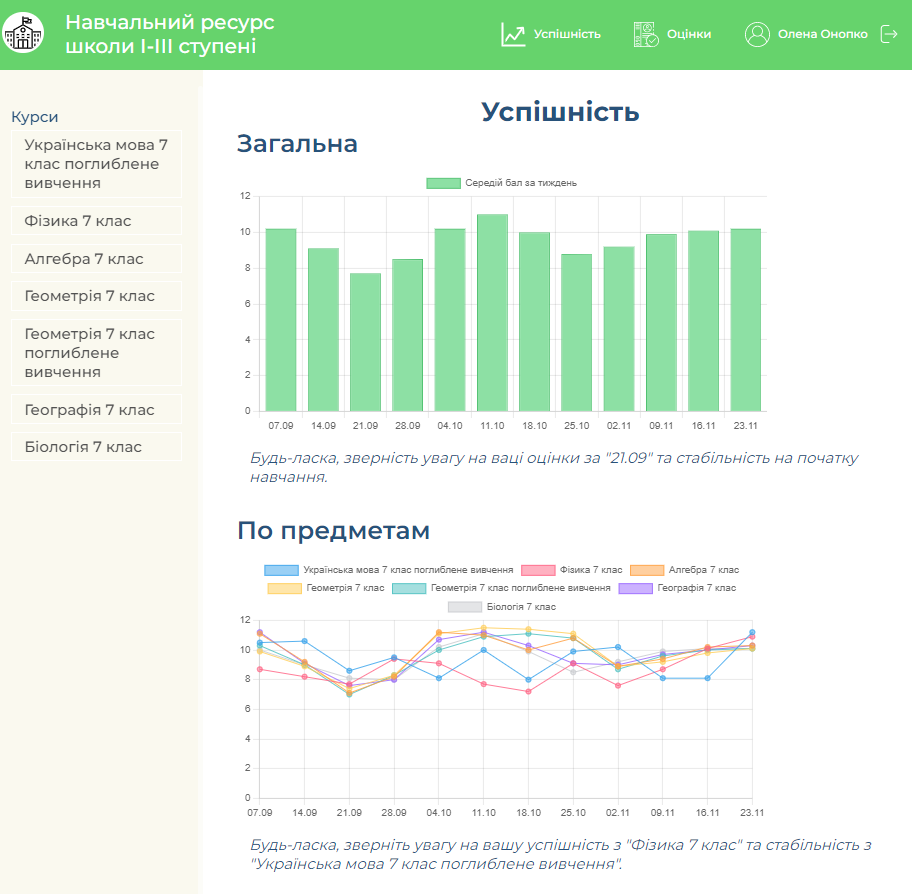


Рисунок 3.10 – Сторінка успішності учня

На першому графіку відображаються середні оцінки за тиждень в цілому, а на другому вони поділені по предметах для кращого розуміння сильних та слабких сторон учня. Також під кожним графіком відображаються зауваження та рекомендації щодо навчання.

## **3.4 Розробка користувацького інтерфейсу адміністратора**

Розроблено програмний модуль для реалізації функціоналу адміністратора інформаційної системи електронного навчання.

У меню зліва адміністратор може переходити між вкладками «Вчителі» «Класи», «Учні» та «Предмети». Перейшовши на вкладку «Класи» відображається таблиця зі всіма класами, наявними у системі, а також вся додаткова інформація: предмети які вивчає цей клас та учні що в ньому навчаються. Також на цій сторінці можна редагувати наявні записи, додавати нові чи видаляти старі. Аналогічні таблиці присутні на інших вкладках із таблицями вчителів, учнів, предметів або класів (рис. 3.11 – 3.14).

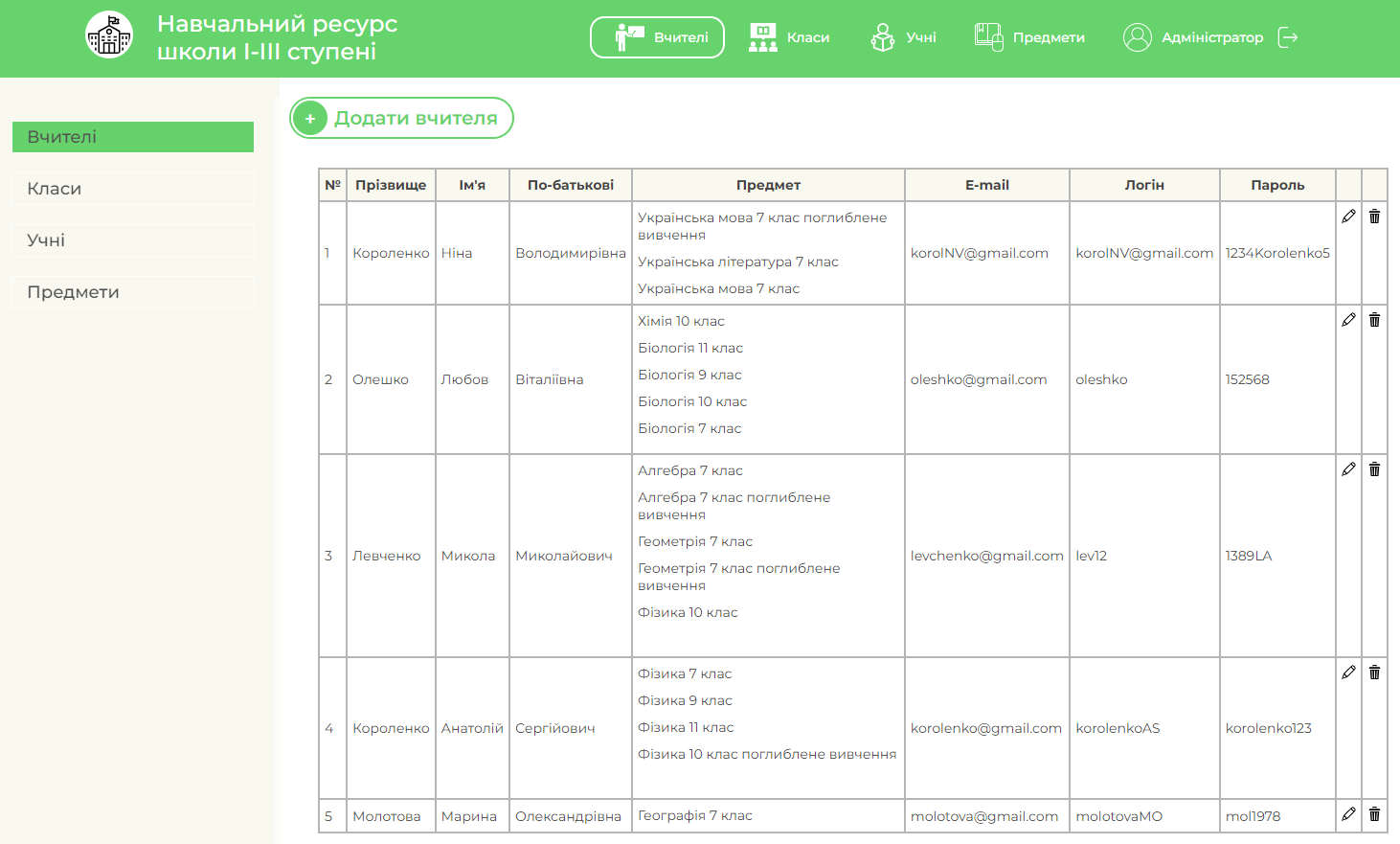


Рисунок 3.11 – Сторінка зі списком вчителів

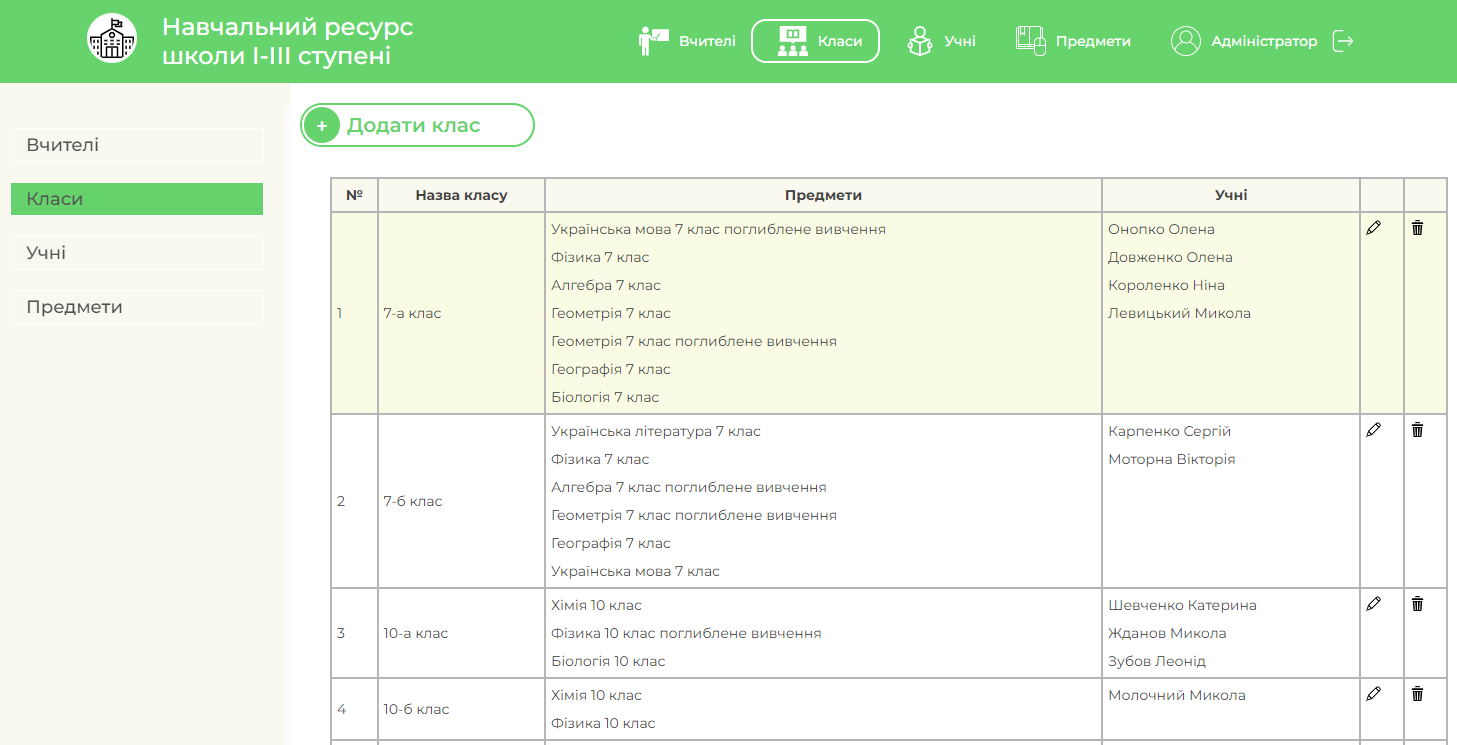


Рисунок 3.12 – Сторінка зі списком класів

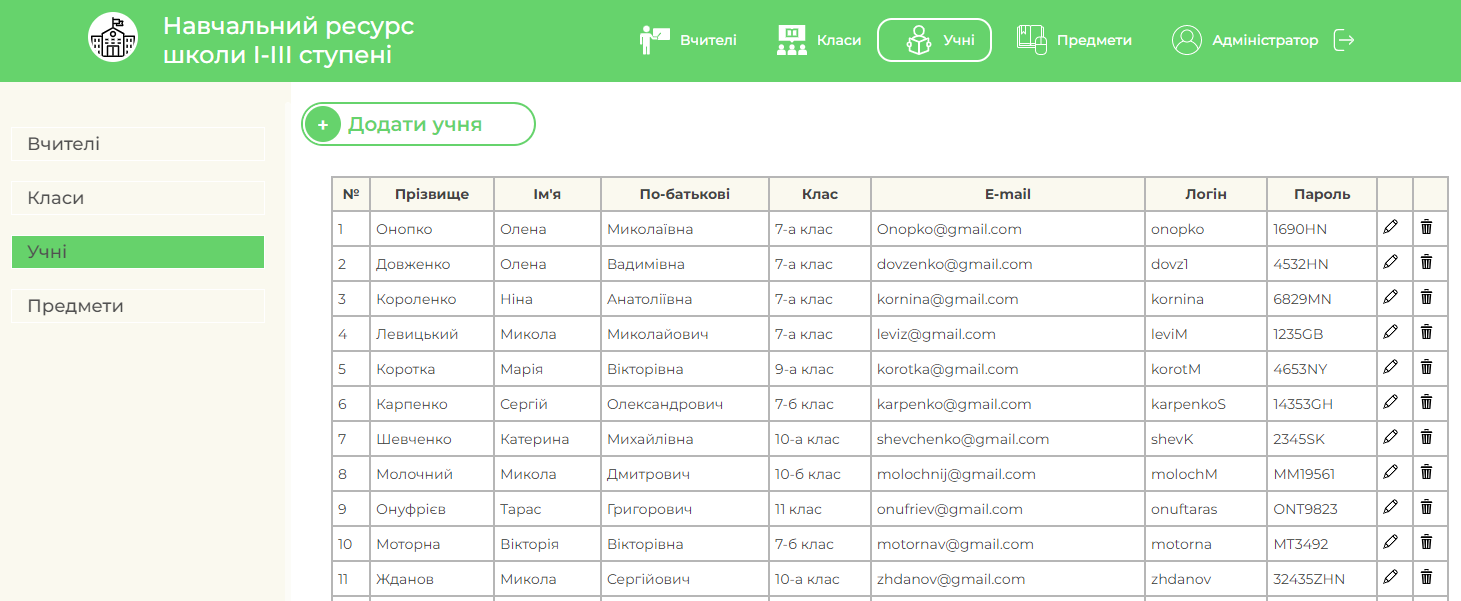


Рисунок 3.13 – Сторінка зі списком учнів

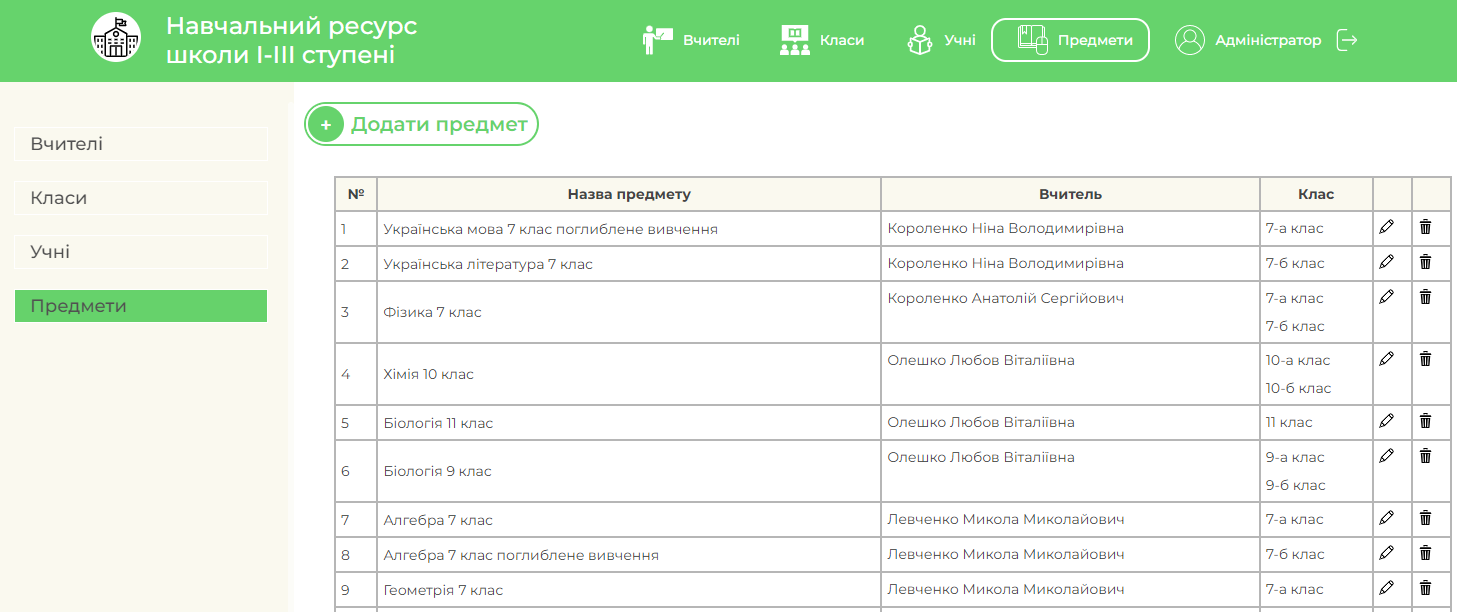


Рисунок 3.14 – Сторінка зі списком предметів

На сторінці додавання нового запису до системи присутні поля для вводу для яких використовуються об’єкти input типу text та випадаючі списки типу select (рис. 3.15 – 3.18).

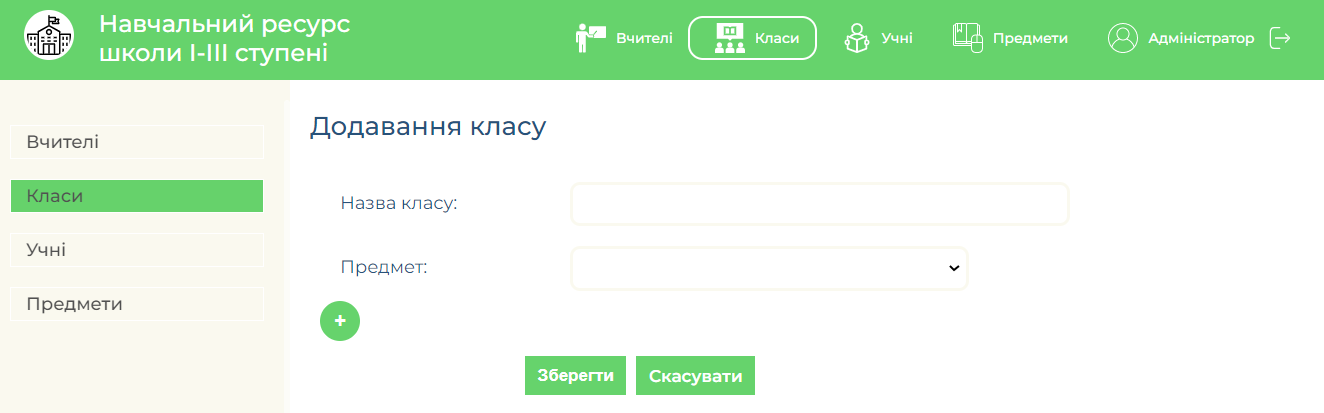


Рисунок 3.15 – Сторінка додавання класу

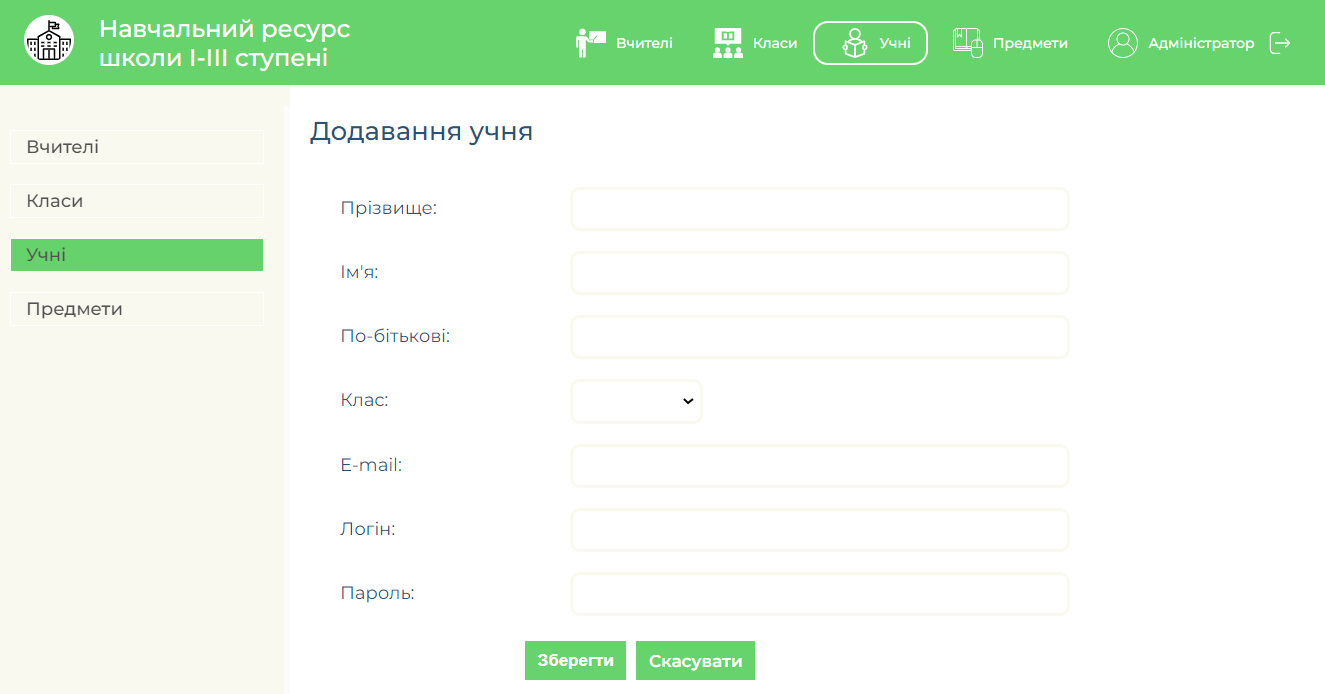


Рисунок 3.16 – Сторінка додавання учня

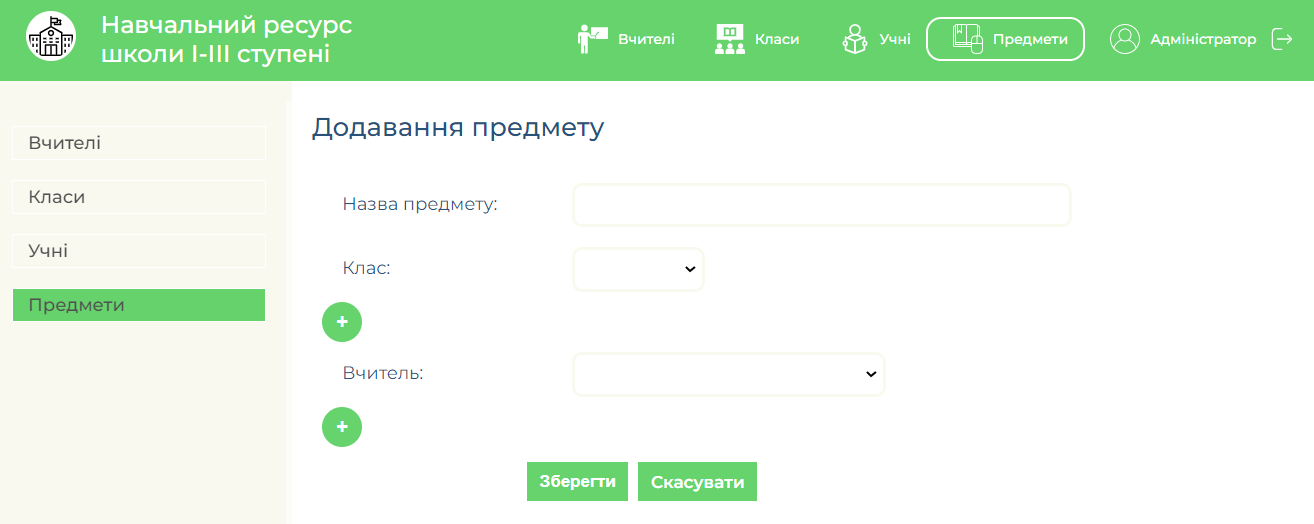


Рисунок 3.17 – Сторінка додавання предмету

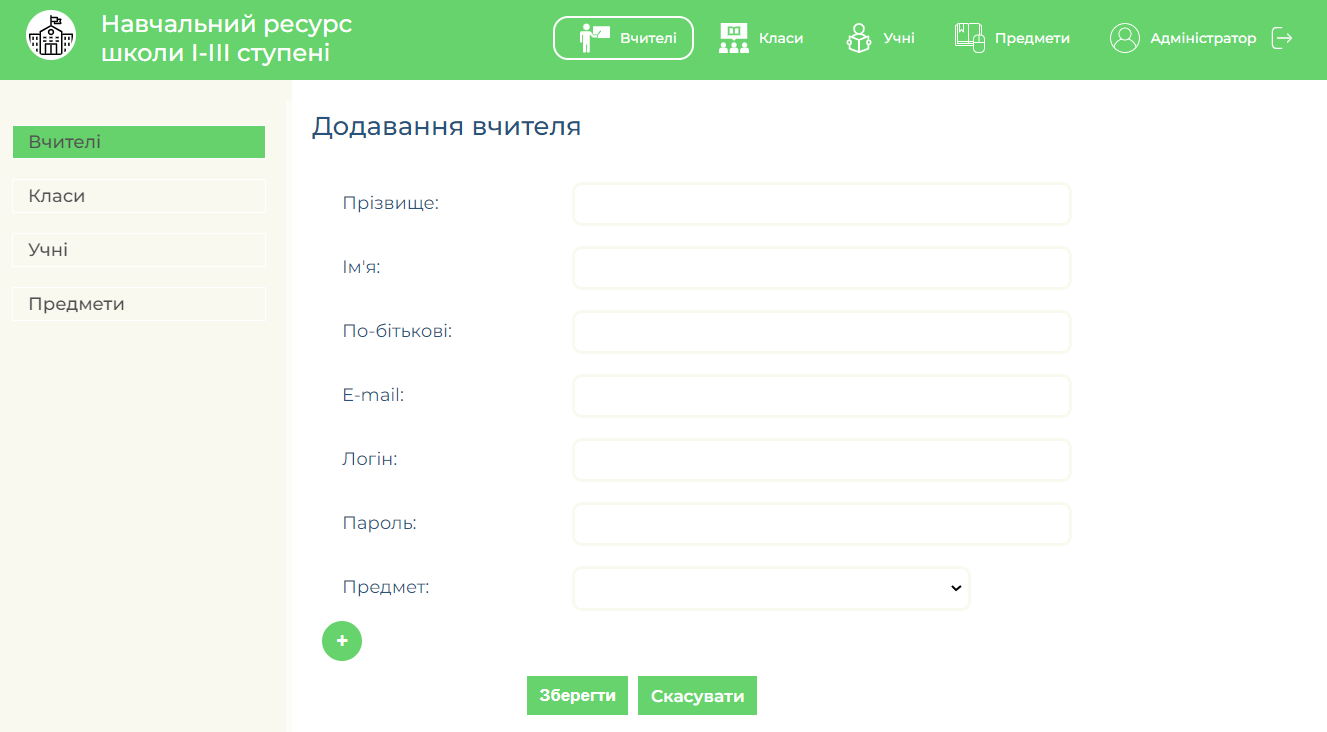


Рисунок 3.18 – Сторінка додавання вчителя

Для додавання декількох предметів, при додаванні вчителя на відповідній сторінці, потрібно натиснути кнопку зі знаком плюс, після чого з’явиться поле для вибору додаткового предмету.

# РОЗДІЛ 4 ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ

## **4.1 Вибір способу тестування**

Процес тестування програмних продуктів дозволяє дізнатися про більшість проблем, які можуть виникнути у користувача користувач у своїй роботі з платформою. Тестування є невід ємною частиною розробки якісного програмного забезпечення, а більше залишається необхідним і впродовж подальшої експлуатації. Головною метою тестування є можливість виявити відмінності у заявленому та кінцевому функціоналі фактичного програмного продукту.

Головними перевагами тестування ПЗ є:

* забезпечення безпеки;
* виявлення додаткових вимог до програмного продукту;
* перевірка фактичної відповідності поставленим вимогам;
* отримання інформації про якість програмного забезпечення;
* виявлення слабких місць на всіх етапах розробки [27].

Для перевірки якості та відповідності заявленим вимогам спроектованої системи вибрано функціональне тестування, через те що воно дозволяє перевірити коректну реалізацію поставлених можливостей практичним та наочним чином.

## **4.2 Тест план**

При тестуванні попередньо було створено тест план окремо для кожного модуля. Для модуля учня було визначено наступні основні функції, що буде протестовано:

* перегляд оцінок;
* проходження тесту;
* додавання виконаного домашнього завдання;
* перегляд курсу;
* перегляд навчальних матеріалів;
* перегляд результату тесту.

Далі, відповідно до визначених функцій, було створено тест-кейси відображені в таблиці 4.1.

Таблиця 4.1 – Тест-кейси для модуля учня

|  |  |  |
| --- | --- | --- |
| **Опис:** | Перегляд навчальних матеріалів | |
| **Умова:** | Виконано вхід в обліковий запис учня. | |
| Кроки: | Очікуваний результат: | |
| 1. В лівому навігаційному меню натиснути лівою кнопкою миші на посилання курсу. | Відбувається перехід на сторінку навчального предмета з відображеними навчальними матеріалами та тестами. | |
| 2. Лівою кнопкою миші натиснути на посилання навчального матеріалу. | Відкривається сторінка з обраним навчальним матеріалом. | |
| **Опис:** | Перегляд курсу | |
| **Умова:** | Виконано вхід в обліковий запис учня. | |
| Кроки: | Очікуваний результат: |
| 1 В лівому навігаційному меню натиснути лівою кнопкою миші на посилання курсу. | Відбувається перехід на сторінку навчального предмета з відображеними навчальними матеріалами та тестами. |
| **Опис:** | Додавання домашнього завдання |
| **Умова:** | Виконано вхід в обліковий запис учня. |
| Кроки: | Очікуваний результат: |
| 1. В лівому навігаційному меню натиснути лівою кнопкою миші на посилання курсу. | Відбувається перехід на сторінку навчального предмета з відображеними навчальними матеріалами та тестами. |
| 2. Лівою кнопкою миші натиснути на посилання з потрібним завданням. | Відкривається сторінка з інформацією про обране завдання та полями для додавання завдання. |

Продовження таблиці 4.1

|  |  |
| --- | --- |
| 3. Ввести коректні дані | Введено коректні дані. |
| 4. Натиснути кнопку «Додати». | Домашнє завдання надсилається вчителю для перевірки та його статус змінюється на «Відправлене». |
| **Опис:** | Перегляд оцінок |
| **Умова:** | Виконано вхід в обліковий запис учня. |
| Кроки: | Очікуваний результат: |
| 1. У верхньому меню натиснути лівою кнопкою миші на посилання «Оцінки». | Відбувається перехід на сторінку оцінок учня, де відображається таблиця з оцінками по кожному предмету. |
| **Опис:** | Проходження тесту |
| **Умова:** | Виконано вхід в обліковий запис учня. |
| Кроки: | Очікуваний результат: |
| 1. В лівому навігаційному меню натиснути лівою кнопкою миші на посилання курсу. | Відбувається перехід на сторінку навчального предмета з відображеними навчальними матеріалами та тестами. |
| 2. Лівою кнопкою миші натиснути на посилання з потрібним тестом. | Відкривається сторінка з детальною інформацією про обраний тест. |
| 3. Натиснути на посилання «Розпочати тест». | Відкривається сторінка проходження тесту з відображенням питань та можливістю відповіді на них. Ліве меню замінено на навігаційне меню тесту, з можливістю перейти на попереднє, наступне та будь-яке інше питання, або завершити тест. |
| 4. Надати відповіді на питання. | Всі відповіді збережено. |
| 5. Натиснути кнопку «Завершити тест» | Відбувається переадресація на сторінку перегляду результату тестування. |
| **Опис:** | Перегляд результату тесту |
| **Умова:** | Виконано вхід в обліковий запис учня. |
| Кроки: | Очікуваний результат: |
| 1. В лівому навігаційному меню натиснути лівою кнопкою миші на посилання курсу. | Відбувається перехід на сторінку навчального предмета з відображеними навчальними матеріалами та тестами. |

Продовження таблиці 4.1

|  |  |
| --- | --- |
| 2. Лівою кнопкою миші натиснути на посилання з потрібним тестом. | Відкривається сторінка з детальною інформацією про обраний тест. |
| 3. Натиснути на посилання «Переглянути результат». | Відкривається сторінка перегляду відповідей на тест, зі збереженими та оціненими відповідями, що надавав учень, правильні та неправильні відповіді виділено кольором. |
| 4. Натиснути на посилання «Завершити перегляд». | Відбувається переадресація на сторінку тесту. |

Для модуля адміністратора було визначено наступні основні функції, що буде протестовано:

* реєстрація учня в системі;
* створення навчального класу;
* створення предмету;
* реєстрація вчителя в системі.

Далі, відповідно до визначених функцій, було створено тест-кейси відображені в таблиці 4.2.

Таблиця 4.2 – Тест-кейси для модуля адміністратора

|  |  |
| --- | --- |
| **Опис:** | Створення нового класу |
| **Умова:** | Виконано вхід в обліковий запис адміністратора та відкрито сторінку «Класи» |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою натиснути на посилання «Додати клас» | Відкривається сторінка додавання нового навчального класу. |
| 2. Ввести назву класу | Введено назву. |
| 3. Обрати всі предмети, що викладаються в класі, з переліку запропонованих в об’єкті типу select | Предмет обрано. |
| 4. Натиснути кнопку «Зберегти» | Відбувається переадресація на сторінку «Класи» з останнім записом нового створеного класу. |
| **Опис:** | Створення нового предмету |

Продовження таблиці 4.2

|  |  |
| --- | --- |
| **Умова:** | Виконано вхід в обліковий запис адміністратора та відкрито сторінку «Предмети» |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою натиснути на посилання «Додати предмет» | Відкривається сторінка додавання нового навчального предмета. |
| 2. Ввести назву предмету | Введено назву. |
| 3. Обрати клас, в якому викладається даний предмет, з переліку запропонованих в об’єкті типу select | Клас обрано. |
| 3. Обрати вчителя, що викладає предмет, з переліку запропонованих в об’єкті типу select | Вчителя обрано. |
| 4. Натиснути кнопку «Зберегти» | Відбувається переадресація на сторінку «Предмети» з останнім записом нового створеного предмета. |
| **Опис:** | Реєстрація нового учня в системі |
| **Умова:** | Виконано вхід в обліковий запис адміністратора та відкрито сторінку «Учні» |
| Кроки: | Очікуваний результат: |
| 1. Натиснути лівою кнопкою на посилання «Додати учня» | Відкривається сторінка додавання нового учня. |
| 2. Ввести коректні особисті дані учня для реєстрації. | Введено коректні дані. |
| 3. Обрати клас, в якому навчається учень, з переліку запропонованих в об’єкті типу select | Клас обрано. |
| 4. Натиснути кнопку «Зберегти» | Відбувається переадресація на сторінку «Учні» з останнім записом нового зареєстрованого учня. |

Продовження таблиці 4.2

|  |  |
| --- | --- |
| **Опис:** | Реєстрація нового вчителя в системі |
| **Умова:** | Виконано вхід в обліковий запис адміністратора та відкрито сторінку «Вчителі» |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою натиснути на посилання «Додати вчителя» | Відкривається сторінка додавання нового вчителя. |
| 2. Ввести коректні особисті дані вчителя для реєстрації. | Введено коректні дані. |
| 3. Обрати всі предмети, що викладає вчитель, з переліку запропонованих в об’єкті типу select | Предмети обрано. |
| 4. Натиснути кнопку «Зберегти» | Відбувається переадресація на сторінку «Вчителі» з останнім записом нового зареєстрованого вчителя. |

# ВИСНОВОК

В ході виконання дипломного проектування було розроблено інтелектуальну інформаційну систему для електронного навчання з елементами штучного інтелекту для користувача учня. Для цього було використано серверну могу програмування PHP та для взаємодії з базою даних обрано MySQL.

Функціонал учня передбачає можливість перегляду сторінок курсів, перегляду або завантаження навчальних матеріалів, виконання тестів та домашніх робіт, перегляду своїх оцінок, а також можливість отримати коментарі та зауваження від вчителя.

На початку виконання роботи було досліджено предметну область, проаналізовано наявні аналоги та виявлено найважливіші аспекти на які варто звертати увагу в ході розробки.

На наступному етапі виконання роботи було спроектовано інтелектуальну інформаційну систему, було створено концептуальні схеми та діаграми використання. Також було створено схему бази даних, детально описано всі її сутності та зв'язки між ними. В результаті сформовано фізичну модель бази, яка включає в себе пов'язаних між собою 19-ти таблиць.

Наступним кроком був розроблений повнофункціональний користувацький інтерфейс для учнів та адміністратора. Створено тест план та тест-кейси та методом білого ящика було виконано ручне тестування функціоналу та інтерфейсу системи. В результаті чого отримали готову до роботи інтелектуальну інформаційну систему електронного навчання, яка покриває всі необхідні учням потреби.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кіяновська Н.М. Поняття електронного навчання в контексті сучасної педагогічної науки [Електронний ресурс] – Режим доступу: http://www.rusnauka.com/29\_DWS\_2012/Pedagogica/1\_120037.doc.htm
2. Дистанційне навчання як сучасна освітня технологія [Електронний ресурс] : матеріали міжвузівського вебінару (м. Вінниця, 31 березня 2017 р.) / відп. ред. Л.Б.Ліщинська. – Вінниця : ВТЕІ КНТЕУ, 2017. – 102 с.
3. Міністерство освіти України Організація дистанційного навчання в школі: методичні рекомендації – Київ, 2020. – 36 с.
4. Что такое система дистанционного обучения [Електронний ресурс] – Режим доступу: <https://www.mirapolis.ru/blog/chto-takoe-sdo>
5. О. Б. Вовк Системи електронного навчання – нові форми сучасної освіти // Математичні машини і системи №3. – 2015. – с. 79 – 86.
6. Система дистанционного обучения [Електронний ресурс] – Режим доступу: <https://teachbase.ru/learning/obuchenie/sistema-distancionnogo-obucheniya>  
   -obshij-obzor
7. Quality Criteria for Digital Learning Resources [Електронний ресурс]. – Режим доступу: [https://iktsenteret.no/sites/iktsenteret.no/files/attachments/quality\_  
   criteria\_dlr.pdf](https://iktsenteret.no/sites/iktsenteret.no/files/attachments/quality_criteria_dlr.pdf)
8. Катерняк І. Посібник з підготовки та організації електронного навчання. – К.: ТОВ «Фарбований лист», 2016.-48 с
9. Дизайн у дистанційному курсі [Електронний ресурс] – Режим доступу: <https://dl.khadi.kharkov.ua/mod/book/tool/print/index.php?id=37518>
10. ISO/IEC 25000:2014 Systems and software engineering [Електронний ресурс] – Режим доступу: <https://www.iso.org/obp/ui#iso:std:iso-iec:25000>
11. K. D. Schewe, and B. Thalheim, "Design and development of Web information systems", in Systematic Development of Web Information Systems. Springer-Verlag GmbH Germany, part of Springer Nature, 2019, с. 471 – 529.
12. Світовий досвід організації та розвитку університетської системи дистанційного навчання / Ін-т вищої освіти КНЕУ ім. В. Гетьмана ; уклад.: Кулага І. В., Ільницький Д. О., Стрельник С. О. – Київ, 2013. – 38 с. – Текст доступний в Інтернеті: <http://kneu.edu.ua/userfiles/education2_0/13-4713_verstka(1).pdf>.
13. ATutor [Електронний ресурс] – Режим доступу: <http://www.atutor.ca>
14. Docebo [Електронний ресурс] – Режим доступу: <https://www.docebo.com>
15. Ispring Learn [Електронний ресурс] – Режим доступу: <https://www.ispringsolutions.com>
16. Обзор СДО Docebo: возможности и решаемые бизнес-задачи [Електронний ресурс] – Режим доступу: <https://lmslist.ru/sdo/obzor-docebo>
17. Docebo Software 2018 Pricing & Features // Business Software Reviews, SaaS & Cloud Applications Directory | GetApp [Электронный ресурс] – Режим доступа: https://www.getapp.com/hr-employee-management-software/a/ docebo-e-learning
18. iSpring Learn Pricing, Features, Reviews & Comparison of Alternatives // Business Software Reviews, SaaS & Cloud Applications Directory | GetApp [Электронный ресурс] – Режим доступа: https://www.getapp.com/ education-childcare-software/a/ispring-learn
19. Обзор СДО iSpring Learn: возможности и решаемые бизнес-задачи [Электронный ресурс] – Режим доступа: <https://lmslist.ru/sdo/obzor-ispring-online>
20. Як штучний інтелект може допомогти освіті [Електронний ресурс] – Режим доступу: <https://osvitoria.media/experience/yak-shtuchnyj-intelekt-mozhe-dopomogty-osviti>
21. Штучний Інтелект в освітньому процесі [Електронний ресурс] – Режим доступу: https://www.futureschool.online/post/artificial\_intelligence/
22. Штучний інтелект в сфері освіти [Електронний ресурс] – Режим доступу: <https://vlada-rykova.com/ua/iskusstvennyj-intellekt-v-obrazovanii/>
23. Machine Learning – Машинне навчання [Електронний ресурс] – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning>
24. Pierce D., Hathaway A. The Promise (and Pitfalls) of AI for Education. [Електронний ресурс] – Режим доступу: [https://thejournal.com/Articles/2018  
    /08/29/The-Promise-of-AI-for-Education.aspx?Page=1](https://thejournal.com/Articles/2018/08/29/The-Promise-of-AI-for-Education.aspx?Page=1)
25. Язык программирования PHP [Електронний ресурс] – Режим доступу: <https://depix.ru/articles/yazyk_programmirovaniya_php>
26. Что Такое AJAX И Как Он Работает [Електронний ресурс] – Режим доступу: <https://www.hostinger.ru/rukovodstva/chto-takoje-ajax>
27. Видытестирования программного обеспечения [Електронний ресурс] – Режим доступу:<http://www.protesting.ru/testing/testtypes.html>

# ДОДАТОК А ЛІСТИНГ КОДУ ДЛЯ РЕАЛІЗАЦІЇ ТЕСТУВАННЯ

<?php

session\_start();

if(!isset($\_SESSION["userType"])){

header("Location: ../");

die;

}

require\_once('../functions/connection.php');

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="../style.css">

<link rel="icon" href="../img/favicon.svg" />

<title>School</title>

</head>

<body >

<?php require\_once('../blocks/\_header-2.php'); ?>

<main class="main">

<?php

if(isset($\_GET['result']) || isset($\_GET['review'])){

if(isset($\_GET['result'])) $id = $\_GET['result']; else $id = $\_GET['review'];

$query = mysqli\_query($db, "SELECT \* FROM studenttest WHERE id = $id");

if(mysqli\_num\_rows($query) == 1) $studenttest = mysqli\_fetch\_all($query);

else {echo "Error 404"; die;}

$query = mysqli\_query($db, "SELECT \* FROM testresult WHERE id\_studenttest = {$studenttest[0][0]}");

$testresult = mysqli\_fetch\_all($query);

$query = mysqli\_query($db, "SELECT \* FROM test WHERE id = {$studenttest[0][1]}");

$test = mysqli\_fetch\_all($query);

$query = mysqli\_query($db, "SELECT \* FROM question WHERE id\_test = {$studenttest[0][1]} ORDER BY position ASC");

$question = mysqli\_fetch\_all($query);

echo '

<div class="test-nav">

<div class="test-nav\_\_quest">';

for($i = 0; isset($question[$i]); $i++){

$tf = false;

for($j = 0; isset($testresult[$j]); $j++){

if($testresult[$j][0] == $question[$i][0]){

if($testresult[$j][4] == '0'){

echo '<a href="#" class="test-nav\_\_quest-notcorect">'.($i+1).'</a>';

}elseif($testresult[$j][4] == $question[$i][3]){

echo '<a href="#"class="test-nav\_\_quest-corect">'.($i+1).'</a>';

}elseif($testresult[$j][4] == NULL){

echo '<a href="#"class="">'.($i+1).'</a>';

}else{

echo '<a href="#"class="test-nav\_\_quest-partcorect">'.($i+1).'</a>';

}

$tf = true;

}

}

if(!$tf) echo '<a href="#"class="">'.($i+1).'</a>';

}

echo '</div>

<form class="test-nav\_\_end-btn">

<a href="'.$\_SERVER['HTTP\_REFERER'].'">

Завершити перегляд

</a>

<form>

</div>

<section class="test-wrap ">

<div class="container">

<h3 class="homework-title">

'.$test[0][1].'

</h3>

<p class="test-result-info">

<span>Статус: </span>'; if($studenttest[0][5] != '') echo 'Завершений';

else echo 'В процесі';

echo '</p>

<p class="test-result-info">

<span>Почато: </span>'.$studenttest[0][4].'</p>

<p class="test-result-info">

<span>Закінчено: </span>'; if($studenttest[0][5] != '') echo $studenttest[0][5];

echo '</p>

<p class="test-result-info">

<span>Витрачено часу: </span> ';

if($studenttest[0][5] != '') {

$query = mysqli\_query($db, "SELECT TIMEDIFF(endTime, startTime) FROM studenttest WHERE id = {$studenttest[0][0]}");

$time = mysqli\_fetch\_all($query);

echo $time[0][0];

}

echo '

</p>

<p class="test-result-info">

<span>Бал: </span> ';

if($studenttest[0][5] != '') {

$query = mysqli\_query($db, "SELECT SUM(points) FROM testresult WHERE id\_studenttest = {$studenttest[0][0]}");

$points = mysqli\_fetch\_all($query);

$query = mysqli\_query($db, "SELECT SUM(points) FROM question WHERE id\_test = {$test[0][0]}");

$allpoints = mysqli\_fetch\_all($query);

echo round($points[0][0], 2).' / '.$allpoints[0][0];

}

echo '

</p>

<p class="test-result-info">

<span>Оцінка: </span>';

echo round($points[0][0]/$allpoints[0][0], 2);

echo ' / 12

</p>';

for($i = 0; isset($question[$i]); $i++){

$query = mysqli\_query($db, "SELECT \* FROM testresult WHERE id\_question = {$question[$i][0]} AND id\_studenttest = $id");

$testresult = mysqli\_fetch\_all($query);

if($testresult[0][4] != '') $tr = $testresult[0][4];

elseif($testresult[0][0] != '') $tr = '?';

else $tr = 0;

echo '<div class="test-question-block';

if($question[$i][4] == 'vid') echo ' test-question\_\_vid';

echo '">

<div class="test-question\_\_title">

<p>Завдання '.($i+1).'</p>

<p>Набрано балів: <span>'.$tr.' з '.$question[$i][3].'

</span></p>

</div>

<p class="test-question\_\_text line">

'.$question[$i][2].'

</p>';

if($question[$i][4] == 'one'){

$query = mysqli\_query($db, "SELECT \* FROM answer WHERE id\_question = {$question[$i][0]}");

$answer = mysqli\_fetch\_all($query);

for($j = 0; isset($answer[$j]); $j++){

echo '<div class="test-question\_\_answ">

<input disabled type="radio"'; if($j == $testresult[0][2] && '' != $testresult[0][2]) echo ' checked';

echo '>

<label for="'.$i.$j.'">

'.$answer[$j][2].'

</label>

</div>';

}

}

elseif($question[$i][4] == 'multi'){

$res = explode('///' , $testresult[0][2]);

$query = mysqli\_query($db, "SELECT \* FROM answer WHERE id\_question = {$question[$i][0]}");

$answer = mysqli\_fetch\_all($query);

for($j = 0; isset($answer[$j]); $j++){

echo '

<div class="test-question\_\_answ">

<input disabled type="checkbox" '; foreach($res as $v){if($v == $j && $v != '') echo ' checked';}

echo '>

<label for="'.$i.$j.'">

'.$answer[$j][2].'

</label>

</div>';

}

}

elseif($question[$i][4] == 'short'){

echo '

<div class="test-question\_\_answ">

<label for="1">

Відповідь:

</label>

<input disabled type="text" value="'.$testresult[0][2].'" class="test-question\_\_answ-short">

';

}

elseif($question[$i][4] == 'vid'){

$res = explode('///' , $testresult[0][2]);

$query = mysqli\_query($db, "SELECT \* FROM answer WHERE id\_question = {$question[$i][0]}");

$answer = mysqli\_fetch\_all($query);

$k = 0;

for($j = 0; isset($answer[$j]); $j++){

if(substr($answer[$j][2], 0, strpos($answer[$j][2], "///")) == '') continue;

echo '

<div class="test-question\_\_answ">

<label for="'.$i.$j.'">

'.substr($answer[$j][2], 0, strpos($answer[$j][2], "///")).'

</label>

<select disabled name="que'.$i.$j.'" id="'.$i.$j.'">

<option>

'.$res[$k].'

</option>

</select>

</div>';

$k++;

}

}

elseif($question[$i][4] == 'essay'){

echo '

<div class="test-question\_\_answ-essay">

<label for="1">

Відповідь:

</label>

<div class="">

<a href="../course/files/'.$studenttest[0][2].'-'.$i.'-testresult-'.$testresult[0][3].'" download="'.$testresult[0][3].'"> '.$testresult[0][3].'</a>

<textarea disabled cols="60" rows="10">'.$testresult[0][2].'</textarea>

</div>';

$query = mysqli\_query($db, "SELECT points FROM testresult WHERE id\_question = {$question[$i][0]} AND id\_studentTest = $\_GET[review]");

$grade = mysqli\_fetch\_all($query);

}

echo '</div>

</div>';

}

}

elseif(isset($\_GET['id'])){

$query = mysqli\_query($db, "SELECT \* FROM studenttest WHERE id\_student = $\_SESSION[userID] AND id\_test = $\_GET[id]");

if(mysqli\_num\_rows($query) == 1){

$studenttest = mysqli\_fetch\_all($query);

$query = mysqli\_query($db, "SELECT \* FROM testresult WHERE id\_studenttest = {$studenttest[0][0]}");

$testresult = mysqli\_fetch\_all($query);

}

else{

$query = mysqli\_query($db, "INSERT INTO studenttest (id\_student, id\_test) VALUES ($\_SESSION[userID], $\_GET[id])");

$query = mysqli\_query($db, "SELECT \* FROM studenttest WHERE id\_student = $\_SESSION[userID] AND id\_test = $\_GET[id]");

$studenttest = mysqli\_fetch\_all($query);

}

$query = mysqli\_query($db, "SELECT \* FROM test WHERE id = $\_GET[id]");

$test = mysqli\_fetch\_all($query);

$query = mysqli\_query($db, "SELECT \* FROM question WHERE id\_test = $\_GET[id] ORDER BY position ASC");

$question = mysqli\_fetch\_all($query);

echo '

<div class="test-nav">

<div class="test-nav\_\_quest">';

if($test[0][10] != 'Усі'){

$i = 0; $j = 1;

while(isset($question[$i])){

$pages = $j;

for($k = 0; $k<$test[0][10] && isset($question[$i]); $k++){

echo '<a href="?id='.$\_GET['id'].'&page='.$j.'"';

for($l = 0; isset($testresult[$l]); $l++){

if($testresult[$l][0] == $question[$i][0]) {echo ' class="test-nav\_\_quest-answered"'; break;}

}

echo '>'.($i+1).'</a>';

$i++;

}

$j++;

}

}

else{

for($i = 0; isset($question[$i]); $i++){

echo '<a href="#"';

for($l = 0; isset($testresult[$l]); $l++){

if($testresult[$l][0] == $question[$i][0]) {echo ' class="test-nav\_\_quest-answered"'; break;}

}

echo '>'.($i+1).'</a>';

}

}

echo '</div>

<p>Час, що залишився:</p>

<p class="test-nav\_\_time" id="timer"></p>

<div class="test-nav\_\_btns">

<div>

<a href="#" class="button test-nav\_\_end-btn-prev">

Попередня <br> сторінка

</a>

</div>

<div>

<a href="#" class="button test-nav\_\_end-btn-next">

Наступна <br> сторінка

</a>

</div>

</div>

<form action="" class="test-nav\_\_end-btn">

<button type="button" class="test-nav\_\_end-btn-end">

Завершити тест

</button>

</form>

</div>

<form action="testquery.php" method="POST" class="test-wrap" enctype="multipart/form-data">

<input name="action" value="end" id="action" class="d-none">

<input name="testid" value="'.$\_GET['id'].'" class="d-none">

<input name="page" value="'.$\_GET['page'].'" class="d-none">

<input name="studenttest" value="'.$studenttest[0][0].'" class="d-none">

<div class="container">

<h3 class="homework-title">

'.$test[0][1].'

</h3>';

if($test[0][10] != 'Усі'){

$count = (($\_GET['page']-1)\*$test[0][10]);

$condition = ($\_GET['page']\*$test[0][10]);

}

else{

$count = 0;

$condition = count($question) + 1;

}

for($i = $count; $i<$condition && isset($question[$i]); $i++){

echo '<div class="test-question-block';

if($question[$i][4] == 'vid') echo ' test-question\_\_vid';

echo '">

<div class="test-question\_\_title">

<p>Завдання '.($i+1).'</p>

<p>Кількість балів: <span>'.$question[$i][3].'

</span></p>

</div>

<p class="test-question\_\_text line">

'.$question[$i][2].'

</p>';

if($question[$i][4] == 'one'){

$query = mysqli\_query($db, "SELECT \* FROM answer WHERE id\_question = {$question[$i][0]}");

$answer = mysqli\_fetch\_all($query);

for($j = 0; isset($answer[$j]); $j++){

echo '<div class="test-question\_\_answ">

<input type="radio" name="que'.$i.'" id="'.$i.$j.'" value="'.$j.'"';

for($l = 0; isset($testresult[$l]); $l++){

if($testresult[$l][2] == $j) {echo ' checked'; break;}

}

echo '>

<label for="'.$i.$j.'">

'.$answer[$j][2].'

</label>

</div>';

}

}

elseif($question[$i][4] == 'multi'){

$query = mysqli\_query($db, "SELECT \* FROM answer WHERE id\_question = {$question[$i][0]}");

$answer = mysqli\_fetch\_all($query);

for($j = 0; isset($answer[$j]); $j++){

echo '

<div class="test-question\_\_answ">

<input type="checkbox" name="que'.$i.$j.'" id="'.$i.$j.'" value="'.$j.'"';

for($l = 0; isset($testresult[$l]); $l++){

if($question[$i][0] == $testresult[$l][0]) {

$res = explode('///' , $testresult[$l][2]);

foreach($res as $v){if($v == $j && $v != '') echo ' checked';}

break;

}

}

echo '>

<label for="'.$i.$j.'">

'.$answer[$j][2].'

</label>

</div>';

}

}

elseif($question[$i][4] == 'short'){

echo '

<div class="test-question\_\_answ">

<label for="1">

Відповідь:

</label>

<input type="text" name="que'.$i.'" id="'.$i.'" class="test-question\_\_answ-short"';

for($l = 0; isset($testresult[$l]); $l++){

if($question[$i][0] == $testresult[$l][0]) {

echo ' value="'.$testresult[$l][2].'"';

}

}

echo '>';

}

elseif($question[$i][4] == 'vid'){

$query = mysqli\_query($db, "SELECT \* FROM answer WHERE id\_question = {$question[$i][0]}");

$answer = mysqli\_fetch\_all($query);

for($l = 0; isset($testresult[$l]); $l++){

if($question[$i][0] == $testresult[$l][0]) {

$res = explode('///' , $testresult[$l][2]);

break;

}

}

for($j = 0, $a = 0; isset($answer[$j]); $j++){

if(substr($answer[$j][2], 0, strpos($answer[$j][2], "///")) == '') continue;

echo '

<div class="test-question\_\_answ">

<label for="'.$i.$j.'">

'.substr($answer[$j][2], 0, strpos($answer[$j][2], "///")).'

</label>

<select name="que'.$i.$j.'" id="'.$i.$j.'">

<option value="">

</option>';

foreach($answer as $an){

echo '<option value="'.substr($an[2], 3+strpos($an[2], "///")).'"';

if($res[$a] == substr($an[2], 3+strpos($an[2], "///"))) echo ' selected';

echo '>

'.substr($an[2], 3+strpos($an[2], "///")).'

</option>';

}

echo '</select>

</div>

';

$a++;

}

}

elseif($question[$i][4] == 'essay'){

echo '

<div class="test-question\_\_answ-essay">

<label for="1">

Відповідь:

</label>

<div class="">';

for($l = 0; isset($testresult[$l]); $l++){

if($question[$i][0] == $testresult[$l][0]) {

echo '<a href="files/'.$\_SESSION['userID'].'-'.$i.'-testresult-'.$testresult[$l][3].'" download="'.$testresult[$l][3].'">'.$testresult[$l][3].'</a>';

}

}

echo '

<input name="file'.$i.'" type="file"><textarea name="que'.$i.'" id="'.$i.'" cols="60" rows="10">';

for($l = 0; isset($testresult[$l]); $l++){

if($question[$i][0] == $testresult[$l][0]) {

echo $testresult[$l][2];

}

}

echo '</textarea>

</div>

';

}

echo '</div>

</div>';

}

echo '

</div>

</form>

';

if((substr($studenttest[0][4], 14, 2)+$test[0][9]) >= 60){

$m = (substr($studenttest[0][4], 14, 2)+$test[0][9]-60);

$h = (substr($studenttest[0][4], 11, 2)+1);

if($m<10) $m='0'.$m;

if($h<10) $h='0'.$h;

}

else{

$m = (substr($studenttest[0][4], 14, 2)+$test[0][9]);

$h = (substr($studenttest[0][4], 11, 2));

}

echo '

<script>

document.querySelector(".test-nav\_\_end-btn-end").addEventListener("click", function(){

if(confirm("Ви дійсно бажаете завершити тест?")) document.querySelector(".test-wrap").submit()

})

document.querySelector(".test-nav\_\_end-btn-prev").addEventListener("click", function(){';

if($\_GET['page'] != 1){

echo 'document.getElementById("action").value = "prev"

document.querySelector(".test-wrap").submit()';

}

echo '})

document.querySelector(".test-nav\_\_end-btn-next").addEventListener("click", function(){';

if($\_GET['page'] != $pages){

echo 'document.getElementById("action").value = "next"

document.querySelector(".test-wrap").submit()';

}

echo '})

</script>

';

echo '

<script>

let timer\_show = document.getElementById("timer");

function diffSubtract(date1, date2) {

return date2 - date1;

}

let end\_date = {

"full\_year": "'.substr($studenttest[0][4], 0, 4).'",

"month": "'.substr($studenttest[0][4], 5, 2).'",

"day": "'.substr($studenttest[0][4], 8, 2).'",

"hours": "'.$h.'",

"minutes": "'.$m.'",

"seconds": "'.substr($studenttest[0][4], 17, 2).'"

}

let end\_date\_str = `${end\_date.full\_year}-${end\_date.month}-${end\_date.day}T${end\_date.hours}:${end\_date.minutes}:${end\_date.seconds}`;

timer = setInterval(function () {

let now = new Date();

let date = new Date(end\_date\_str);

let ms\_left = diffSubtract(now, date);

if (ms\_left <= 0) {

clearInterval(timer);

document.querySelector(".test-wrap").submit();

} else {

let res = new Date(ms\_left);

let str\_timer = `${res.getUTCMinutes()}:${res.getUTCSeconds()}`;

timer\_show.innerHTML = str\_timer;

}

}, 1000)

</script>

';

}

else{

echo 'Error 404';

}

?>

</main>

<?php require\_once('../blocks/\_footer.php'); ?>

</body>

</html>

# ДОДАТОК Б ЛІСТИНГ КОДУ ДЛЯ РЕАЛІЗАЦІЇ ПЕРЕГЛЯДУ НАВЧАЛЬНИХ МАТЕРІАЛІВ

<?php

session\_start();

if(!isset($\_SESSION["userType"])){

header("Location: ../");

die;

}

if($\_GET['view'] == 'test'){

$query = mysqli\_query($db, "SELECT \* FROM test WHERE id = $\_GET[id]");

$test = mysqli\_fetch\_all($query);

echo '<section class="test-wrap">

<div class="container">

<h3 class="homework-title">

'.$test[0][1].'

</h3>

<div class="homework-nav">';

require('../blocks/\_nav.php');

echo '</div>

<div class="test-info">

<p class="test-info\_\_text">

Тест доступний з <span>'.$test[0][7].'</span>

</p>

<p class="test-info\_\_text">

Тест завершується в <span>'.$test[0][8].'</span>

</p>

<p class="test-info\_\_text">

Кількість часу на тест <span>'.$test[0][9].' хвилин(a)</span>

</p>';

if($\_SESSION['userType'] == 'student') {

$query = mysqli\_query($db, "SELECT \* FROM studenttest WHERE id\_test = $\_GET[id] AND id\_student = $\_SESSION[userID]");

if(mysqli\_num\_rows($query) == 1){

$result = mysqli\_fetch\_all($query);

if($result[0][5] == ''){

echo '<a href="test.php?id='.$\_GET['id'].'&page=1" target="\_blank" class="button">Продовжити тест</a>';

}

else{

echo '<p class="test-info\_\_text">

Оцінка: <span>'.$result[0][3].'</span>

</p>';

echo '<a href="test.php?result='.$result[0][0].'" class="button">Переглянути тест</a>';

}

}

else echo '<a href="test.php?id='.$\_GET['id'].'&page=1" target="\_blank" class="button">Розпочати тест</a>';

}

echo '</div>';

echo '</div>

</section>';

}

elseif($\_GET['view'] == 'task'){

if($\_SESSION['userType'] == 'student') {

$query = mysqli\_query($db, "SELECT \* FROM student WHERE id = $\_SESSION[userID]");

$student = mysqli\_fetch\_all($query);

}

$query = mysqli\_query($db, "SELECT \* FROM teacher WHERE id = (SELECT id\_teacher FROM subjectteacher WHERE id\_subject = $\_GET[c])");

$teacher = mysqli\_fetch\_all($query);

$query = mysqli\_query($db, "SELECT \* FROM task WHERE id = $\_GET[id]");

$task = mysqli\_fetch\_all($query);

$text = explode('<br />', $task[0][2]);

$query = mysqli\_query($db, "SELECT \* FROM taskresult WHERE id\_task = $\_GET[id] AND id\_student = $\_SESSION[userID]");

if(mysqli\_num\_rows($query) == 1) $taskresult = mysqli\_fetch\_all($query);

echo '

<section class="homework">

<div class="container">

<h3 class="homework-title">

'.$task[0][1].'

</h3>

<div class="homework-nav">';

require('../blocks/\_nav.php');

echo '</div>

<div class="homework-description-wrap">';

for($k = 0; isset($text[$k]); $k++){

echo '<p class="homework-description">

'.$text[$k].'

</p>';

}

echo '

</div>

<div class="homework-parameters">';

if($\_SESSION['userType'] == 'student') {

echo '<div class="homework-flex">

<p>

Учень:

</p>

<p class="homework-author">

'.$student[0][2].' '.$student[0][1].'

</p>

</div>';}

echo '<div class="homework-flex">

<p>

Вчитель:

</p>

<p class="homework-checker">

'.$teacher[0][1].' '.$teacher[0][2].' '.$teacher[0][3].'

</p>

</div>';

if($\_SESSION['userType'] == 'student') {

echo '<div class="homework-flex">

<p class="homework-status">

Статус:

</p>

<p >';

if(isset($taskresult)){

if($taskresult[0][6] != '') echo 'Оцінене';

else echo 'На перевірці';

} else echo 'Потребує сдачі';

echo '</p>

</div>

<div class="homework-flex">

<p>

Оцінка:

</p>

<p class="homework-grade">';

if(isset($taskresult) && $taskresult[0][6] != '') echo $taskresult[0][6];

echo '</p>

</div>

<div class="homework-flex">

<p>

Коментарій:

</p>

<p class="homework-comment">';

if(isset($taskresult) && $taskresult[0][5] != '') echo $taskresult[0][5];

echo '</p>

</div>';}

echo '<div class="homework-flex">

<p>

Дата:

</p>

<p class="homework-date">

'.$task[0][5].'

</p>

</div>';

if($\_SESSION['userType'] == 'student') {

echo '<div class="homework-flex">

<p>

Відповідь:

</p>

<p class="homework-answer">';

if(isset($taskresult) && $taskresult[0][4] != '') echo $taskresult[0][4];

echo '</p>

</div>

<div class="homework-flex">

<p>

Файл:

</p>

<p class="homework-file">';

if(isset($taskresult) && $taskresult[0][2] != '') echo '<a href="files/'.$\_SESSION['userID'].'-'.$\_GET['id'].'-taskresult-'.$taskresult[0][2].'" download="'.$taskresult[0][2].'">'.$taskresult[0][2].'</a>';

echo '</p>

</div>

</div>';}

if(!isset($taskresult) && ($\_SESSION['userType'] == 'student')){

echo '

<form action="../course/query.php" method="POST" class="homework-add\_\_form" enctype="multipart/form-data">

<input name="task" value="'.$\_GET['id'].'" class="d-none">

<div class="homework-add\_\_form-input">

<label>

Ваш коментар:

</label>

<textarea name="studentcomment" id="" cols="60" rows="10" class="homework-add\_\_form-input-item"></textarea >

</div>

<div class="homework-add\_\_form-input">

<label>

Наданий файл:

</label>

<input type="file" name="file" class="homework-add\_\_form-input-item homework-add\_\_form-input-item-file">

</div>

<button type="Sybmit" class="button">

Додати

</button>

</form>

';

}

echo '</div>

</section>

';

}

elseif($\_GET['view'] == 'note'){

$query = mysqli\_query($db, "SELECT \* FROM note WHERE id = $\_GET[id]");

$note = mysqli\_fetch\_all($query);

$text = explode('<br />', $note[0][2]);

echo '

<section class="homework">

<div class="container">

<h3 class="homework-title">

'.$note[0][1].'

</h3>

<div class="homework-nav">';

require('../blocks/\_nav.php');

echo '</div>

<div class="homework-description-wrap">';

for($k = 0; isset($text[$k]); $k++){

echo '<p class="note-text">

'.$text[$k].'

</p>';

}

echo '</div>

</div>

</section>

';

}

elseif($\_GET['view'] == 'grades'){

if($\_SESSION["userType"] == 'student'){

$query = mysqli\_query($db, "SELECT \* FROM student WHERE id = $\_SESSION[userID]");

$student = mysqli\_fetch\_all($query);

$query = mysqli\_query($db, "SELECT id\_subject FROM subjectclass WHERE id\_class = {$student[0][4]}");

$id\_subject = mysqli\_fetch\_all($query);

echo '

<section class="test-wrap">

<div class="container">

<h3 class="homework-title">

Оцінки

</h3>

<div class="test-table-wrap">

<table class="grades-table">

<tr>

<td class="description\_\_table-name">Предмет</td>

<td class="description\_\_table-name">

Перший семестр

</td>

<td class="description\_\_table-name">

Другий семестр

</td>

<td class="description\_\_table-name">

Річна оцінка

</td>

</tr>';

for($i = 0; isset($id\_subject[$i]); $i++){

$query = mysqli\_query($db, "SELECT \* FROM subject WHERE id = {$id\_subject[$i][0]}");

$subject = mysqli\_fetch\_all($query);

$query = mysqli\_query($db, "SELECT \* FROM grade WHERE id\_student = {$\_SESSION['userID']} AND id\_subject = {$id\_subject[$i][0]} ORDER BY topic");

$grade = mysqli\_fetch\_all($query);

echo '

<tr>

<td class="description\_\_table-value">'.$subject[0][1].'</td>

<td class="description\_\_table-value"><input type="text" value="'.$grade[0][4].'" readonly></td>

<td class="description\_\_table-value"><input type="text" value="'.$grade[1][4].'" readonly></td>

<td class="description\_\_table-value"><input type="text" value="'.$grade[2][4].'" readonly></td>

</tr>

';

}

echo '

</table>

</div>

</div>

</section>

';

}

else echo 'Error 404';

}

?>