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

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

Виконала: студентка\_6\_ курсу, групи \_601-ТН\_\_

спеціальності

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

(шифр і назва напряму)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ломанченко А. 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»**

**Студентки групи 601-ТН Ломанченко Анастасії Сергіївни**

Керівник роботи

кандидат технічних наук,

доцент Двірна О.А.

Завідувач кафедри

кандидат технічних наук,

доцент Головко Г.В.

Полтава – 2022

**РЕФЕРАТ**

Кваліфікаційна робота магістра: 75с., 31 малюнків, 1 таблиці, 2 додатки, 25 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 вчителя.

**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: 75p., 31 pictures, 1 tables, 2 appendices, 25 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 teacher.

**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](#_Toc121956603)

[ВСТУП 7](#_Toc121956604)

[РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 9](#_Toc121956605)

[1.1 Сучасні тенденції розвитку та застосування електронного навчання 9](#_Toc121956606)

[1.2 Сильні та слабкі сторони електронного навчання 13](#_Toc121956607)

[1.3 Огляд інформаційних систем для електронного навчання 17](#_Toc121956608)

[1.4 Штучний інтелект в електронному навчанні 19](#_Toc121956609)

[РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ 22](#_Toc121956610)

[2.1 Моделювання інформаційної системи електронного навчання 22](#_Toc121956611)

[2.2 Функціонал та варіанти використання системи 23](#_Toc121956612)

[2.3 Проектування бази даних 31](#_Toc121956613)

[РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ 34](#_Toc121956614)

[3.1. Вибір засобів розробки 34](#_Toc121956615)

[3.2. Розробка бази даних 36](#_Toc121956616)

[3.3 Розробка користувацького інтерфейсу 44](#_Toc121956617)

[РОЗДІЛ 4 ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ 55](#_Toc121956618)

[4.1. Вибір способу тестування 55](#_Toc121956619)

[4.2. Тест план 56](#_Toc121956620)

[ВИСНОВОК 61](#_Toc121956621)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 62](#_Toc121956622)

[ДОДАТОК А ЛІСТИНГ КОДУ ДЛЯ РЕАЛІЗАЦІЇ РЕДАГУВАННЯ ТЕСТІВ 65](#_Toc121956623)

[ДОДАТОК Б ЛІСТИНГ КОДУ ДЛЯ РЕАЛІЗАЦІЇ СТОРІНКИ РЕДАГУВАННЯ КУРСУ 72](#_Toc121956624)

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

**ІТ** – інформаційні технології.

**ІС** – інформаційна система.

**БД** – база даних.

**СУБД** – система управління базами даних.

**SQL** – Structured Query Language.

**HTTP** – HyperText Transfer Protocol.

**UML** – Unified Modeling Language.

**AJAX** – Asynchronous JavaScript And XML.

**HTML** – HyperText Markup Language.

**XML** – Extensible Markup Language.

**SCORM** – Sharable Content Object Reference.

**МБ** – мегабайт.

**IDEF0** – Integration Definition for Function Modeling.

**PHP** – Hypertext Preprocessor.

**IMS** – IP Multimedia Subsystem.

**CSV** – Comma-Separated Values.

**GIF** – Graphics Interchange Format.

**CSS** – Cascading Style Sheets.

**URL** – Uniform Resource Locator.

**JSON** – JavaScript Object Notation.

# ВСТУП

У наш час дуже багато аспектів людського життя починають швидко переходити в електронну та вебформу, чим значно пришвидшують темпи розвитку інформаційного суспільства, не стала виключенням й сфера освіти. Інформатизація освіти є одним з вагомих факторів, що сприяють модернізації навчальної системи та відкриває багато нових перспектив для збільшення результативності освітнього процесу.

Розвиток мережі Інтернет та швидке зростання її комунікаційної та інформаційних можливостей сприяє стрімкому розвитку електронної форми навчання. Саме тому пріоритетним напрямком програми модернізації освітньої діяльності України обрано електронне навчання.

Телекомунікаційні технології, що існують у сьогоденні, дають змогу значно підвищити індивідуалізацію навчання та надати постійний зворотний зв’язок між учнем та вчителем, шляхом широкого впровадження електронної форми навчання.

До основних переваг електронного навчання належить:

* новітні освітні технології;
* доступність для кожного;
* відсутність необхідності постійного відвідування школи;
* постійний зв’язок учня з вчителя;
* індивідуалізація процесу навчання.

В той самий час, аби надати необхідний рівень електронного навчального процесу, однією з найважливіших задач є проектування нових інформаційних систем, що можуть допомогти вчителям використовувати у освітній діяльності різні форми комунікації з учнями. Тому мета даної кваліфікаційної роботи, а саме, розробка інтелектуальної інформаційної системи електронного навчання є дуже актуальною задачею.

Результати роботи можуть надалі застосовуватись загальноосвітніми навчальними закладами в дистанційній та змішаній формі освіти, для забезпечення комунікації викладача та учня, перевірки завдань та створення звітів про успішність класів.

Поки що дистанційне навчання прокладає собі дорогу більше у вигляді окремих проектів, в більшості його розвитку сприяють розвинені навчальні заклади, а обмін інформацією, як правило, відбувається лише на спеціалізованих семінарах і конференціях. Такий стан речей також є однією з причин того, що e-learning досі не є достатньо відомим терміном у нашій країні, більше того, його визначають дуже різноманітно, часто розпливчасто та заплутано.

Динамічний розвиток та постійні експерименти у сфері електронного навчання також висувають нагальну потребу у створенні спеціальних інструментів оцінювання, які сприяли б оцінюванню якості та ефективності окремих електронних курсів та галузей навчання на їх основі, що викликає велику потребу у створенні інформаційних систем електронного навчання.

Інтелектуальна інформаційна система електронної освіти дозволяє організувати найкомфортніший формат навчання, полегшити доступ до навчальних матеріалів та мінімізувати витрати на організацію навчального процесу.

Темою дипломної роботи було обрано розробку функціоналу вчителя інтелектуальної інформаційної системи електронного навчання.

# РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## **1.1 Сучасні тенденції розвитку та застосування електронного навчання**

Електронне навчання – це таке навчання, що своїм фундаментом має принцип відкритості та широко використовує новітні комп’ютерні програми освітнього спрямування. При подібній формі навчання є обов’язковим створення інформаційного освітнього середовища на базі сучасних телекомунікацій, що використовується для постачання навчального матеріалу та організації спілкування [1].

Однією з основних рис електронного навчання є те, що викладач є віддалений від учнів у просторі і часі та підтримує діалог з ними за допомогою засобів комунікації. Такий тип навчання може дозволити зменшити кількість класних занять та звільнити час для самоосвіти учнів, що забезпечує індивідуалізацію навчання [2].

Електронна форма навчання забезпечує більшу самостійність студентів, зокрема самостійність пошуку та аналізу інформації, самоорганізацію, систематизації, самоконтроль. Така система освіти також передбачає впровадження нових моделей навчання, зокрема: проведення конференцій, проектної роботи та інших видів роботи з комп’ютерними технологіями.

Електронне навчання має значущі переваги у порівнянні з іншими формами навчання, однією з них є можливість підтримувати регулярний зв’язок з учителем, знаходячись у будь-якому місці, за допомогою інформаційних технологій. Використовуючи таку електронну форму навчання учень також може отримувати структурований навчальний матеріал в електронному вигляді, що є зручним, сучасним та універсальним способом подання інформації [3].

У пошуках причини провалу багатьох ініціатив, які просувають повну електронну освіту, стало зрозуміло, що проблема полягала, серед іншого, в концепції освіти, яка була спрощена до простої передачі знань у формі створення доступних інформаційних ресурси (навчання на основі ресурсів), тоді як успішні проекти розуміють освіту як конструктивний процес побудови знань. Таким чином, розвиток частково повертається до початку, і акцент робиться не на технологічному аспекті, а навпаки, на педагогічному. Зараз дискусії зосереджуються більше на методологічних і дидактичних аспектах електронного навчання, особливо на його якості. Йдеться вже не про засоби, за допомогою яких відбуватиметься навчання, а про те, яку роль у цьому вихованні виконуватимуть окремі суб’єкти, акцент також переноситься на цілі та зміст навчання, його методи і форми.

При створенні курсів тенденції звертаються до створення простих, універсальних компонентів, відповідних стандартам, з яких згодом складаються електронні курси, які є загальнозастосовними та загальнодоступними. При складанні курсів акцент робиться на підході співпраці та партнерства. У сфері технологій помітна тенденція до використання стандартизованих систем та уніфікації середовища електронної освіти в окремих установах, однак існує дуже сильний нахил до систем з відкритим вихідним кодом, які підтримують широке співтовариство користувачів.

Пріоритетним напрямком у модернізації освіти в Україні є електронне навчання, оскільки в сьогоденні існує велика потреба в отриманні освіти дистанційно, зокрема тому, що таке навчання дозволяє учням швидше освоїти матеріал, а саме навчання зробити більш доступним, наприклад для людей з особливими потребами [4].

Головною проблемою впровадження новітніх форм навчання є вибір найоптимальнішого співвідношення інструментарію інформаційно-комунікаційних технологій, традиційної освіти та інновацій в педагогіці.

Електронного навчання набуло поширення у багатьох країнах світу і його популярність щороку зростає ще більше. Першою країною, у якій дистанційне навчання почало набувати популярності, була Велика Британія.

Мотивація для створення онлайн-курсів випливає насамперед із прагнення підвищити якість навчального процесу та розширити пропозицію навчання за прийнятною ціною. Для того, щоб електронне навчання відповідало цим ідеям, воно має бути інтегроване в довгострокову стратегію, а його впровадження має стати справою всієї навчальної установи.

Електронне навчання представляє нове середовище, яке дає багато раніше безпрецедентних можливостей, але, з іншого боку, також підвищує вимоги до студентів. Перш за все, необхідно, щоб студент мав доступ до комп’ютера та Інтернету та достатню мотивацію для активної участі в цьому нетрадиційному процесі, яким, безперечно, є електронне навчання. Необхідно, щоб він був готовий до роботи з інформаційно-комунікаційними технологіями, а не стриманий спочатку своїм негативним ставленням до комп’ютерних технологій або простою невмінням оволодіти елементарним керуванням технічними засобами, за допомогою яких здійснюється навчання.

Студент повинен, якщо можливо, підключатися до онлайн-курсу регулярно, бажано щодня, навіть якщо це лише побіжна перевірка реакції на його внески в дискусію. У цьому випадку часті короткі навчання, як правило, більш ефективні, ніж багатогодинне одноразове навчання. Підтримка онлайн-курсів дає унікальну можливість для такого постійного навчання. Також важливо, щоб учень був якомога активніше залучений до онлайн-спілкування, що запобігає відчуттю ізоляції.

Учитель не єдиний, хто може принести цікаві стимули та інформацію, але й учні можуть бути джерелом багатьох знань або стимулюючих ідей. Електронне навчання часто звинувачують у соціальній ізоляції, але певною мірою цей недолік можна перетворити на позитив. Студентів можна заохочувати до обговорення та представлення своїх поглядів. Асинхронне спілкування в цьому випадку пропонує більше часу для роздумів. і сформулювати запитання та відповіді [5].

Роль викладача значно відрізняється залежно від концепції електронного навчання. У разі дистанційної освіти мова йде скоріше про репетитора, у випадку змішаної форми навчання – про вчителя чи інструктора. Дуже небезпечно думати про економію часу та зусиль викладачів при впровадженні електронного навчання. Це припущення є необґрунтованим і ще ніде не застосовувалося з позитивними результатами. Навпаки, здається, що обов’язки та необхідні дії, необхідні для успішного проведення курсу у випадку застосування онлайн-викладання, чи то основного рішення, чи просто доповнення до очного навчання, дещо більше.

Подаючи знання та навички, важливо не прив’язуватися до теоретичних текстів чи простіших структур, взятих із цих текстів, а використовувати досвід практичного життя.

Серед характерних видів діяльності викладача в онлайн-курсі також є ознайомлення студентів із концепцією та змістом курсу, що також включає підготовку студентів до конкретних проблем і методів онлайн-навчання, а також чітке визначення метод оцінювання студентів. Це також підкреслює істотну роль викладача в оцінюванні онлайн-курсу. Це питання, наскільки необхідно у випадку електронного навчання розподіляти ролі тих, хто забезпечує хід курсу, між кількома людьми [6].

Теоретично можна розглядати, наприклад, керівника проекту, який досконало знає принципи електронного навчання та здатний координувати підготовку курсу та його впровадження. Крім того, можна розглянути роль розробника курсу, який працює з контентом та інтегрує його у відповідну інформаційну систему електронного навчання, одночасно тісно співпрацюючи з експертом – викладачем, а також програмістом, який розробляє більш складні мультимедійні компоненти курсу, графічний дизайнер, який розробляє привабливі матеріали, веб-майстер, який керує курсом електронного навчання на сервері та адміністративно-організаційний персонал [7].

## **Сильні та слабкі сторони електронного навчання**

Нові можливості електронного навчання проявляються не лише у сфері практичного навчання, а й пропонують широку платформу для розробки різноманітних форм навчання, особливо у сфері дистанційного та комбінованого. Електронне навчання забезпечує можливості побудувати школу будь-де, будь-коли та для будь-кого. Завдяки новим можливостям, які дає електронне навчання, незалежно від його форми, освіта стає доступною всюди, де доступні необхідні інформаційні та комунікаційні технології. При цьому динаміка поширення цих технологій наразі колосальна.

Основними перевагами електронного навчання є те, що воно дає змогу брати участь у освітньому процесі практично з будь-якого місця, тому воно дуже сприятливо впливає на доступність навчання. Студенту практично більше не потрібно переїжджати на навчання. Електронне навчання в його чистому вигляді дає можливість навчатися повністю дистанційно, а різні форми змішаного навчання дають можливість скоротити час, необхідний для очного навчання, тому учневі більше не потрібно фізично перебувати в школі кожного дня.

За своєю природою електронне навчання також відкриває простір для значної диверсифікації можливостей навчання та програм (курсів), уможливлюючи широкий профіль випускників. Архітектура систем електронного навчання дозволяє, завдяки своїй модульності, створювати курси,

майже безпосередньо адаптовані до окремих студентів. Подібну модульність можна застосувати до створення пропозиції навчальних програм, які можуть бути більш різноманітними.

Завдяки технологіям освіта набуває надзвичайної гнучкості. Спілкування, яке є набагато більш відкритим і може вести в усіх напрямках, також може бути сильною стороною. Спілкуючись за допомогою тексту, студенти менш сором’язливі, а також мають більше часу, щоб обміркувати свої спостереження, запитання та реакцію, тому ризик того, що студент не задасть важливе запитання, тому що він соромиться або не знає, як саме запитати, є зменшений.

Уся комунікація між студентами та викладачами, а також усі навчальні матеріали доступні протягом усього курсу, щоб усі студенти мали однакові умови та доступ до однакових основних матеріалів. Це виключає проблеми пропусків очних занять через хворобу чи з інших серйозних причин. Навчання у електронній формі передбачає певну здатність до самонавчання та відповідальність за власне навчання. Ці дуже корисні навички продовжують значно зміцнюватися завдяки електронному навчанню.

Багато потужних інструментів також доступні вчителям у середовищі електронного навчання. Головними перевагами з точки зору викладачів є:

* гарантується, що всі студенти матимуть доступ до основних джерел інформації, до яких вони зможуть продовжувати звертатися. Тож їм не доведеться витрачати час на повторення того, що всі вже знають або мали можливість дізнатися заздалегідь. Таким чином, відпадає необхідність викладати теоретичні основи даного питання, а місце приділяється дрібним деталям і області практичного застосування;
* для вчителя відкривається простір для індивідуальної роботи з окремими учнями, з якими він може обговорювати теми та їхні індивідуальні завдання, не затримуючи інших учнів;
* учитель також отримує велику гнучкість у часі, оскільки він більше не прив’язаний лише до певного та важко змінюваного періоду особистого навчання. викладач може сам вирішувати, скільки часу буде приділяти спілкуванню з учнями, або уникнути ситуації, коли він не зможе охопити весь очікуваний матеріал;
* завдяки електронній системі та штучного інтелекту, що використовується в ній, викладач отримує доступ до автоматично створеної статистики діяльності студентів, тож має достатній огляд результатів навчання, участі в дискусіях, успішності тестів та ретельності виконання поставлених завдань.

Електронне навчання також відкриває великі можливості у сфері підтримки навчання. По-перше, це можливість використовувати більший діапазон різних типів навчальних матеріалів. Крім текстів, графіків і зображень, можлива також робота з аудіозаписами, що корисно не тільки для слухових типів учнів або незрячих, але, перш за все, знаходить широке практичне застосування у великій кількості дисциплін. Крім того, в електронних курсах використовуються мультимедійні анімації та відеофайли або віртуальні середовища. Цей широкий спектр навчальних матеріалів дозволяє більш повно охоплювати теми курсу та може зробити саме навчання значно привабливішим для студентів. Нові види віртуальних навчальних посібників дозволяють досягти більшої ілюстративності та повчальності, що сприяє значно глибшому проникненню в явища досліджуваних тем.

Недоліки та слабкі сторони електронного навчання. Електронне навчання зустрічає великий ступінь скептицизму, коли воно застосовується до певних галузей навчання. Електронне навчання важко застосувати в галузях, де навчання вже базується на практичних ситуаціях і вправах, наприклад у медицині чи практичних сферах мистецтва. Ці галузі також включають переважну більшість галузей університетського навчання, які ще не викладаються в повністю електронному форматі навчання ніде в світі. Бар’єри для розширення електронного навчання у вищій освіті знаходяться також у сфері доступності інформаційно-комунікаційних технологій. Незважаючи на дуже позитивний і динамічний розвиток, це все ще дуже проблематично, особливо в менш розвинених і бідніших країнах. Електронне навчання дає змогу зробити навчальні програми престижних університетів доступними для студентів у будь-якій частині світу, але на практичному рівні воно стикається з такими немислимими проблемами, як неможливість підключення до Інтернету чи відсутність електропостачання. Також перешкодою є погане знання роботи з інформаційно-комунікаційними технологіями. Серйозність цієї проблеми, яка через очікувані тенденції в освіті віщує критичне поглиблення розриву між розвиненим і бідним світом, усвідомлюють і впливові міжнародні організації, які виступають із проектами, які б зробили інформаційно-комунікаційні технології доступними для студентів у всьому світі.

У випадку електронного навчання найчастіше обговорюється аспект соціального контакту по відношенню до викладача та інших студентів. Дуже проблематично визначити, наскільки інтенсивність соціальних відносин впливає на сам процес виховання. Очевидно, що електронне навчання більше орієнтоване на індивідуалізацію навчання, але не можна стверджувати, що воно призведе до соціальної ізоляції студентів. Електронне навчання пропонує широкий спектр інструментів спілкування та часто дає набагато більше простору для встановлення контактів, ніж обмежене в часі та часто дуже формальне середовище самонавчання. Тим не менш, загалом очікуються більші проблеми з набуттям соціально-інтерактивних та інших професійних навичок у результаті прямого особистого контакту, який є незамінним технологіями.

Електронне навчання може не бути ефективним для погано мотивованих студентів або тих, хто не здатний до самостійної роботи, самооцінки та самодисципліни. Мотивація є дуже серйозною проблемою при впровадженні електронного навчання на практиці. Очевидно, що якщо вчитель не робить суттєвого акценту на ролі навчання, недостатньо стимулює, заохочує та підтримує студентів, то відсоток тих, хто втрачає мотивацію до навчання, є високим.

Електронне навчання також пред'являє відносно високі вимоги до матеріального оснащення та інформаційної та комп'ютерної грамотності студентів. Особливо складніші навчальні програми можуть значно перевищувати вимоги стандартних домашніх комп'ютерів і нормальний рівень програмного керування.

Недоліки щодо організації навчання Впровадження електронного навчання передбачає, перш за все, відносно великі початкові витрати, необхідні для придбання технічного обладнання, побудови середовища електронного навчання, придбання программного забезпечення і створення самих електронних курсів. Це призводить до зростання цін на освітні продукти. Велика кількість різних систем і форм електронного навчання також ускладнює сумісність навчання в різних навчальних закладах. Електронне навчання, незважаючи на всі його проблеми та недоліки, знаходить своє застосування у все більшій кількості навчальних закладів. Він може покладатися насамперед на дуже динамічний технічний розвиток, який вирішує все більше проблемних областей.

## **1.3 Огляд інформаційних систем для електронного навчання**

Існує велика кількість електронних освітніх ресурсів. Аналіз наявних систем електронної освіти дозволив виділити найпопулярніші, серед них: Moodle [8], Ilias [9], Blackboard [10].

**Moodle** є відкрите безкоштовне програмне забезпечення для дистанційного навчання австралійського виробництва, яке дає змогу користуватися великою кількістю інструментів для взаємодії вчителя з учнями. Зокрема ця платформа дозволяє представляти матеріал у різних формах, проводити тестування учнів з використанням широкого спектра питань відкритої та закритої форми, а учні можуть відправляти вчителю виконане домашнє завдання прикріпивши на ресурс відповідний файл. На сьогоднішній день платформа Moodle є найбільш розповсюдженою системою дистанційної освіти.

Навчальна платформа Moodle написана мовою програмування PHP з використанням SQL-бази даних та відповідає стандарту програмного забезпечення [SCORM](https://uk.wikipedia.org/wiki/SCORM).

Головними перевагами Moodle є:

* код ресурсу знаходиться у відкритому доступі. Це дає змогу повністю налаштувати функціонал та дизайн сайту під конкретні потреби та побажання;
* платформа надає великий вибір способів додавання користувачів. Користувачі мають змогу зареєструватись у системі самотужки, адміністратор може додати користувача вручну, запросити учнів по по e-mail, або, за потреби, імпортувати групу користувачів з файлу;
* розвинута спільнота, що постійно оновлює систему та додає нові плагіни [11].

Головним мінусом системи Moodle є складність для звичайного користувача, що означає додаткові витрати на спеціального працівника щ досвідом роботи в інформаційних технологіях, який буде займатися адміністрацією ресурсу та швидким встановленням і запуском платформи навчальному закладу. Також попри великі можливості системи, недоліком є її переобтяжений функціонал, що досить сповільнює роботу з нею [12].

**Ilias** – інформаційна система електронного навчання вироблена в Німеччині, головною особливістю якої є простий та інтуїтивно зрозумілий інтерфейс, схожий на соціальну мережу, що дозволяє швидко на неї звикнути та навчитися використовувати систему викладачам та школярам.

Хоча Ilias на відміну від Moodle є дуже простим у налаштуванні і використанні, його встановлення та початок використання потрібує людину обізнану в інформаційних технологіях.

Дана платформа має вбудований конструктор контенту, за допомогою якого викладачі мають можливість створювати тести, вправи та опитування для учнів. Система дозволяє створювати та призначати навчальні матеріали.

Ilias створений за допомогою мови програмування JavaScript та використовує технології [MySQL](https://ru.wikipedia.org/wiki/MySQL), [XML](https://ru.wikipedia.org/wiki/XML), [Apache](https://ru.wikipedia.org/wiki/Apache_HTTP_Server) і [PHP](https://ru.wikipedia.org/wiki/PHP), підтримує стандарти IMS, а також SCORM 1.2 і v2004 RD3, це означає, що матеріал з системи можуть бути експортовані та використані за потреби у будь-якій іншій інформаційній системі.

Платформа може надавати звіти зі статистики у формату XML та CSV, в яких стовпці можуть бути налаштовані за допомогою спеціальних плагінів.

**Blackboard** має наступні можливості:

* використання аудіо- і відеофайлів;
* створення тексту з різними візуальними ефектами (різні шрифти: жирні, похилі, підкреслені, перекреслені символи, а також символи з межею над ним, підрядкові та надрядкові індекси та інше);
* використання GIF анімації;
* створення банку тестових завдань з можливістю генерації тестів для організації тестування на різних етапах контролю знань і умінь в електронному навчальному курсі;
* використання зображень та будь-яких OLE-об'єктів;
* швидке перемикання в режим перегляду створюваного електронного курсу в режимі учня [13].

Також великою перевагою платформи **Blackboard** є наявність вбудованого редактору формул, а також наявність чату, для надання можливості школярам спілкуватись між собою в межах групи, а також з учителями в реальному режиму часу.

Головними недоліками системи BlackBoard є:

* знижена пропускна здатність мережі під час вантаження матеріалів;
* дорожнеча системи;
* обмежений функціонал системи на певних операційних системах;
* довге вивчення користування системою [13].

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

У сьогоденні технології штучного інтелекту вже успішно використовується в дуже багатьох сферах людського життя. Хоча галузь освіти вважається одним з найбільш консервативних секторів, в якій кожна інновація зустрічається дуже прохолодно, можливості штучного інтелекту мають змогу розв’язати складні проблеми, з якими зустрічаються учні й викладачі, а також підвищити якість навчання. Тому у наш час уже існують дієві методики та технології, які активно використовуються великими освітніми центрами [14].

Електронне навчання є одним з найбільш вагоміших напрямків для впровадження штучного інтелекту в освіту.

При організації процесу освіти певної дисципліни, кожен раз виникає проблема, що у майбутніх учнів наявний різний рівень початкових знань і здібностей до освоєння навчальної програми, а також, безумовно, кожен учень має свій рівень мотивації. В результаті, кожен викладач стикається з нерозв’язним завданням: як донести найважливіше до «слабких» і при цьому дати необхідне навантаження здібним учням. Вирішення цього завдання – у впровадженні адаптивних технологій.

Штучний інтелект має стежити за успішністю кожного окремого учня і створювати для нього індивідуальний план навчання, підбираючи найбільш вдалу послідовність тем і завдань. Інший варіант – на основі аналізу навчання кожного учня інформувати викладача про ступінь засвоюваності певних блоках курсу [15].

На сьогоднішній день велика частина часу викладача витрачається на визначення індивідуальних особливостей учнів та створення відповідного, індивідуалізованого навчального контенту, що мусить відповідати навчальним планам і тому, часу на безпосередній контакт із школярами вчитель отримує замало, хоча діти, особливо ті, що навчаються в початкових класах, потребують індивідуальний підхід для більш повного розкриття своїх талантів і потенціалу. Данну задачу досить важко, а інколи навіть неможливо, виконати на потрібному рівні, не використовуючи потужні комп’ютерні та інформаційні інструменти.

Після проведення оцінювання багатогранності та складності поставленої задачі – такі інструменти повинні бути створені на базі комп’ютерних технологій з використанням інформаційних технологій та штучного інтелекту.

Штучний інтелект є такий набір функцій комп’ютерних технології, що найбільш точно відтворюють людські навички та мислення, такі, як вести осмислений та зв’язний діалог, виконувати обґрунтовані висновки, проводити аналіз складних систем та інше [16].

Штучний інтелект та машинне навчання вже не є витвором фантастики, а є невід’ємною складовою інноваційної економіки сьогодення.

Але не дивлячись на те, що штучний інтелект може сприяти персоналізованому підходу в навчанні, він також забезпечує можливість надати той навчальний план, якого учень потребує в певний час. Уже в наш час існують такі програмні продукти, що за допомогою використання штучного інтелекту, полегшують викладачам рутинну роботу на різних етапах освітнього процесу та дозволяють їм виділити більше часу на працю, яка не може бути виконана автоматично та потребує участі кваліфікованого вчителя [17].

Останнім часов набирає популярності системи штучного інтелекту під назвою «Експертні системи», які призначені для вирішення задач та аналізу задач у ріхних предметних областях. Це програми які використовують знання у певній формі та на їх основі роблять деякі висновки. Зазвичай використовуються для допомоги, а не для заміни людини.

Найчастіше експертні системи накопичують знання та об’єднують їх разом за допомогою правил та механізмом висновків. Алгоритм зчитує шлях та обробляє факти з якого в результаті робить логічний висновок. Експертні системи працюють на основі баз знань і допомагають людині краще приймати рішення в тих чи інших ситуаціях [18].

Процесом інженерії знань називають побудову та підтримку експертної системи. Він гарантує, що в системі наявно достатньо актуальних знань для вирішення поставлених проблем. Системи мають змогу розширятися з часом шляхом внесення нових наборів правил та розширення бази знань [19].

# РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ

## **2.1 Моделювання інформаційної системи електронного навчання**

На початку проектування створюємо модель функціоналу системи електронного навчання та зображуємо графічно всі процеси і контексти. А також описуємо взаємодію акторів із системою і логічні зв’язки різних частин системи.

З урахуванням нотації IDEF0 побудовано контекстну діаграму для взаємодії користувача вчителя із системою електронного навчання зображену на рисунку 2.1.

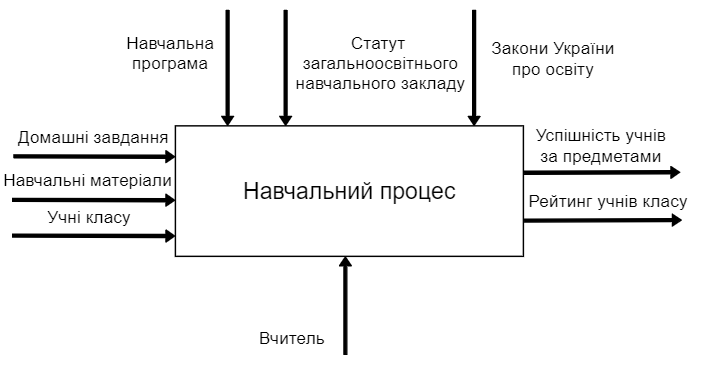


Рисунок 2.1 –Контекстна діаграма ІС електронного навчання

Наступним кроком була проведена декомпозиція функціоналу системи та побудовано відповідну діаграму зображену на рисунку 2.2.

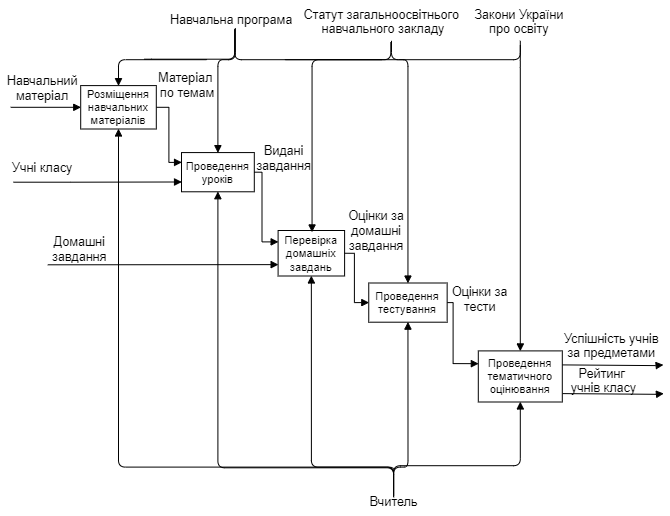


Рисунок 2.2 – Діаграма декомпозиції ІС електронного навчання

## **2.2 Функціонал та варіанти використання системи**

Найголовнішими можливостями, які повинна надавати система є зручна та проста організація обміну навчальним матеріалом між вчителем та учнем, наявність механізмів для оцінки знань, проведення контролів, швидка комунікація при виконанні завдань, а також зрозуміла та прозора система оцінювання (рис. 2.3).



Рисунок 2.3 – Загальна структура системи

Система електронного навчання має три логічні модулі: вчитель, учень та адміністратор. Адміністратор відповідає за налаштування системи та організацією інших користувачів системи (рис. 2.4).

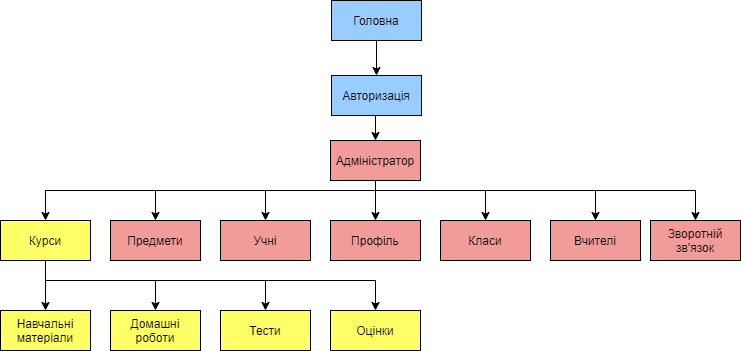


Рисунок 2.4 – Структура системи з погляду адміністратора

Логічний модуль для вчителя відповідає за надання простих та інтуїтивно зрозумілих інструментів для керування навчальним матеріалом на сторінках курсів, додавання тестів та завдань, перевірка виконаних робіт та надання коментарів, а також оцінення (рис. 2.5).

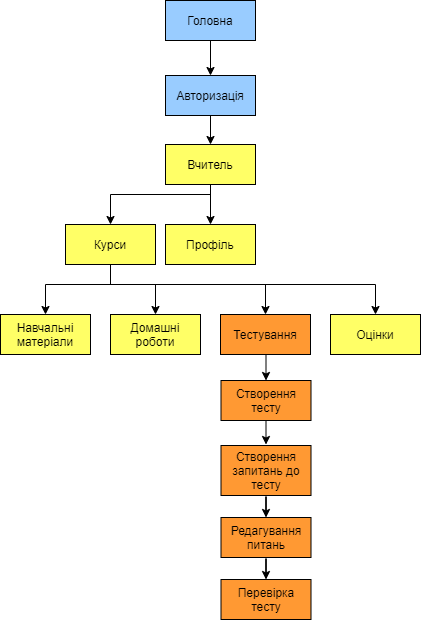


Рисунок 2.5 – Структура системи з погляду вчителя

Зазначимо найголовніші можливості користувачів у системі та варіанти иї використання (рис. 2.6).

Головними можливостями для вчителя у системі є:

* ведення сторінки курсів: додавання, зміна та видалення навчальних матеріалів, а також їх перегляд;
* оцінювання учнів;
* перевірка тестування та їх перегляд;
* оцінювання домашніх робіт та надання коментарів;
* додавання тестів та завдань.

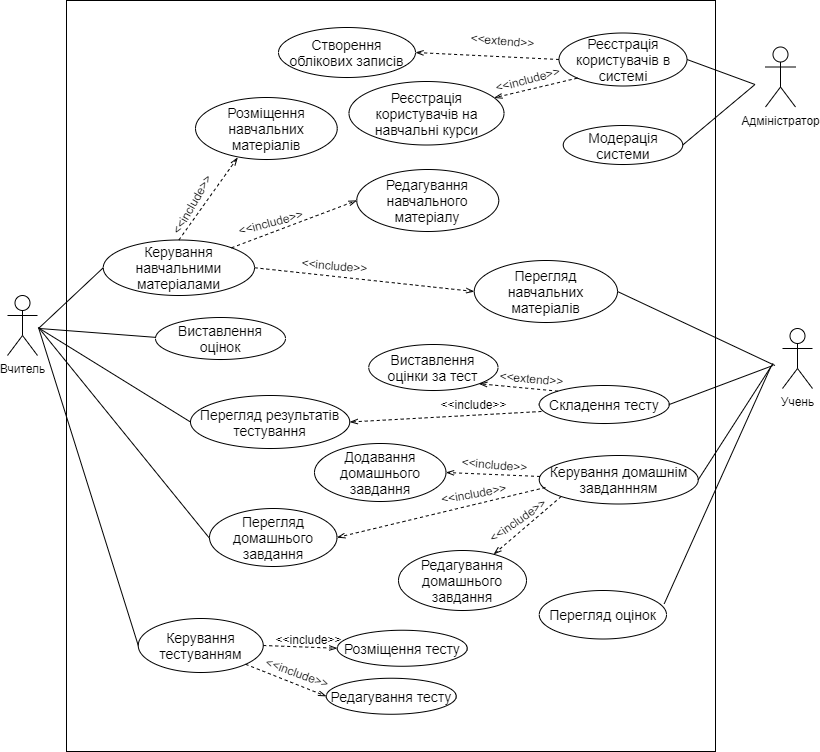
****

Рисунок 2.6 – Діаграма варіантів використання системи

При взаємодії із системою користувач, в залежності від типу, має обмежений доступ до функціоналу, як зображено на діаграмі прецедентів.

Процес завантаження навчального матеріалу на сторінку курсу вчителем зображено у деталях та впорядковано, а також розглянуто взаємодію об’єктів системи електронного навчання (рис. 2.7).

Починає цей процес вчитель, а також разом з ним у взаємодії беруть участь наступні об’єкти:

* об’єкт «Subject» (Предмет);
* об’єкт «MaterialOptions» (Параметри роботи із навчальним матеріалом), що відповідає за вибір можливої дії із матеріалом в даному прецеденті;
* об’єкт «Material» (Навчальний матеріал);
* об’єкт «Teacher» (Вчитель);
* об’єкт «AddNewMaterial» (Додавання нового навчального матеріалу), що відповідає за додавання навчального матеріалу в систему.

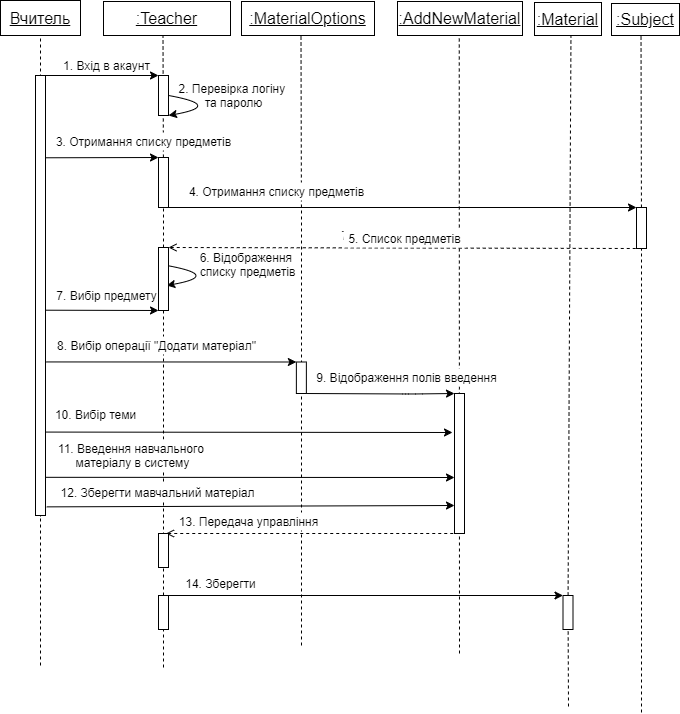


Рисунок 2.7 – Діаграма послідовностей

В ході даного сценарію вчитель авторизується у системі та обирає зі списку потрібний йому предмет. Далі натискає на відповідну кнопку для додавання нового навчального матеріалу, обирає тему в яку потрібно додати, та завантажує матеріал у систему, у вигляді файлу або у вигляді тексту. Після прикрівлення всього потрібного матеріалу вчитель зберігає його в системі.

За допомогою діаграми кооперацій на малюнку 2.8 відображено детальніше з якими елементами взяємодіє вчитель.

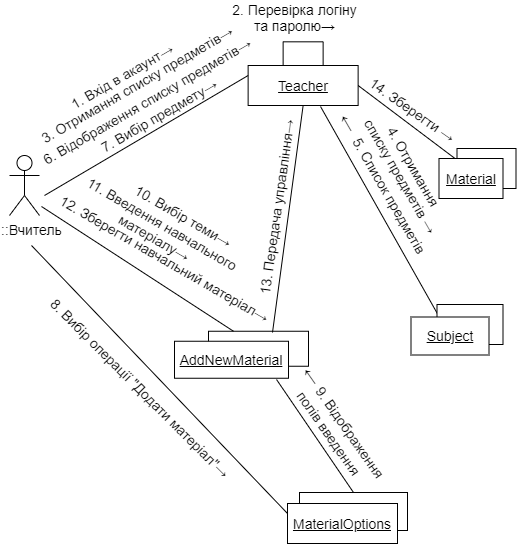


Рисунок 2.8 – Діаграма кооперації

При виконанні деяких процесів система може приймати різні стани, які змінюються під час надходження зовнішніх сигналів або імпульсів. Для користувача «Вчитель» при створенні тесту стани системи зображені на рисунку 2.9.

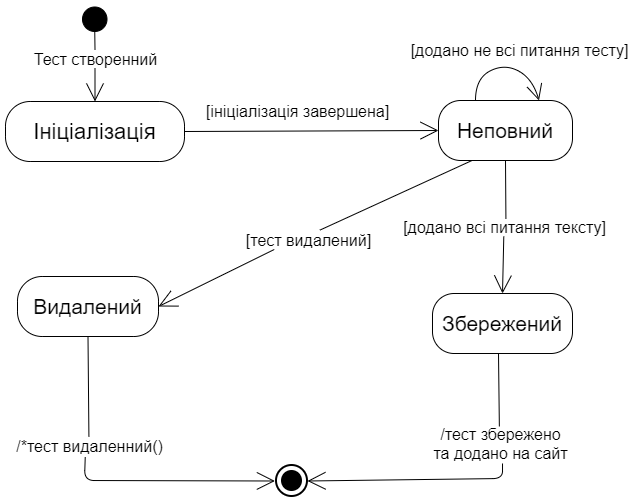


Рисунок 2.9 – Діаграма станів для користувача системи – Вчитель

На початку створення нового тесту система приймає стан «Ініціализація» та відбувається подія ініціалізації. Далі тест отримує стан «Неповний» і пока не додано всі питання знаходиться у ньому. Потім вчитель може або зберегти тест, а система отримає стан «Збережено» і після додавання тесту в систему переходить у кінцевий стан. Якщо ж вчитель вирішив видалити тест, то тистема переходить у стан «Видалений» та після видалення отримує кінцевий стан.

Базуючись на діаграмі поведіки зображеної на рисунку 2.10 створено модель за допомогою мови UML. На ній відображено процес наповнення сторінки курсу навчальним матеріалом.

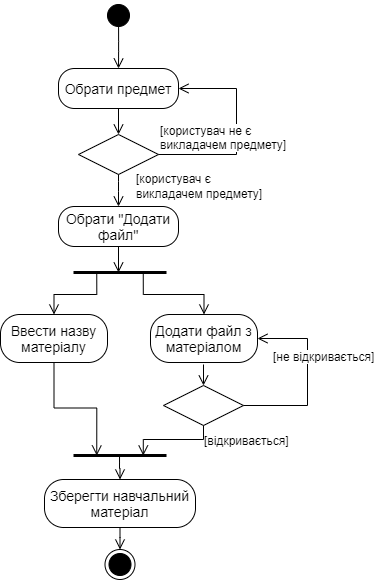


Рисунок 2.10 – Діаграма діяльності

Процес занесення нового матеріалу вчителем відображено на діаграмі діяльності. Спочатку вчитель обирає потрібний предмет та натискає «Додати файл», потім вводить назву файлу та зберігає його. Після цього система зберігає новий навчальний матеріал та відображає його на сторінці курсу, а учні можуть їм скористатися.

У випадку коли вчитель не викладає даний предмет, то система автоматично повертає користувача до списку доступних йому предметів. Також якщо файл який намагаються додати є пошкодженним, то система автоматично запропонує обрати інший.

## **2.3 Проектування бази даних**

Для зберігання інформації та даних інформаційної системи електронного навчання використовуємо базу даних. Найзручнішою в даному випадку є реляційна база даних яка складається із таблиць та зв’язків між ними. Кожна таблиця слугує для окремої сутності у системі.

Система повинна містити у собі дані про всіх користувачів, учнів та вчителів, списки всіх вивчаємих предметів, перелік класів, а також навчальні матеріали всіх сторінок курсів, тести за завдання разом з їх результатами, оцінки.

Головні правила при проектуванні схеми бази даних для системи електронного навчання:

* клас може містити багато учнів, але учень може належати лише до одного класу;
* учень може складати багато тестів і виконувати багато робіт, але результат проходження тесту або відповідь на завдання може належати лише одному конкретному учню;
* виконана робота учнем належить до одного предмета, а по кожному предмету може бути багато виконаних робіт. Аналогічні правила стосовно результатів тестування;
* контроль успішності учнім має проводитись в одному класі з одного предмету, а по кожному предмету в кожному класі може бути декілька контролів успішності.

Для інформаційної системи електронного навчання створено базу даних за допомогою СУБД MySQL та вебінтерфейсу phpMyAdmin, яка містить 19 таблиць, зображена на рисунку 2.11.

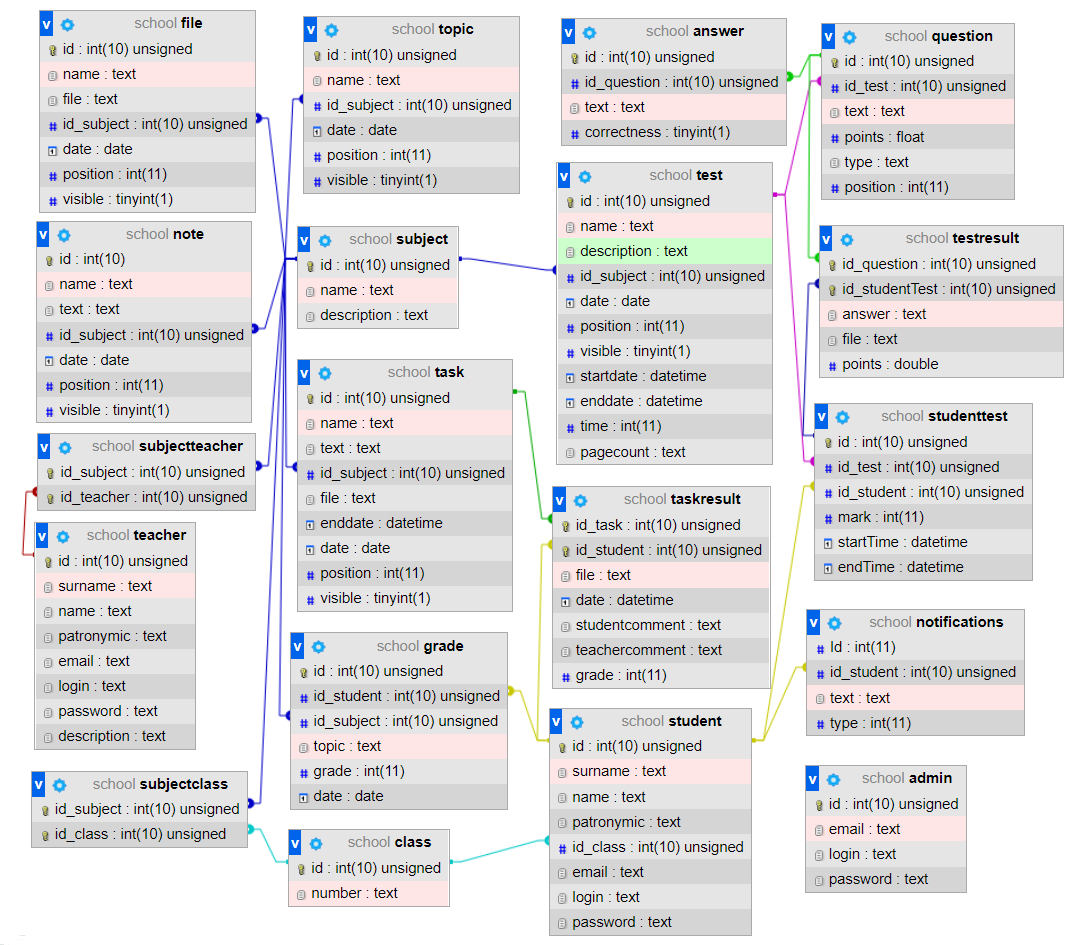


Рисунок 2.11 – Схема даних ІС

Основними таблицями системи є:

* «Клас», в якій знаходиться вся інформація про класи;
* «Предмет», в якій знаходиться вся інформація про предмети;
* «Адміністратор», в якій знаходиться обліковий запис адміністратора;
* «Вчитель», в якій знаходиться вся інформація про вчителя і його обліковий запис;
* «Учень», в якій знаходиться вся інформація про учня і його обліковий запис;
* «Файл», в якій знаходиться інформація про файл на сторінці курсів завантажені вчителем;
* «Тема», в якій знаходиться інформація про теми предмету;
* «Запис», в якій знаходиться інформація про записи додані вчителем на сторінку курсу;
* «Завдання», в якій знаходиться вся інформація про завдання для учнів;
* «Виконана робота», в якій знаходиться інформація про виконані учнем роботи;
* «Тест», в якій знаходиться вся інформація про тести;
* «Питання», в якій знаходиться вся інформація про питання тесту;
* «Відповідь», в якій знаходиться вся інформація про відповіді на питання тестів;
* «Результат тестування», в якій знаходиться інформація про результати тестування учня;
* «Оцінка», в якій знаходиться інформація про оцінки учнів;
* «Сповіщення», в якій знаходиться інформація про сповіщення користувачів системи.

# РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ

## **3.1. Вибір засобів розробки**

Інтелектуальна інформаційна система електронного навчання буде реалізовувати модель архітектури клієнт-сервер. Клієнтська частина программного продукту генерується на сервері за допомогою серверної мови програмування PHP і передається користувачеві за використовуючи HTML та CSS.

У сьогоденні мова програмування PHP є найпопулярнішою для серверної розробки. Понад 80% вебдодатків працюють на PHP.

PHP є оптимізована до швидкого створення інформаційних систем. Використання запитів GET і POST, а також робота з HTML і URL, що притаманні обраній мові програмування, значно полегшують та скорочують час розробки програмного забезпечення.

Беззаперечним плюсом PHP є його універсальність та гнучкість. Дана мова програмування є вбудовуваною, що забезпечує її сумісність з багатьма основними операційними системами: як Windows, Unix і Linux або MacOS; підтримує більшість баз даних та серверів, тому за допомогою РHP можна написати код для крос-платформного веб застосунку.

При функціонуванні додатків написаних на РНР, всі скрипти компілюються на стороні сервера, що унеможливлює виникнення проблем з браузером користувача. PHP є мовою з високою інтегрованістю та динамічністю, зміни в код можна впроваджувати на будь-якій стадії розробки без втрат у часі [21].

Завдяки універсальності мови програмування РНР її підтримують майже всі хостинг-провайдери та надають хостингові послуги для сайтів на ньому.

Для забезпечення зміни вмісту сторінки без перезавантаження, використано стек технологій Ajaxразом з JavaScript (рис. 3.1).

При використання моделі AJAX браузер створює виклик JavaScript, який потім активує XMLHttpRequest. [22].

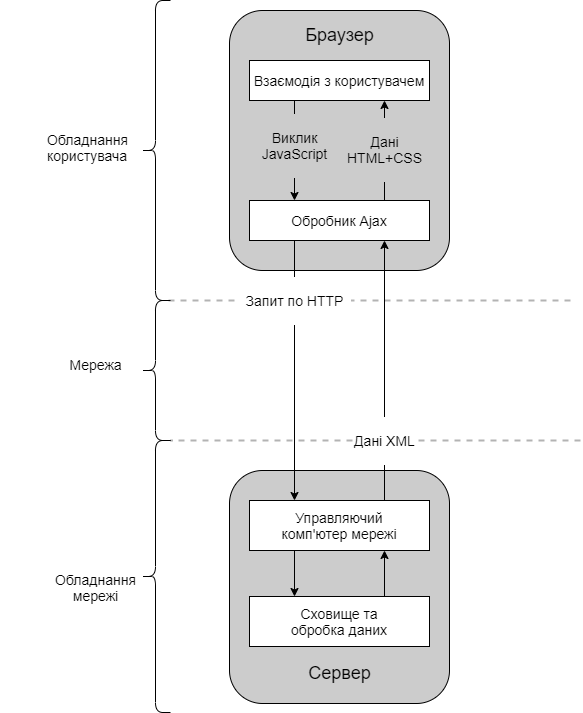


Рисунок 3.1 – Модель вебдодатків з використанням AJAX

Складовою AJAX системи є:

* XML, що використовується для обміну даними;
* HTML, що використовується для основної мови;
* CSS, що використовується для мови представлення;
* JSON, що використовується для управління даними;
* об'єкт XMLHttpRequest, що використовується для асинхронної взаємодії;
* об'єктна модель документа (DOM), що використовується для динамічного відображення даних і їх взаємодії;
* мова програмування JavaScript, що використовується для об'єднання всіх цих технологій [23].

Для роботи з базою даних було вирішено використати систему керування базами даних MySQL, що має змогу ефективно функціонувати у взаємодії з вебплатформами та сайтами. Обрана СУБД володіє великою кількістю доступних інструментів, що можуть бути використанні для створення застосунків та плагінів, які полегшують працю з базами даних.

MySQL досить безпечною та має безліч вбудованих функцій безпеки, які працюють за замовчуванням. Велика продуктивність системи забезпечується через спрощення стандартів, що використовуються в ній [24].

Для більш зручної роботи з БД буде використано phpMyAdmin інтерфейс бази даних.

Обрана панель управління вміщує в собі все необхідне для керування БД цілим MySQL сервером та окремими базами даних, а також вона може бути використана як панель управління сайтом для наповнення і редагування контенту [25].

## **3.2. Розробка бази даних**

Для функціонування системи та збереження в ній інформації була створена база даних (рис. 3.2). В ній присутні наступні сутності:

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» 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 «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 «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 «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 «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 «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 «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 «Відповідь» має поля:

* іd – унікальний ідентифікатор відповіді в системі, не може бути NULL, є первинним ключем, має тип int(10) unsigned;
* іd питання – ідентифікатор питання, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* текст – відповідь на питання, не може бути NULL, має тип text;
* 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 «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 «Сповіщення» має поля:

* іd сповіщення – унікальний ідентифікатор питання, не може бути NULL, є первинним ключем, має тип int(11);
* іd учня – ідентифікатор учня, не може бути NULL, є вторинним ключем, має тип int(10) unsigned;
* текст сповіщення – текст сповіщення для користувача, може бути NULL, має тип text;
* тип сповіщення – зберігає номер типу сповіщення, може бути NULL, має тип int(11). В системі присутні три типа сповіщень: повідомлення, нагадування та застереження, кожне з яких має свій колір.

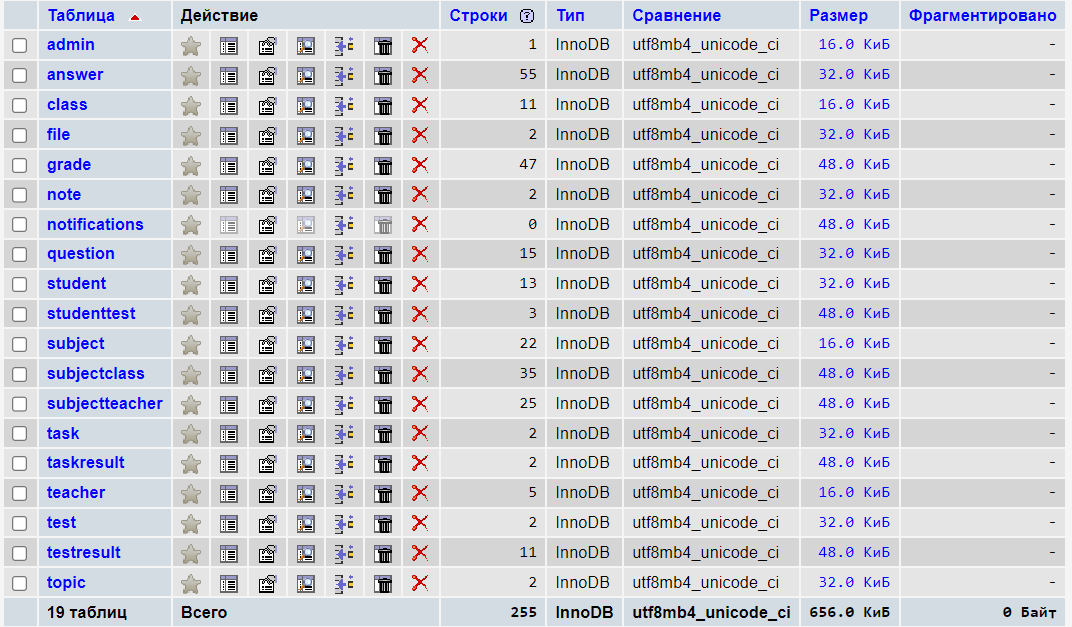


Рисунок 3.2 – Архітектура БД вебсервісу

## **3.3 Розробка користувацького інтерфейсу**

В процесі проектування дипломної роботи було розроблено програмний модуль, що реалізує функціонал вчителя. При першому вході на сайт, користувач бачить зображене на рисунку 3.3 вікно входу.

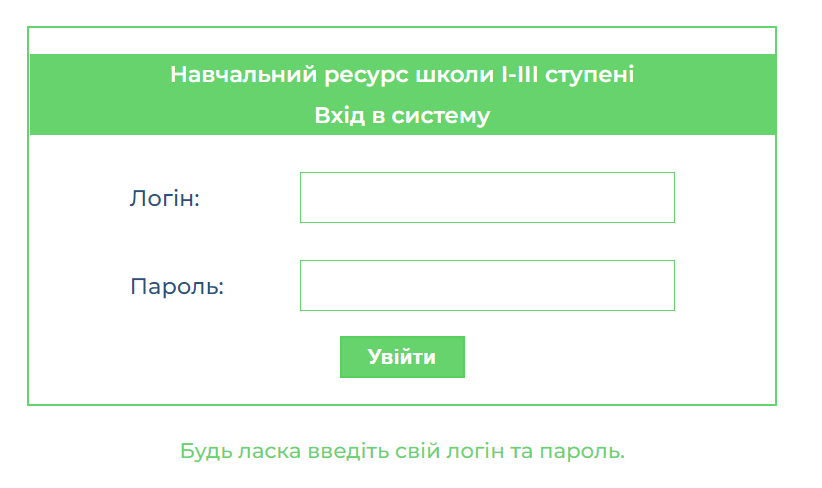


Рисунок 3.3 – Вікно входу в обліковий запис

Після вводу облікових даних користувача система звертається до бази даних та перевіряє наявність його в системі. Якщо відповідність знайдена то перенаправляє його на сторінку особистого кабінету, саме тут можна змінити деякі даті та переглянути загальну інформацію (рис. 3.4). У випадку коли логін або пароль було введено невірно система дає можливість ввести дані знову.

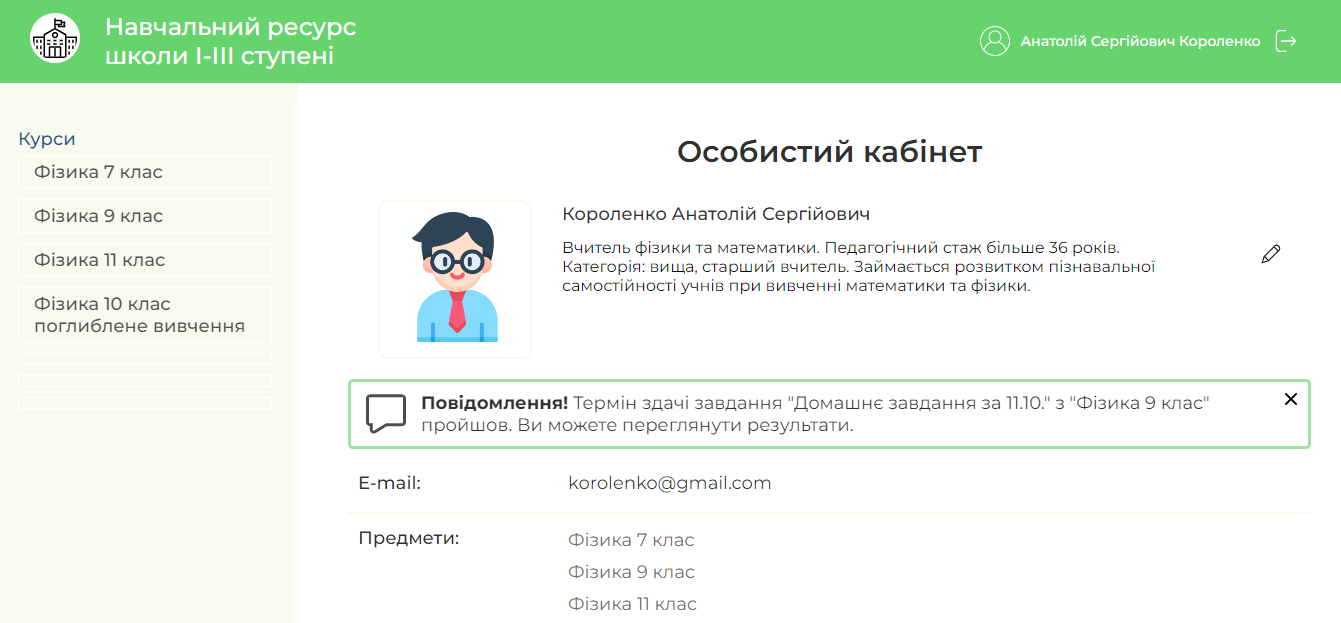


Рисунок 3.4 – Домашня сторінка вчителя

Навігаційне меню знаходиться зліва сторінки у вигляді переліку доступних предметів, натиснувши на який відкриється сторінка відповідного курсу з всім доступним функціоналом (рис. 3.5).

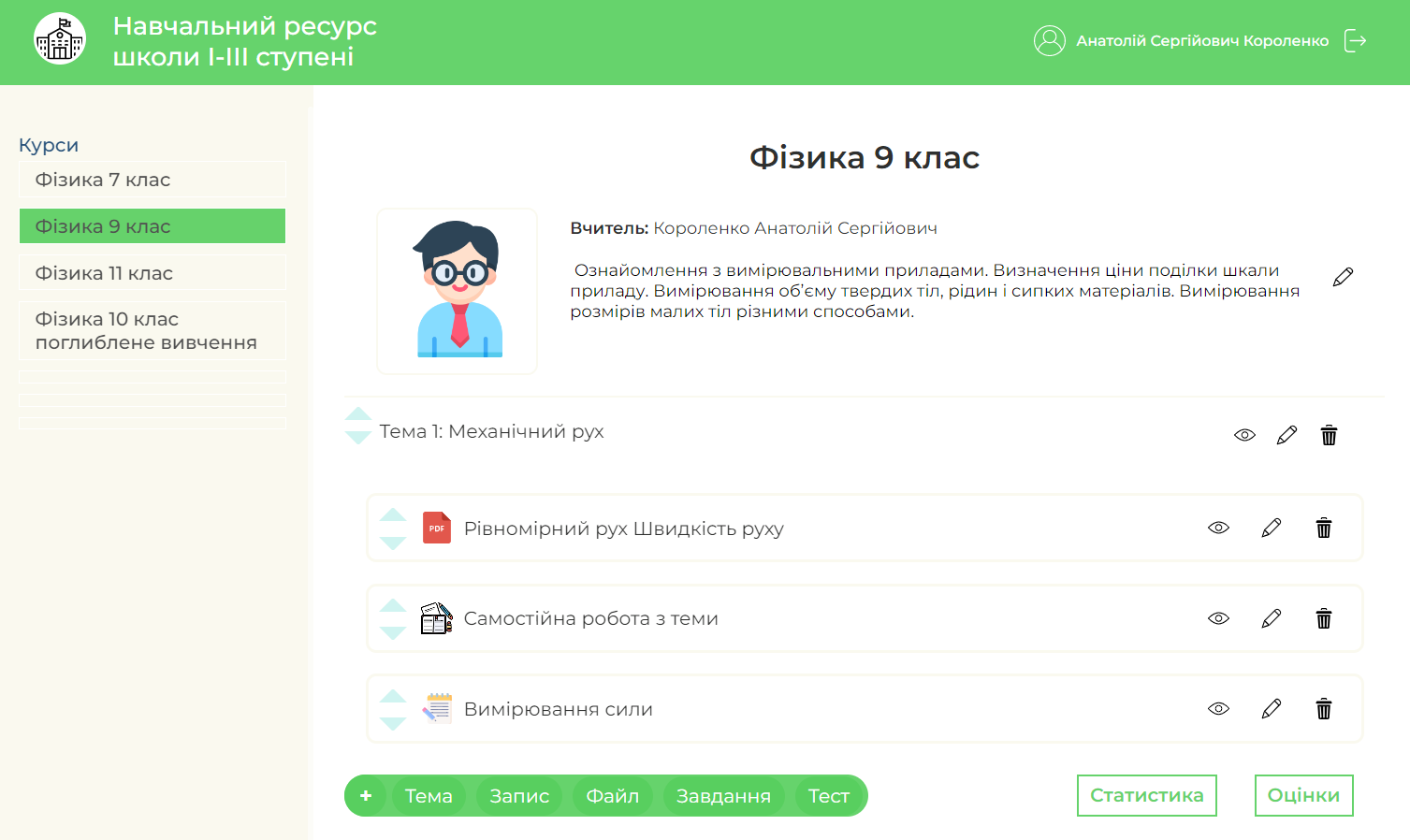


Рисунок 3.5 – Сторінка предмету

Вчитель може редагувати сторінки курсів які він викладає. Також тут він може додати нові елементи для навчання: теми, файли, записи, завдання та тести (рис. 3. 6 – 3.10). Тут же можна їх видалити або змінити порядок їх відображення, а також можливо тимчасово приховати від учнів деякі елементи.

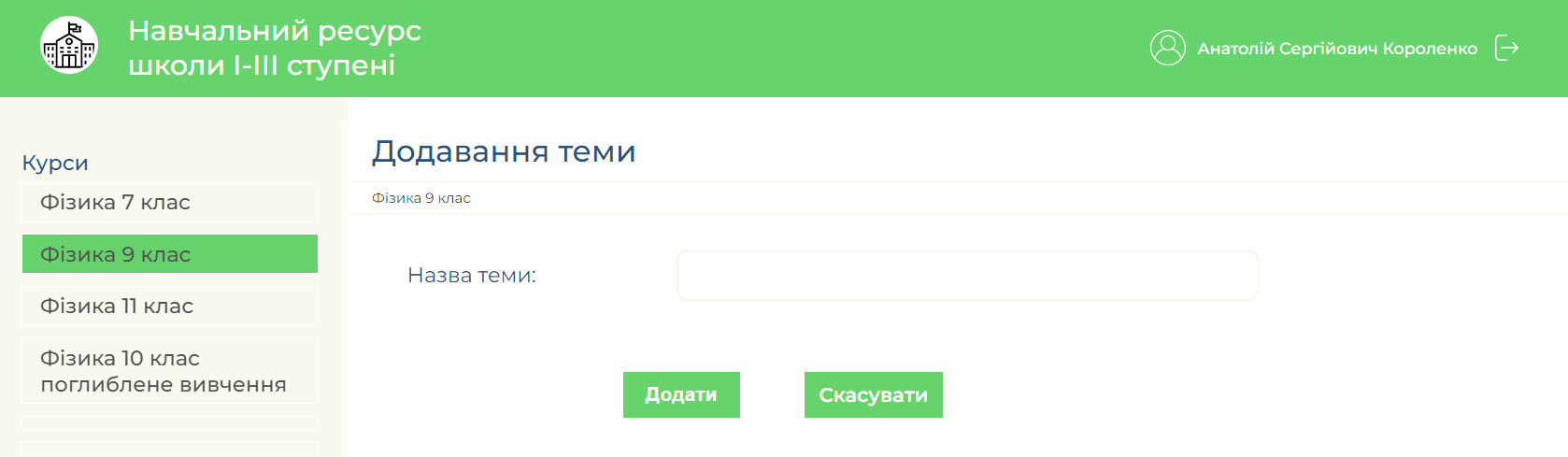


Рисунок 3.6 – Сторінка додавання теми

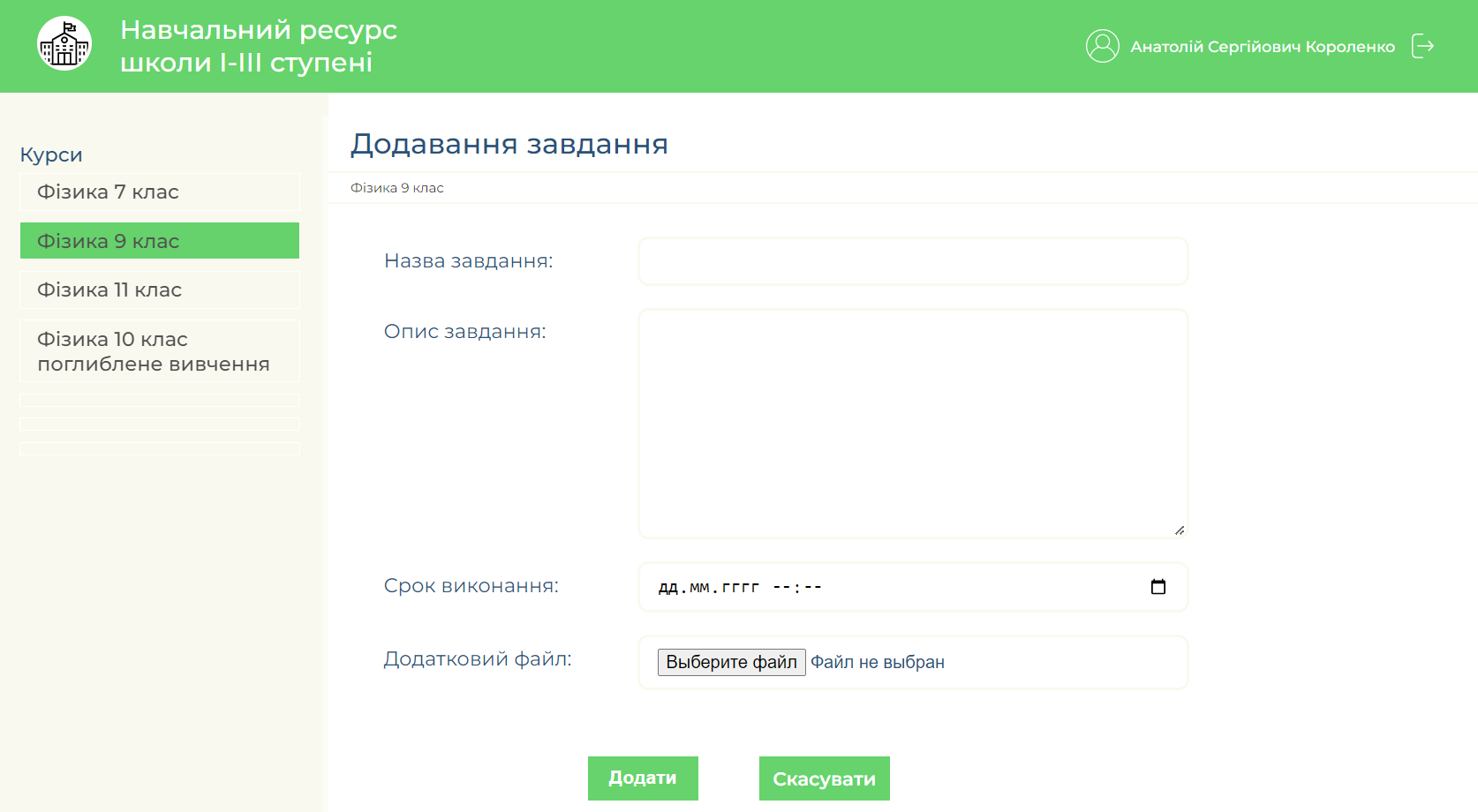


Рисунок 3.7 – Сторінка додавання завдання

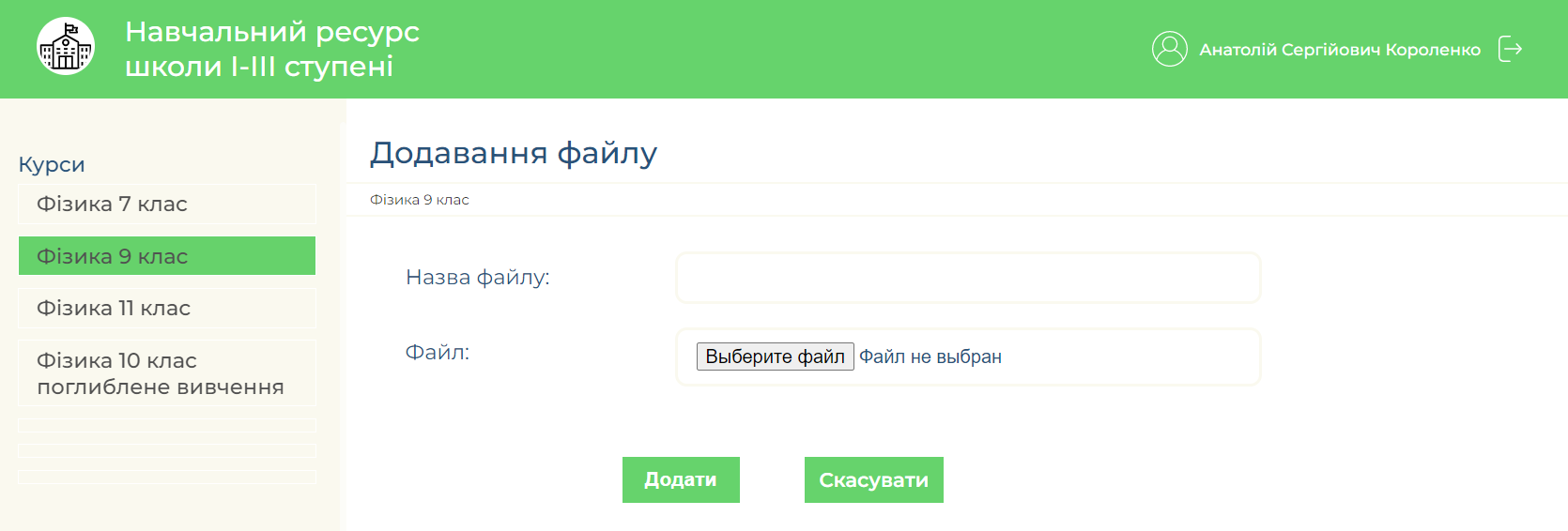


Рисунок 3.8 – Сторінка додавання файлу

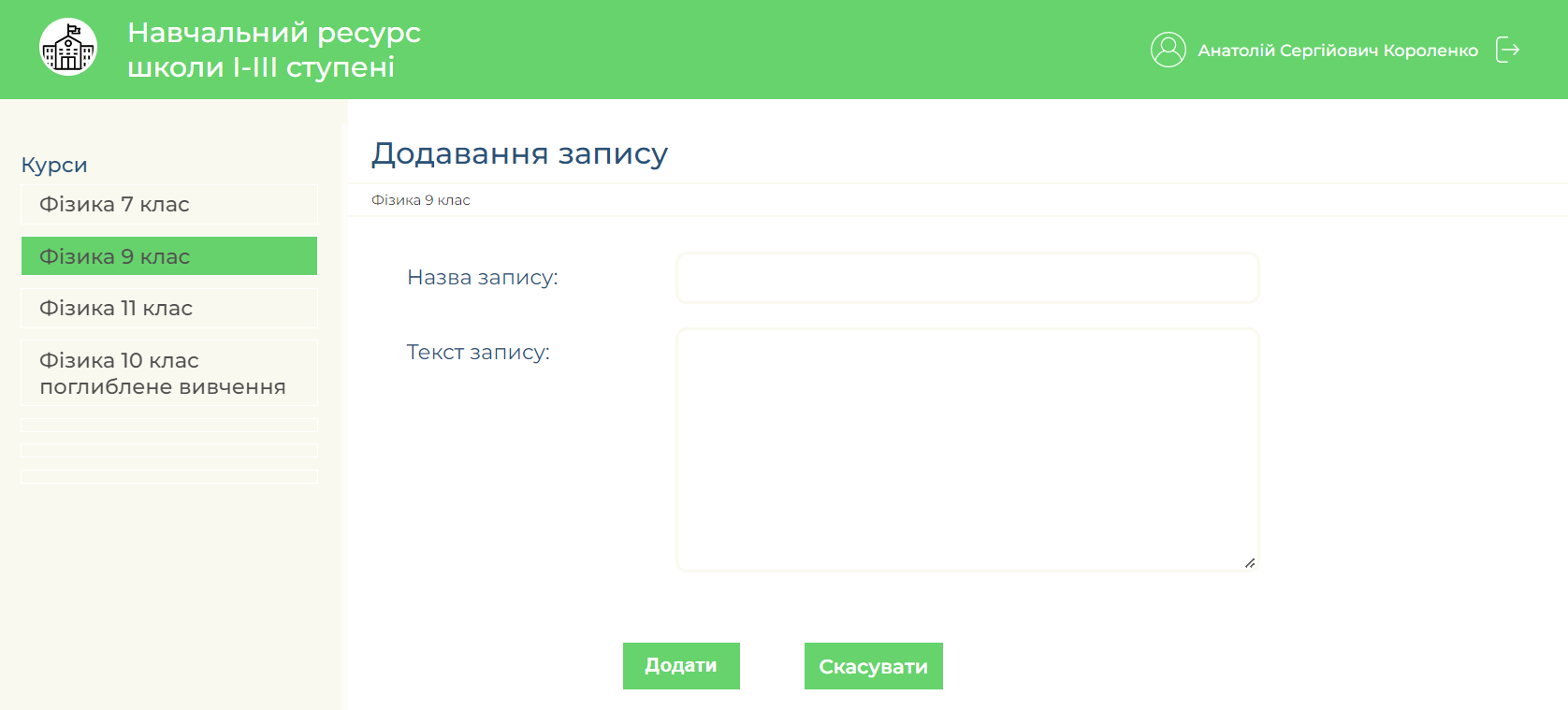


Рисунок 3.9 – Сторінка додавання запису

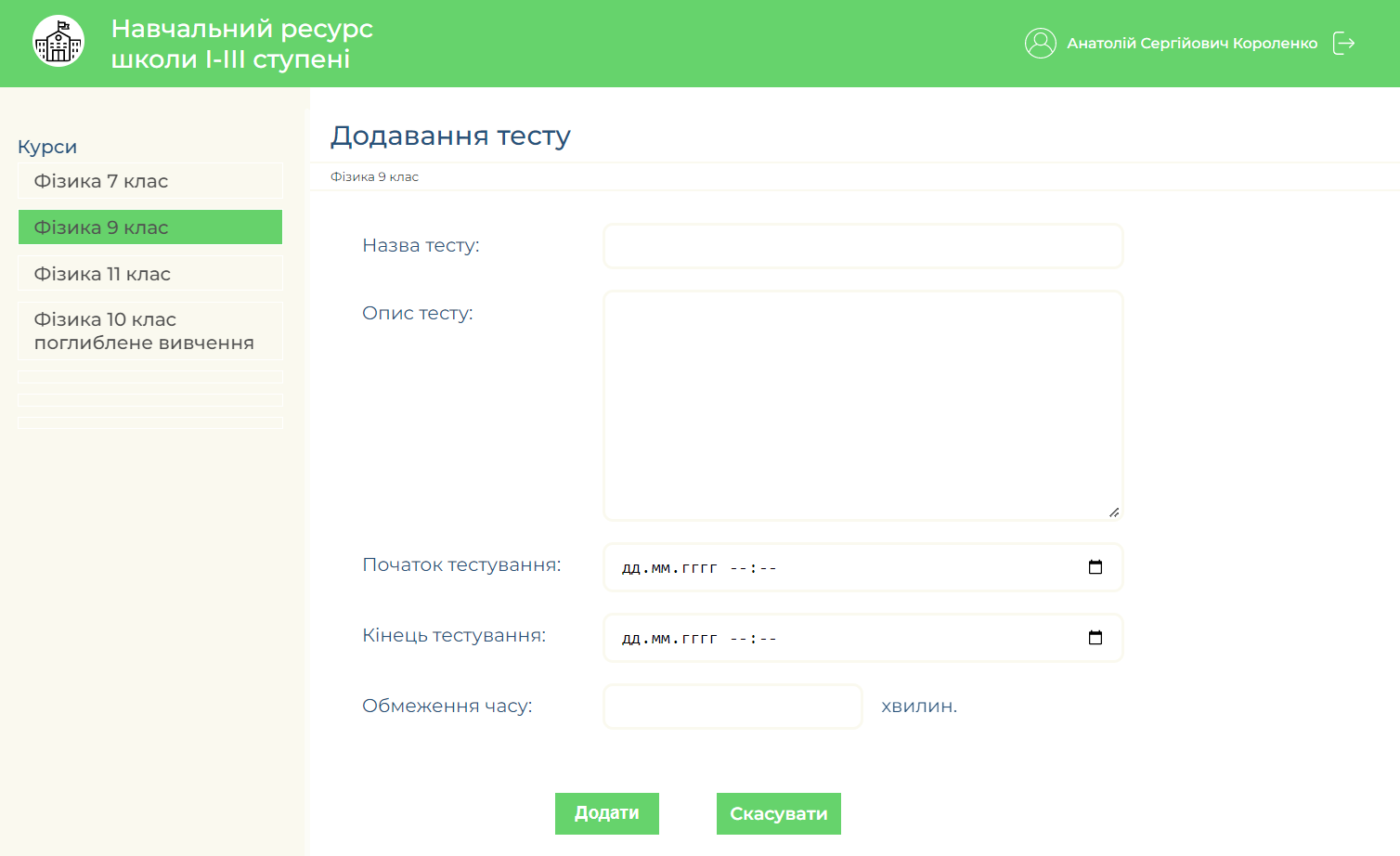


Рисунок 3.10 – Сторінка додавання тесту

Знизу сторінки присутня кнопка «Оцінки» натиснувши на яку вчитель перейде до журналу в якому може виставляти учням оцінки (рис 3.11). Тут відображається список учнів, які вивчають цей предмет, і їх оцінки. Вчитель може як виставити нові оцінки, так і змінити вже наявні. Після всіх потрібних корегувань вчитель натискає кнопку «Зберегти» та система відправляє запит до бази даних та зберігає зміни.



Рисунок 3.11 – Оцінки за предмет

Для перегляду результатів тестування вчитель повинен відкрити сторінку курсу та прогорнути її донизу. Там у вигляді таблиці відображено перелік учнів та їх результат. А також є можливість перегляду спроб та оцінки відкритих питань (рис. 3.12 – 3.13).

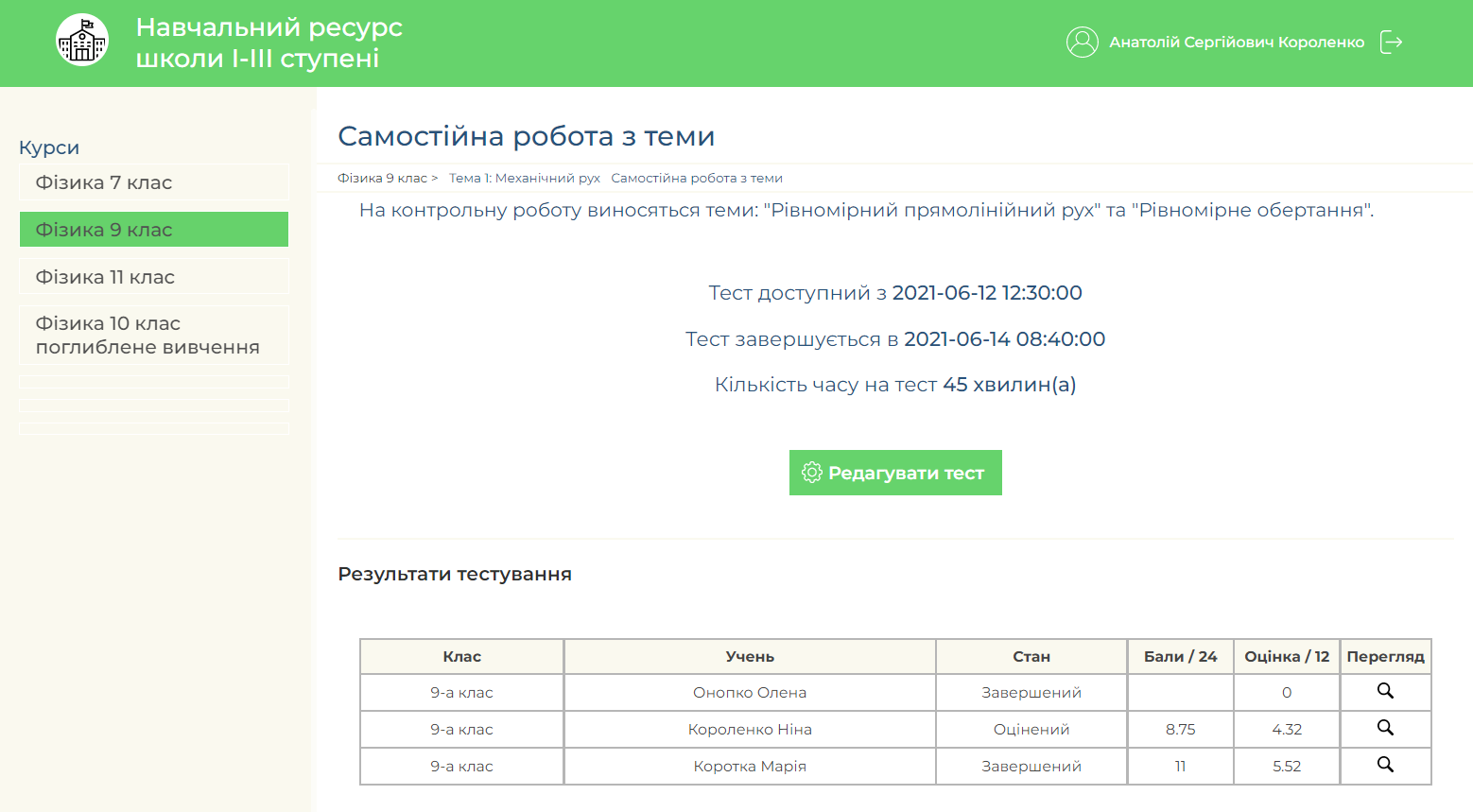


Рисунок 3.12 – Сторінка тесту

При перегляді спроб проходження тестів учнями зліва є навігаційна панель з номерами питань. Кожне питання позначено відповідним кольором: червоні – з невірною відповіддю, з частково правильною – жовтим, з правильною – зеленим.

У разі відсутності відповіді питання позначається білим кольором. Якщо питання є відкритого типу то вчитель повинен його оцінити ввівши оцінку у відповідне поле на натиснути кнопку «Оцінити».

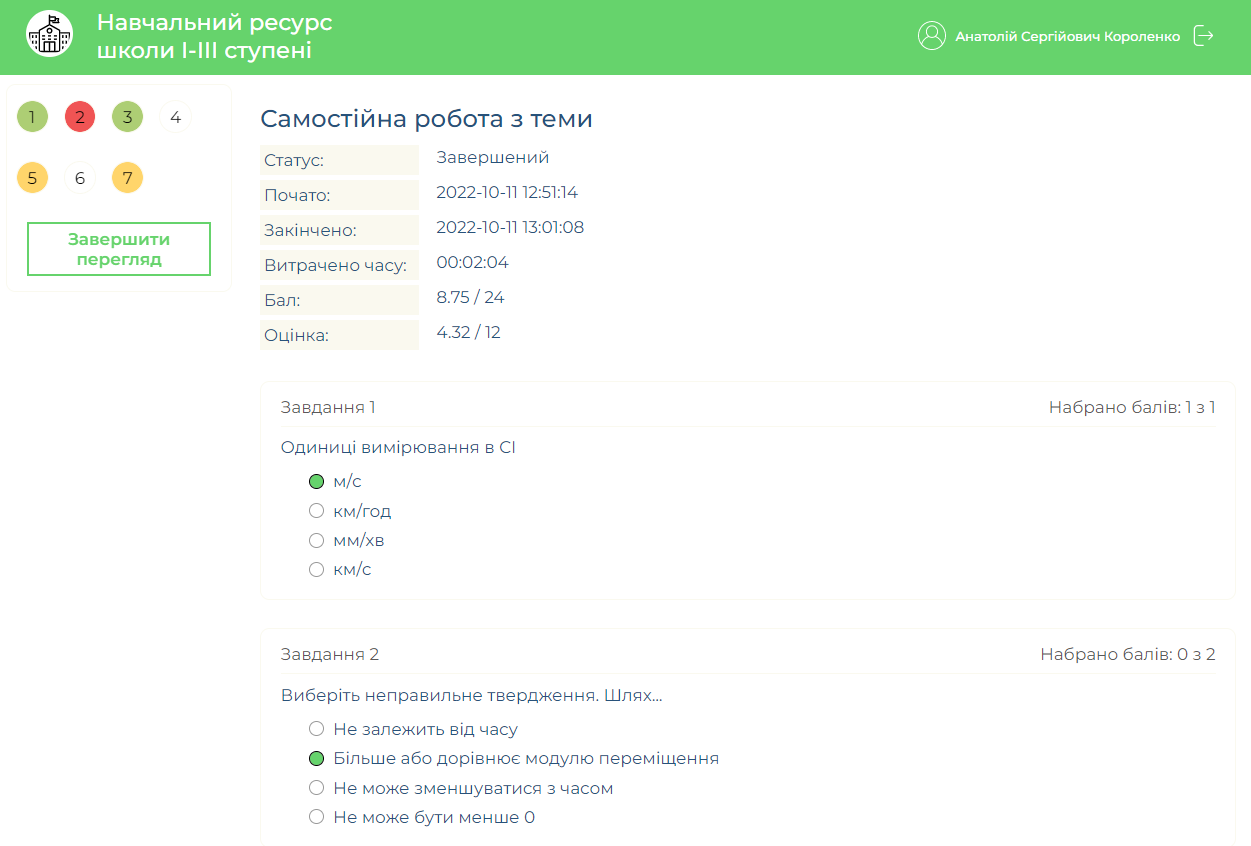


Рисунок 3.13 – Перевірка тесту

За потребою тест можна редагувати. Для цього потрібно натиснути відповідну кнопку та відкриється сторінка редагування тесту, де можна як додати нові питання, так і змінити чи видалити вже наявні (рис. 3.14).

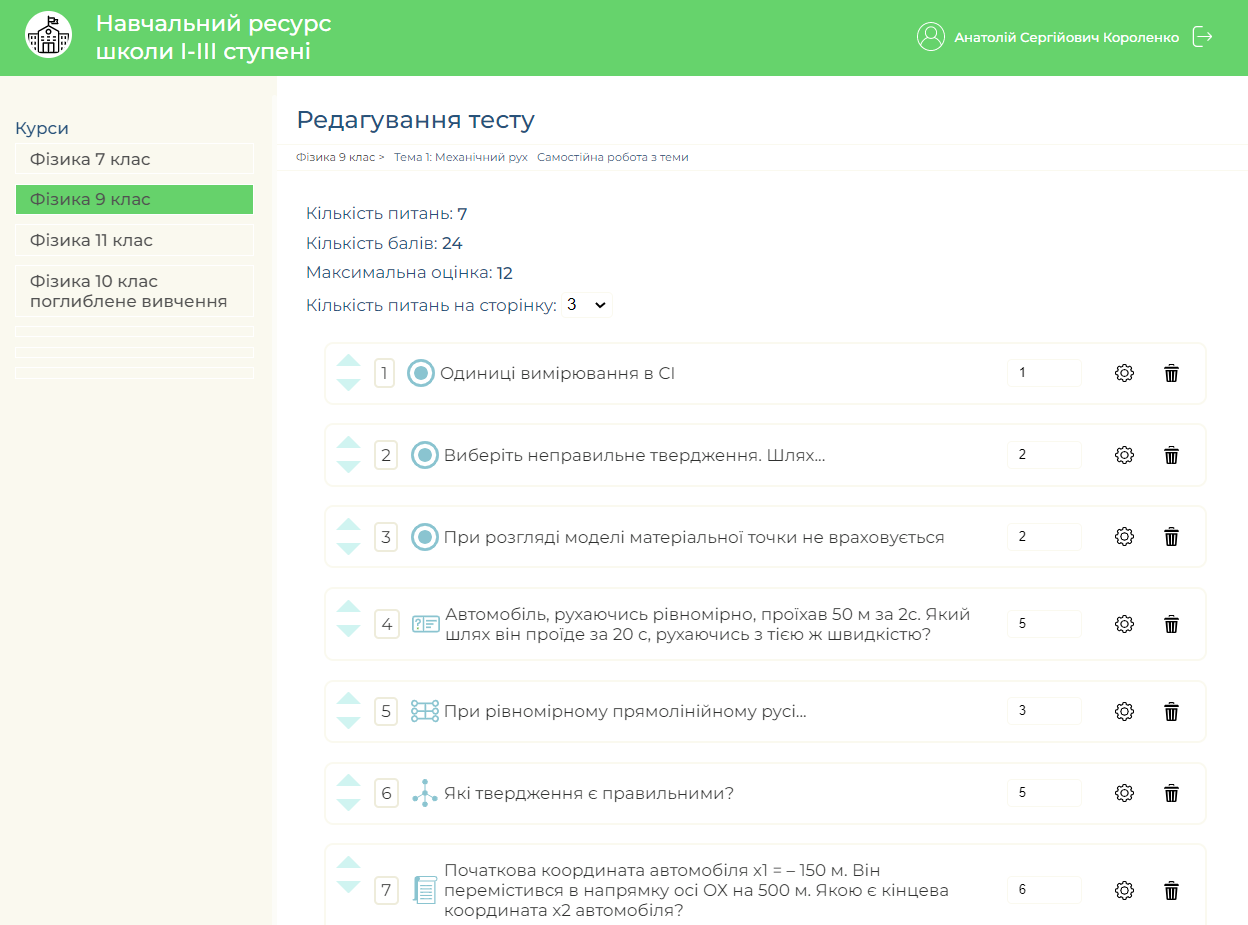


Рисунок 3.14 – Редагування тесту

Система може містити п’ять різних типів питань:

* одиничного вибору (рис. 3.15). При створенні цього типу питань вчитель повинен ввести текст запитання у поле «Занальне» та кількість балів за нього. В блоці «Відповіді» додаються варіанти відповідей та кожен з них помічається як правильний, чи неправильний, але лише один варіант відповіді може бути правильним;
* множинного вибору (рис. 3.16). При створенні цього питання вчитель повинен ввести текст запитання у поле «Занальне» та кількість балів за нього. Для цього типу кількість правильних відповідей не обмежена В блоці «Відповіді» додаються варіанти відповідей та кожен з них помічається як правильний, чи неправильний;
* з короткою відповіддю (рис. 3.17). При створеці цього типу питання в полі «Загальне» вчитель додає текст запитання та встановлює за нього потрібну кількість балів. В блоці «Відповіді» додаються можливі варіанти правильних відповідей, кожна з яких буде зарахована як вірна;
* на відповідність (рис. 3.18). При створенні цього типу питання в полі «Загальне» вчитель додає загальний текст запитання та встановлює за нього потрібну кількість балів. А нижче вводить правильні відповідності питання та відповіді, при цьому вчитель може створити зайві варіанти відповідей, які будуть неправильні;
* есе (рис. 3.19). При створенні цього типу питання вчитель має ввести лише сам текст запитання та кількість балів, а також має можливість дозволити учням прикріпляти файл чи заборонити.

За допомогою кнопки зі знаком «Плюс», у випадку недостатньої кількості полів для відпоідей, можна створити додаткові поля.

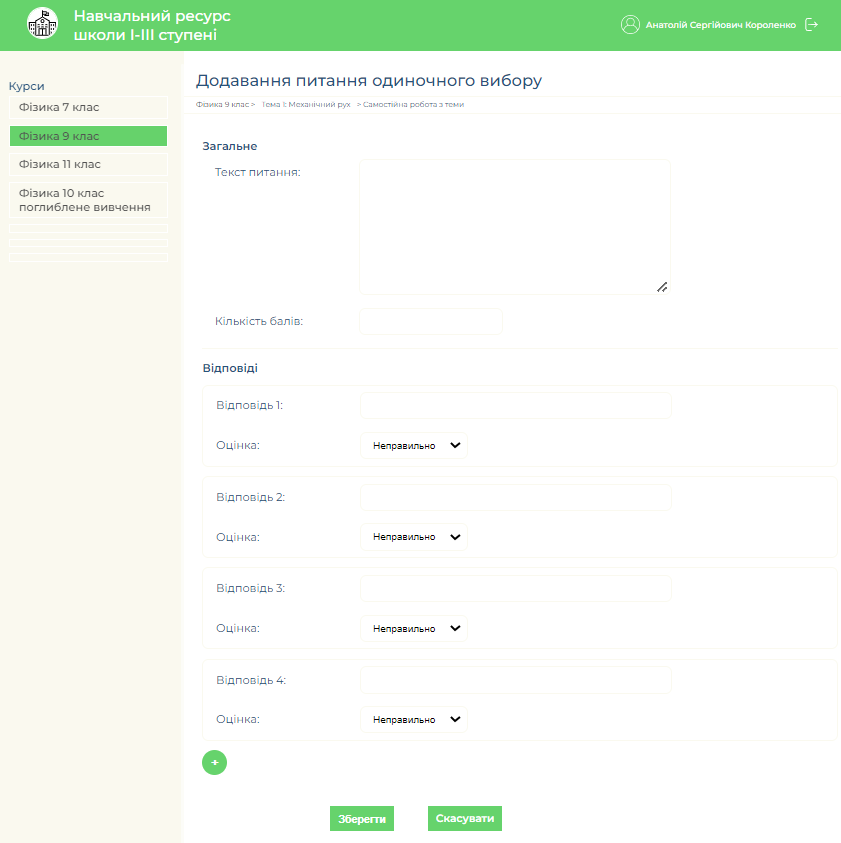


Рисунок 3.15 – Створення питання одиничного вибору

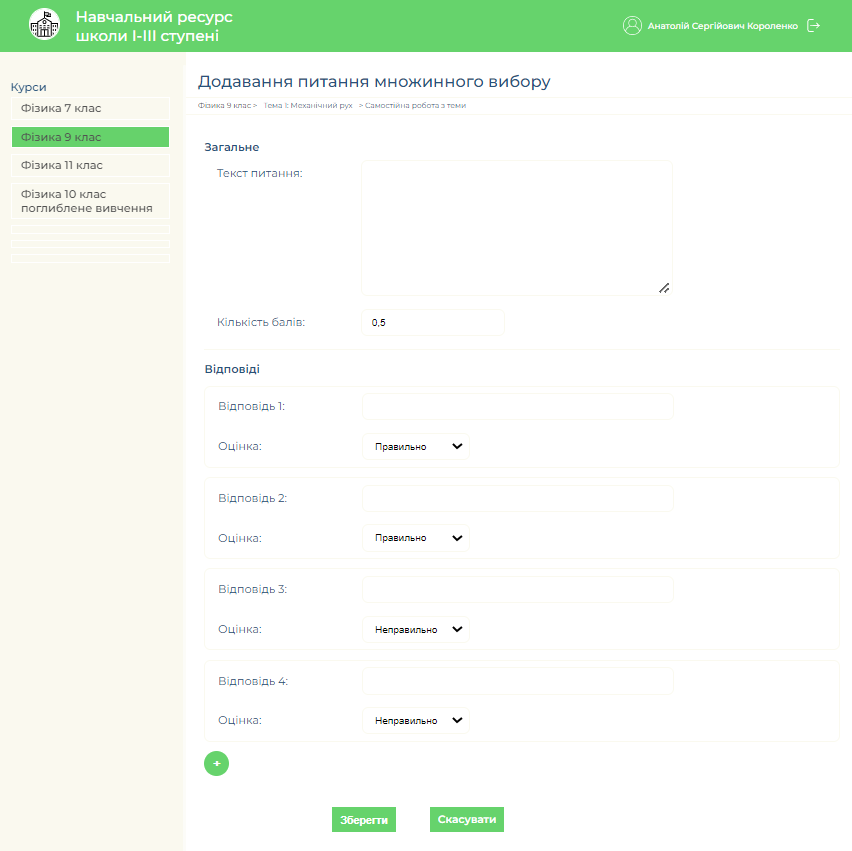


Рисунок 3.16 – Створення питання множинного вибору

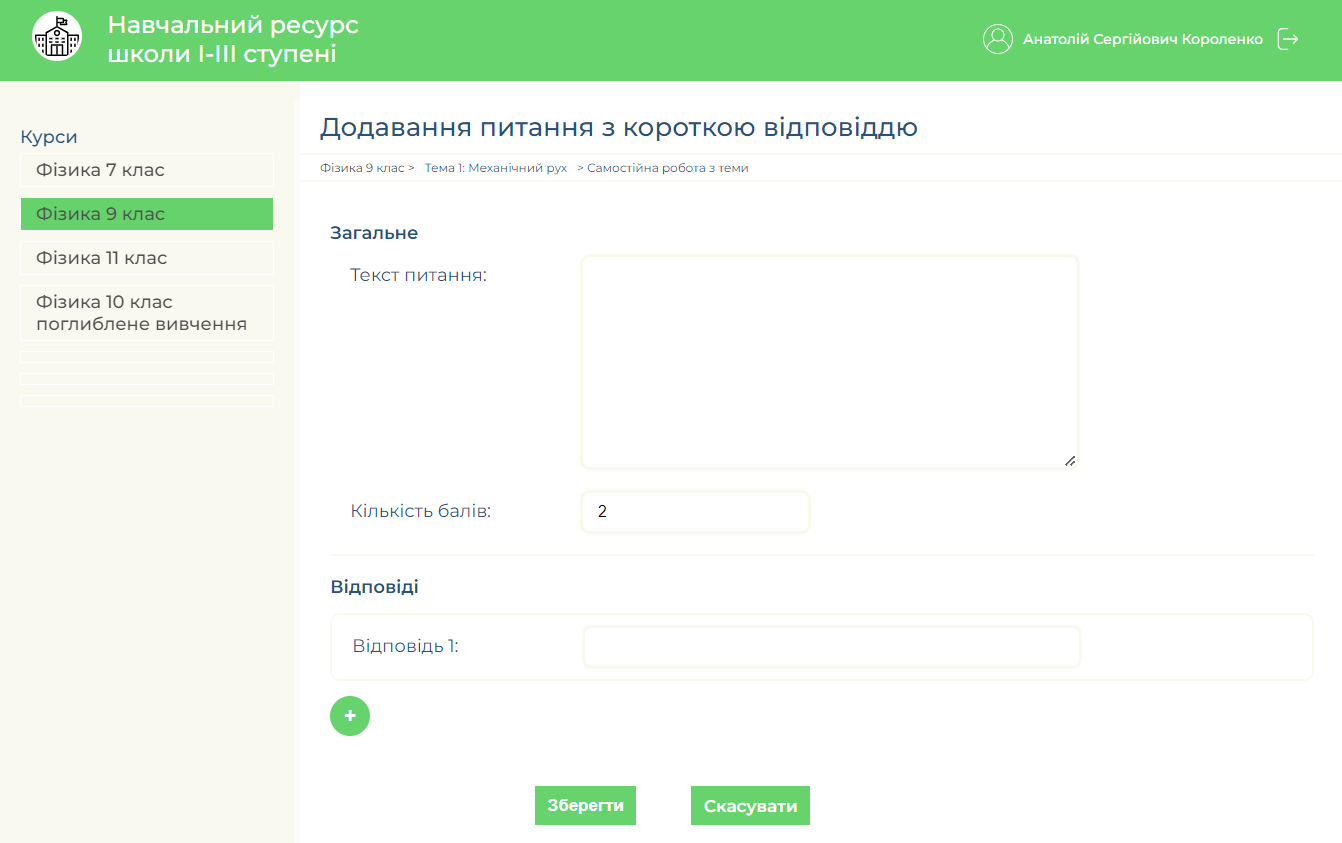


Рисунок 3.17 – Створення питання з короткою відповіддю

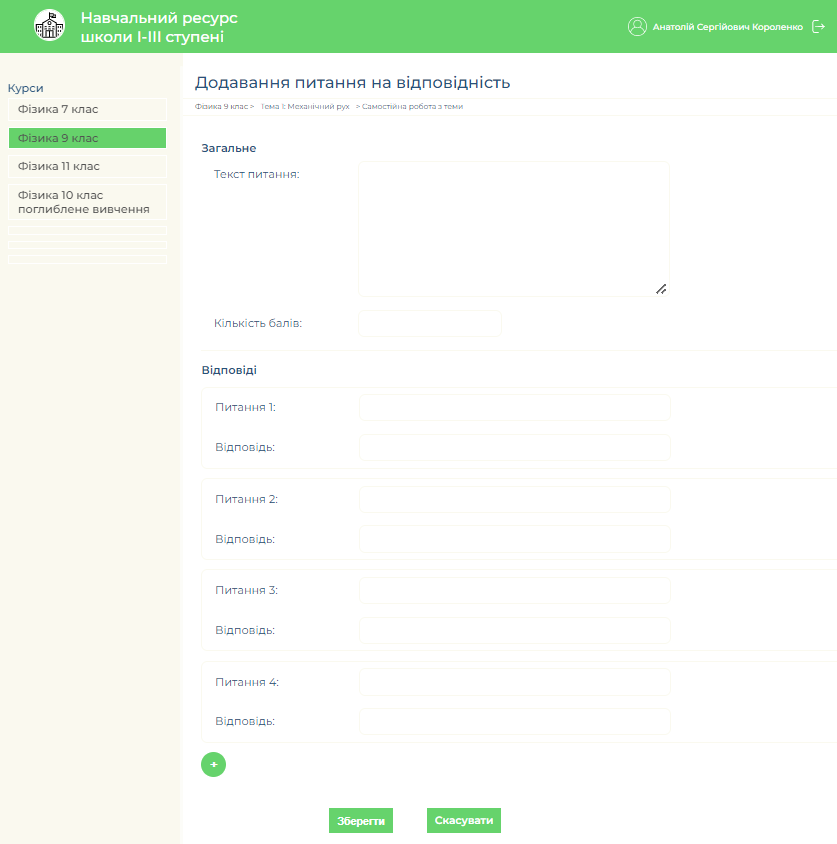


Рисунок 3.18 – Створення питання на відповідність

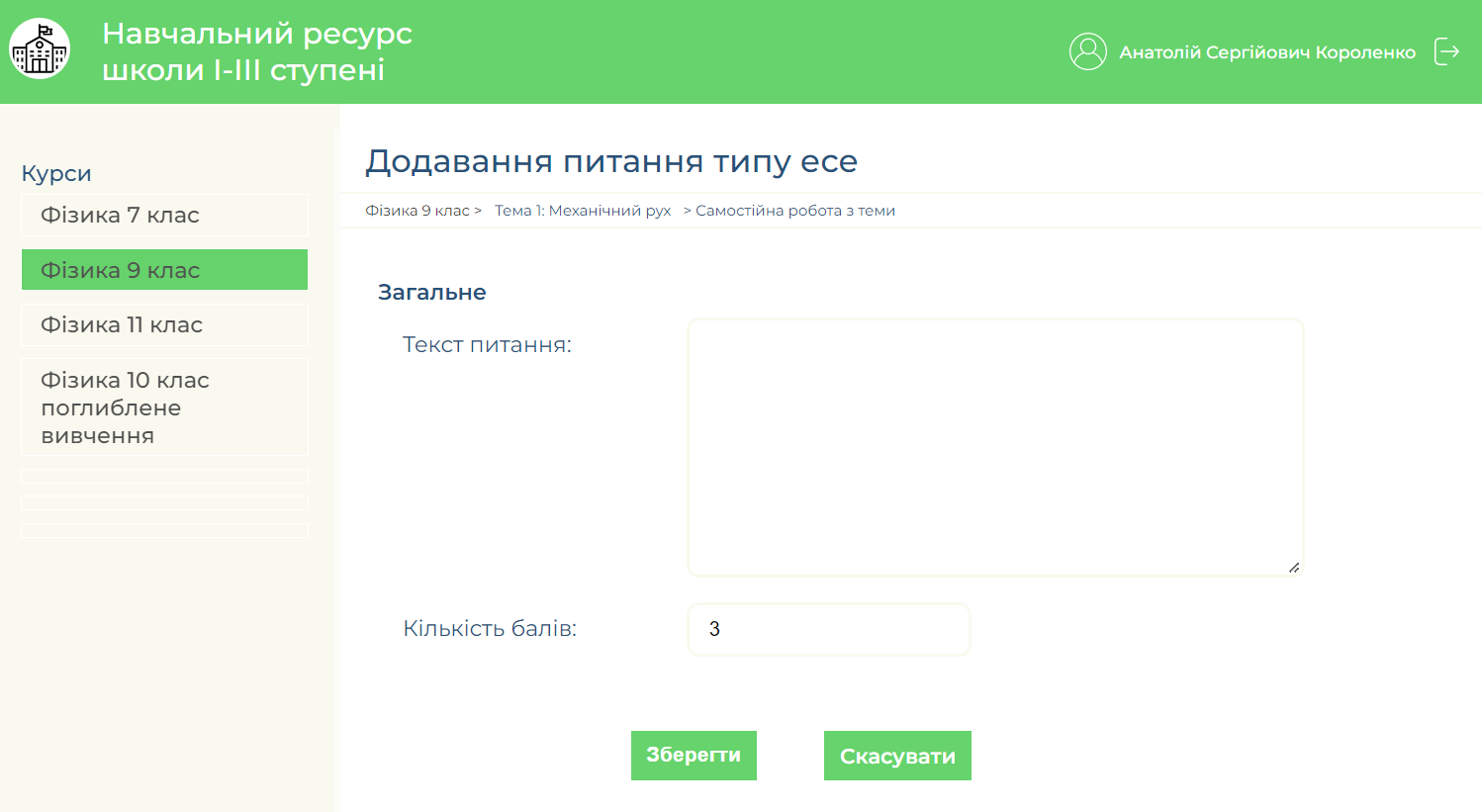


Рисунок 3.19 – Створення питання типу есе

При встановлені кількості балів за питання вчитель може встановити лише додатні числа. Для кожного курсу присутня сторінка зі статистикою (рис. 3.20). Для переходу на неї користувач повинен натиснути кнопку «Статистика».

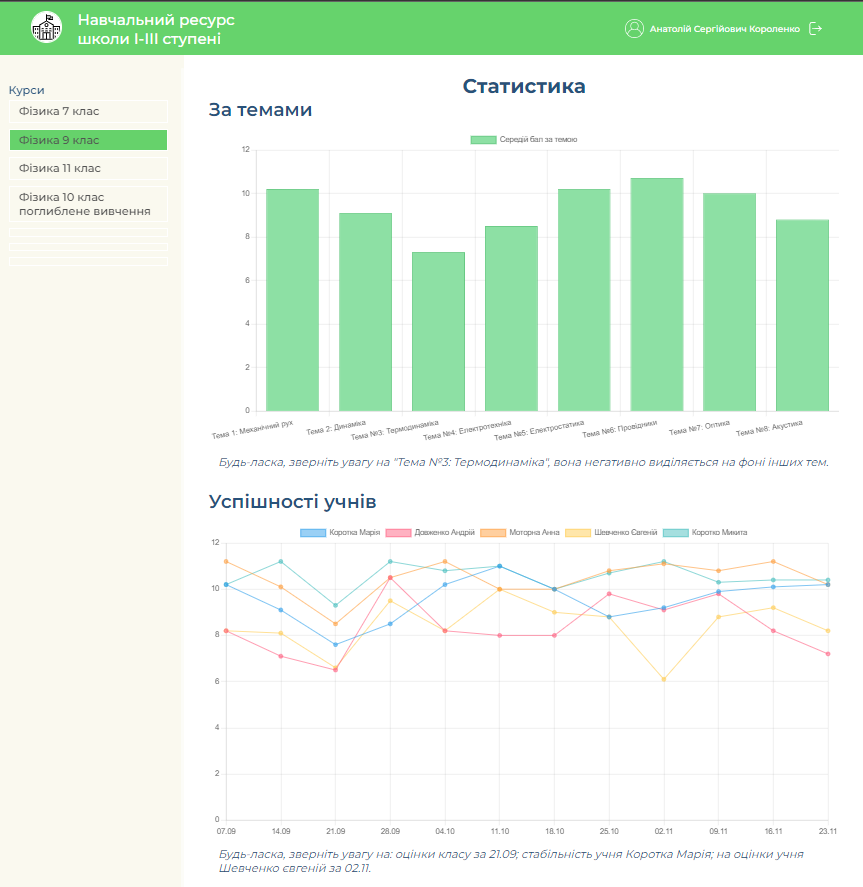


Рисунок 3.20 – Сторінка статистики по предмету

На сторінці відображаються два загальних графіки за результатами аналізу успішності учнів:

* перший графік по успішності учнів за темами. Гістограма на якій відображено загальна успішність всіх учнів классу за кожною темою;
* другий графік по успішності кожного учня за тиждень. Лінійіна діаграма на якій відображено успішність кожного учня окремо по тижням.

Також до кожного графіку знизу присутні висновки та рекомендації згідно аналізу на основі данних графіку. Рекомендації можуть стосуватись як загальної успішності всього класу, так і якогось окремого учня. А також може виділяти окремі теми на які вчителю потрібно звернути увагу.

# РОЗДІЛ 4 ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ

## **4.1. Вибір способу тестування**

Однією з основних частин проектування програмного забезпечення є його тестування. Якість виробленого програмного продукту напряму залежить від того наскільки якісно було проведено його тестування.

Головною метою тестування програмного продукту є довести, що всі заявлені розробником функції працюють правильно, а застосунок готовий до випуску. Якщо належне тестування програмного забезпечення буде відсутнім, це може спровокувати велику хвилю поганих відгуків від користувачів і згодом сформувати негативне ставлення до продукту і цілому.

Основною цілю тестування є забезпечення потрібної якості застосунків та зниження ризику критичних та помилок в цілому. При успішному проходженні тестування, програмне забезпечення починає впроваджуватися, при провалу проходження тестування – продукт відправляється на переробку.

При правильному тестуванні у кінці розробник програмного продукту може переконатися в тому, що всі вимогу до застосунку були дотримані, а система в цілому була реалізована правильно.

Тестування забезпечує зручність, безпеку та надійність створеної програми, а тому вважається важливим в процесі розробки програмного забезпечення.

При тестуванні розробленої інтелектуальної інформаційної системи електронного навчання було обрано функціональне ручне тестування методом «білого ящика», так як є доступ до програмного коду застосунку.

## 

## **4.2. Тест план**

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

* створення нового тесту;
* створення нового завдання;
* додавання нового запису з навчальним матеріалом до курсу;
* додавання нового файлу з навчальним матеріалом до курсу;
* додавання нового питання до тесту;
* перевірка виконаного завдання;
* перегляд та перевірка складених тестів;
* додавання нової теми на сторінку курсу;
* виставлення оцінок.

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

Таблиця 4.1 – Тест-кейси для модуля вчителя

|  |  |
| --- | --- |
| **Опис:** | Додавання нової теми на сторінку курсу |
| **Умова:** | Виконано вхід в обліковий запис викладача та відкрито сторінку предмету, що він викладає. |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою миші натиснути на кнопку з плюсом внизу сторінки курсу. | Відкривається випадаюче меню з вибором типу об’єкта для додавання. |
| 2. Натиснути на посилання «Тема» у випадаючому меню. | Відкривається сторінка додавання нової теми в систему. |
| 3. Ввести коректні дані | Введено коректні дані. |
| 4. Натиснути кнопку «Зберегти» | Відбувається переадресація на сторінку курсу з останнім записом нової теми. |
| **Опис:** | Додавання запису з навчальним матеріалом в систему |
| **Умова:** | Виконано вхід в обліковий запис вчителя та відкрито сторінку предмету, що він викладає. |

Продовження таблиці 4.1

|  |  |
| --- | --- |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою миші натиснути на кнопку з плюсом внизу сторінки курсу. | Відкривається випадаюче меню з вибором типу об’єкта для додавання. |
| 2. У випадаючому меню Натиснути на посилання «Запис». | Відкривається сторінка додавання нового запису в систему. |
| 3. Ввести коректні дані | Введено коректні дані. |
| 4. Натиснути кнопку «Зберегти» | Відбувається переадресація на сторінку курсу з відображеним новим записом внизу сторінки. |
| **Опис:** | Додавання файлу з навчальним матеріалом в систему |
| **Умова:** | Виконано вхід в обліковий запис вчителя та відкрито сторінку предмету, що він викладає. |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою миші натиснути на кнопку з плюсом внизу сторінки курсу. | Відкривається випадаюче меню з вибором типу об’єкта для додавання. |
| 2. У випадаючому меню натиснути на посилання «Файл». | Відкривається сторінка додавання нового файлу в систему. |
| 3. Ввести коректні дані | Введено коректні дані. |
| 4. Натиснути кнопку «Зберегти» | Відбувається переадресація на сторінку курсу з відображеним новим файлом внизу сторінки. |
| **Опис:** | Створення завдання в системі |
| **Умова:** | Виконано вхід в обліковий запис вчителя та відкрито сторінку предмету, що він викладає. |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою миші натиснути на кнопку з плюсом внизу сторінки курсу. | Відкривається випадаюче меню з вибором типу об’єкта для додавання. |
| 2. У випадаючому меню натиснути на посилання «Завдання». | Відкривається сторінка додавання нового завдання в систему. |

Продовження таблиці 4.1

|  |  |
| --- | --- |
| 3. Ввести коректні дані | Введено коректні дані. |
| 4. Натиснути на кнопку «Зберегти». | Відбувається переадресація на сторінку курсу з відображеним новим завданням внизу сторінки. |
| **Опис:** | Створення тесту в системі |
| **Умова:** | Виконано вхід в обліковий запис вчителя та відкрито сторінку предмету, що він викладає. |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою миші натиснути на кнопку з плюсом внизу сторінки курсу. | Відкривається випадаюче меню з вибором типу об’єкта для додавання. |
| 2. У випадаючому меню натиснути на посилання «Тест». | Відкривається сторінка додавання нового тесту в систему. |
| 3. Ввести коректні дані | Введено коректні дані. |
| 4. Натиснути кнопку «Зберегти» | Відбувається переадресація на сторінку курсу з відображеним новим тестом внизу сторінки. |
| **Опис:** | Додавання питання в тест |
| **Умова:** | Виконано вхід в обліковий запис вчителя та відкрито сторінку предмету, що він викладає. |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою миші натиснути на потрібний тест на сторінці курсу | Відкривається сторінка з інформацією про тест. |
| 2. Натиснути на посилання «Редагувати тест». | Відкривається сторінка редагування тесту. |
| 3. Лівою кнопкою миші натиснути на кнопку з плюсом внизу сторінки тесту. | Відкривається випадаюче меню з вибором типу питання для додавання. |
| 4. Натиснути на посилання з потрібним типом питання. | Відкривається сторінка додавання нового питання відповідно по обраного типу. |
| 5. Ввести коректні дані | Введено коректні дані. |
| 6. Натиснути кнопку «Зберегти» | Відбувається переадресація на сторінку редагування тесту з переліком питань, останнім з яких є створене. |

Продовження таблиці 4.1

|  |  |
| --- | --- |
| **Опис:** | Перегляд та перевірка складених тестів |
| **Умова:** | Виконано вхід в обліковий запис вчителя та відкрито сторінку предмету, що він викладає. |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою миші натиснути на потрібний тест на сторінці курсу | Відкривається сторінка з інформацією про тест. |
| 2. Перейти в нижню частину сторінки в розділ «Результати тестування» та, у рядку потрібного учня, натиснути на посилання з лупою у стовпці «Перегляд» | Відкривається сторінка перегляду відповідей учня на тест. |
| 3. Натиснути в навігаційному меню на неперевірене питання, що відмічено білим кольором. | Відбувається перехід на обране питання. |
| 4. В поле з підписом «Бали» вести коректну оцінку за питання. | Введено коректні дані. |
| 5. Натиснути на кнопку «Оцінити». | Оцінка питання збережена. |
| 6. Натиснути на посилання «Завершити перегляд». | Відбувається переадресація на сторінку тесту. |
| **Опис:** | Перевірка виконаного завдання |
| **Умова:** | Виконано вхід в обліковий запис вчителя та відкрито сторінку предмету, що він викладає. |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою миші натиснути на посилання з завданням. | Відкривається сторінка з інформацією про завдання та таблицею надісланих відповідей від учнів внизу сторінки. |
| 2. Ввести коректні дані в стовпці «Бал» та «Коментар» рядка з відповіддю потрібного учня. | Введено коректні дані. |

Продовження таблиці 4.1

|  |  |
| --- | --- |
| 3. Натиснути кнопку «Зберегти» | Відбувається збереження нової інформації, а статус завдання змінюється на «Оцінене». |
| **Опис:** | Виставлення оцінок |
| **Умова:** | Виконано вхід в обліковий запис вчителя та відкрито сторінку предмету, що він викладає. |
| Кроки: | Очікуваний результат: |
| 1. Лівою кнопкою миші натиснути на посилання з потрібним завданням. | Відкривається сторінка з інформацією про завдання та таблицею надісланих відповідей від учнів внизу сторінки. |
| 2. Ввести коректні дані в стовпці «Бал» та «Коментар» рядка з відповіддю потрібного учня. | Введено коректні дані. |
| 3. Натиснути кнопку «Зберегти» | Відбувається збереження нової інформації. |

# ВИСНОВОК

У результаті розробки дипломного проекту було створено функціонал вчителя інтелектуальної інформаційної системи електронного навчання використовуючи мову програмування РНР та систему управління базами даних MySQL.

Створений модуль вчителя дозволяє додавати новий навчальний матеріал до курсу в системі, створювати та перевіряти завдання, створювати та редагувати тести. Тести в розробленій системі можуть мати питання кожного з п’яти типів: питання типу есе, з короткою відповіддю, на відповідність, множинного вибору, одиничного вибору.

На першому етапі розробки проаналізовано обрану предметну область, а саме електронне навчання, та розглянуто основні наявні аналоги інформаційних систем для автоматизації освітньої діяльності, їх зовнішній вигляд, функціональні можливості, а також технології, що використовувались при створенні розглянутих застосунків.

На другому етапі розробки спроектовано інтелектуальну інформаційну систему, а саме створено концептуальну схему програмного продукту, діаграми варіантів використання, виконано логічне моделювання системи з використанням діаграми декомпозиції та контекстної діаграми.

Розроблено базу даних, визначено та описано головні її сутності, у результаті чого спроектовано фізичну модель БД, що складається з 19-ти таблиць.

На заключному етапі було розроблено адаптивний користувацький інтерфейс застосунку, який на протестовано використовуючи функціональне ручне тестування методом білого ящику, попередньо створивши тест план та тест-кейси.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 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.
2. Міністерство освіти України Організація дистанційного навчання в школі методичні рекомендації: методичні рекомендації – Київ, 2020. – 36 с.
3. Биков В. Ю. Моделі організаційних систем відкритої освіти: монографія. Київ: Атіка, 2008. 684 с.
4. Дистанційна освіта в Україні [Електронний ресурс] – Режим доступу: <http://mon.gov.ua/activity/education/distanczijna/>
5. Bissell C. C. The Open University of the United Kingdom / C. C. Bissell // Leadership in Science and Technology: A Reference Handbook. – 2011. – Vol. 2. – P. 24–32.
6. Заріцька С.І., Литвиненко Н.І., Савченко М.І., Сліпченко О.Ю. Методичні аспекти впровадження електронного навчання в закладах загальної середньої освіти: методичний посібник – Київ, 2019. – 64 с.
7. М. В. Мокрієв Структура електронного навчального курсу для заочної (дистанційної) форми навчання // Відкрите освітнє е-середовище сучасного університету, № 3 (2017) – с. 320 – 329.
8. Moodle [Електронний ресурс] – Режим доступу: <https://moodle.org>
9. Ilias [Електронний ресурс] – Режим доступу: <http://www.ilias.de>
10. Blackboard [Електронний ресурс] – Режим доступу: <http://www.blackboard.com>
11. Створення сучасного електронного курсу в системі MOODLE: навчальний посібник / Н.В. Морзе, О.П. Буйницька, Л.О. Варченко-Троценко. – Кам’янець-Подільський : ПП Буйницький О.А., 2016. – 232 с.
12. Система электронного обучения Moodle: обзор возможностей и функционала [Електронний ресурс] – Режим доступу: <https://lmslist.ru/free-sdo/obzor-moodle>
13. What Is Blackboard Learn? | Blackboard Help [Електронний ресурс] – Режим доступу: [https://help.blackboard.com/Learn/Instructor/Getting\_Started/  
    What\_Is\_Blackboard\_Learn](https://help.blackboard.com/Learn/Instructor/Getting_Started/What_Is_Blackboard_Learn)
14. Дистанційне навчання: моделі, технології, перспективи [Електронний ресурс] – Режим доступу: <http://confesp.fl.kpi.ua/node/%201123>
15. А. М. Стрюк // Теорія та методика електронного навчання. – Кривий Ріг: Видавничий відділ ДВНЗ «Криворізький національний університет», 2015. – Том VI. – Випуск 1 (6) : спецвипуск «Монографія в журналі». – 286 с.
16. Kaplana A. Haenleinb M. Siri, Siri, in my hand: Who’s the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. [Електронний ресурс] – Режим доступу: [https://www.sciencedirect.com/science/  
    article/pii/S0007681318301393](https://www.sciencedirect.com/science/article/pii/S0007681318301393)
17. Machine Learning – Машинне навчання [Електронний ресурс] – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning>
18. Штучний Інтелект в освітньому процесі [Електронний ресурс] – Режим доступу: <https://www.futureschool.online/post/artificial_intelligence/>
19. Як штучний інтелект може допомогти освіті [Електронний ресурс] – Режим доступу: <https://osvitoria.media/experience/yak-shtuchnyj-intelekt-mozhe-dopomogty-osviti>
20. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навчальний посібник – Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. – 434 с.
21. Зубкова Т.М. Технология разработки программного обеспечения: навч. посіб. – Оренбург: ОГУ, 2017. – 468 с.
22. Что Такое AJAX И Как Он Работает [Електронний ресурс] – Режим доступу: <https://www.hostinger.ru/rukovodstva/chto-takoje-ajax>
23. Система управления базами данных MySQL [Електронний ресурс] – Режим доступу: <https://depix.ru/articles/sistema_upravleniya_bazami_dannyh_mysql>
24. Гриценко В. Г. Створення та впровадження інформаційно-аналітичної системи управління університетом. Новітні комп’ютерні технології. Кривий Ріг: Видавничий центр ДВНЗ «Криворізький національний університет», 2016. Т. 14. С. 57–58.
25. PHP and MySQL Web Development / L.Welling, L. Thomson – 2016 –652c.

# ДОДАТОК А ЛІСТИНГ КОДУ ДЛЯ РЕАЛІЗАЦІЇ РЕДАГУВАННЯ ТЕСТІВ

<?php

session\_start();

if(!isset($\_SESSION["userType"]) || $\_SESSION["userType"] != 'teacher'){

header("Location: ../");

die;

}

if(($\_GET['test'] == 'one' || $\_GET['test'] == 'multi') && isset($\_GET['id'])){

echo '<section class="add-topic">

<div class="container">

<h3 class="homework-title">';

if ($\_GET['test'] == 'one') {echo 'Додавання питання одиничного вибору';}

else {echo 'Додавання питання множинного вибору';}

echo '</h3>

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

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

echo '</div>

<form action="../course/query.php" Method="POST" class="add-topic-form">

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

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

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

<p class="add-test-title">Загальне</p>

<div class="add-topic\_\_input">

<label>

Текст питання:

</label>

<textarea name="text" id="" cols="60" rows="10" class=""></textarea>

</div>

<div class="add-topic\_\_input">

<label>

Кількість балів:

</label>

<div class="add-test-num">

<input name="points" type="number" value="1.0" min="0" step="0.1" >

</div></div>

<p class="add-test-title line">Відповіді</p>

<div class="add-test\_\_block">

<div class="add-topic\_\_input">

<label>

Відповідь 1:

</label>

<input name="answer1" type="text">

</div>

<div class="add-topic\_\_input">

<label>

Оцінка:

</label>

<div class="add-test-num">

<select name="istrue1" id="">

<option value="0">Неправильно</option>

<option value="1">Правильно</option>

</select>

</div></div></div>

<div class="add-test\_\_block">

<div class="add-topic\_\_input">

<label>

Відповідь 2:

</label>

<input name="answer2" type="text">

</div>

<div class="add-topic\_\_input">

<label>

Оцінка:

</label>

<div class="add-test-num">

<select name="istrue2" id="">

<option value="0">Неправильно</option>

<option value="1">Правильно</option>

</select>

</div></div></div>

<div class="add-test\_\_block">

<div class="add-topic\_\_input">

<label>

Відповідь 3:

</label>

<input name="answer3" type="text">

</div>

<div class="add-topic\_\_input">

<label>

Оцінка:

</label>

<div class="add-test-num">

<select name="istrue3" id="">

<option value="0">Неправильно</option>

<option value="1">Правильно</option>

</select>

</div></div></div>

<div class="add-test\_\_block">

<div class="add-topic\_\_input">

<label>

Відповідь 4:

</label>

<input name="answer4" type="text">

</div>

<div class="add-topic\_\_input">

<label>

Оцінка:

</label>

<div class="add-test-num">

<select name="istrue4" id="">

<option value="0">Неправильно</option>

<option value="1">Правильно</option>

</select>

</div></div></div>

<button type="button" class="course-add-material-button">

+

</button>

<div class="add-topic\_\_input-btn">

<button class="button">

Зберегти

</button>

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

Скасувати

</a>

</div>

</form>

</div>

</section>

<script>

let number = 5

document.querySelector(".course-add-material-button").addEventListener("click", function(){

let text = document.createElement("div")

text.className = "add-test\_\_block"

text.innerHTML = \'<div class="add-topic\_\_input"><label>Відповідь \'+ number +\':</label><input name="answer\'+ number +\'" type="text"></div><div class="add-topic\_\_input"><label>Оцінка:</label><div class="add-test-num"><select name="istrue\'+ number +\'" id=""><option value="0">Неправильно</option><option value="1">Правильно</option></select></div></div>\'

document.querySelector(".course-add-material-button").before(text)

number++

})

</script>';

}

elseif($\_GET['test'] == 'short' && isset($\_GET['id'])){

echo '<section class="add-topic">

<div class="container">

<h3 class="homework-title">

Додавання питання з короткою відповіддю

</h3>

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

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

echo '</div>

<form action="../course/query.php" Method="POST" class="add-topic-form">

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

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

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

<p class="add-test-title">Загальне</p>

<div class="add-topic\_\_input">

<label>

Текст питання:

</label>

<textarea name="text" id="" cols="60" rows="10" class=""></textarea >

</div>

<div class="add-topic\_\_input">

<label>

Кількість балів:

</label>

<div class="add-test-num">

<input name="points" type="number" value="1.0" min="0" step="0.1" >

</div></div>

<p class="add-test-title line">Відповіді</p>

<div class="add-test\_\_block">

<div class="add-topic\_\_input">

<label>

Відповідь 1:

</label>

<input name="answer1" type="text">

</div></div>

<button type="button" class="course-add-material-button">

+

</button>

<div class="add-topic\_\_input-btn">

<button class="button">

Зберегти

</button>

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

Скасувати

</a>

</div>

</form>

</div>

</section>

<script>

let number = 2

document.querySelector(".course-add-material-button").addEventListener("click", function(){

let text = document.createElement("div")

text.className = "add-test\_\_block"

text.innerHTML = \'<div class="add-test\_\_block"><div class="add-topic\_\_input"><label>Відповідь \'+ number +\':</label><input name="answer\'+ number +\'" type="text"></div></div>\'

document.querySelector(".course-add-material-button").before(text)

number++

})

</script>';

}

elseif($\_GET['test'] == 'vid' && isset($\_GET['id'])){

echo '<section class="add-topic">

<div class="container">

<h3 class="homework-title">

Додавання питання на відповідність

</h3>

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

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

echo '</div>

<form action="../course/query.php" Method="POST" class="add-topic-form">

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

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

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

<p class="add-test-title">Загальне</p>

<div class="add-topic\_\_input">

<label>

Текст питання:

</label>

<textarea name="text" id="" cols="60" rows="10" class=""></textarea >

</div>

<div class="add-topic\_\_input">

<label>

Кількість балів:

</label>

<div class="add-test-num">

<input name="points" type="number" value="1.0" min="0" step="0.1" >

</div>

</div>

<p class="add-test-title line">Відповіді</p>

<div class="add-test\_\_block">

<div class="add-topic\_\_input">

<label>

Питання 1:

</label>

<input name="question1" type="text">

</div>

<div class="add-topic\_\_input">

<label>

Відповідь:

</label>

<input name="answer1" type="text">

</div></div>

<div class="add-test\_\_block">

<div class="add-topic\_\_input">

<label>

Питання 2:

</label>

<input name="question2" type="text">

</div>

<div class="add-topic\_\_input">

<label>

Відповідь:

</label>

<input name="answer2" type="text">

</div></div>

<div class="add-test\_\_block">

<div class="add-topic\_\_input">

<label>

Питання 3:

</label>

<input name="question3" type="text">

</div>

<div class="add-topic\_\_input">

<label>

Відповідь:

</label>

<input name="answer3" type="text">

</div></div>

<div class="add-test\_\_block">

<div class="add-topic\_\_input">

<label>

Питання 4:

</label>

<input name="question4" type="text">

</div>

<div class="add-topic\_\_input">

<label>

Відповідь:

</label>

<input name="answer4" type="text">

</div></div>

<button type="button" class="course-add-material-button">

+

</button>

<div class="add-topic\_\_input-btn">

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

Зберегти

</button>

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

Скасувати

</a>

</div>

</form>

</div>

</section>

<script>

let number = 5

document.querySelector(".course-add-material-button").addEventListener("click", function(){

let text = document.createElement("div")

text.className = "add-test\_\_block"

text.innerHTML = \'<div class="add-test\_\_block"><div class="add-topic\_\_input"><label>Питання \'+ number +\':</label><input name="question\'+ number +\'" type="text"></div><div class="add-topic\_\_input"><label>Відповідь:</label><input name="answer\'+ number +\'" type="text"></div></div>\'

document.querySelector(".course-add-material-button").before(text)

number++

})

</script>';

}

elseif($\_GET['test'] == 'essay' && isset($\_GET['id'])){

echo '<section class="add-topic">

<div class="container">

<h3 class="homework-title">

Додавання питання типу есе

</h3>

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

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

echo '</div>

<form action="../course/query.php" Method="POST" class="add-topic-form">

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

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

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

<p class="add-test-title">Загальне</p>

<div class="add-topic\_\_input">

<label>

Текст питання:

</label>

<textarea name="text" id="" cols="60" rows="10" class=""></textarea >

</div>

<div class="add-topic\_\_input">

<label>

Кількість балів:

</label>

<div class="add-test-num">

<input name="points" type="number" value="1.0" min="0" step="0.1" >

</div>

</div>

<div class="add-topic\_\_input">

<label>

Прикріпляти файл:

</label>

<div class="add-test-num">

<select name="" id="">

<option value="Заборонити">Заборонити</option>

<option value="Дозволити">Дозволити</option>

</select>

</div>

</div>

<div class="add-topic\_\_input-btn">

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

Зберегти

</button>

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

Скасувати

</a>

</div>

</form>

</div>

</section>';

}

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

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

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

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

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

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

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

echo '

<section class="test-wrap">

<div class="container">

<h3 class="homework-title">

Редагування тесту

</h3>

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

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

echo '</div>

<div class="test-edit">

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

Кількість питань: <span> '.count($question).' </span>

</p>

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

Кількість балів: <span> '.$points[0][0].' </span>

</p>

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

Максимальна оцінка: <span> 12 </span>

</p>

<div class="test-edit\_\_text">

<lable>

Кількість питань на сторінку:

</lable>

<select name="" id="questionnumber">

<option value="1" '; if($test[0][10] == '1') echo 'selected'; echo '>1</option>

<option value="2" '; if($test[0][10] == '2') echo 'selected'; echo '>2</option>

<option value="3" '; if($test[0][10] == '3') echo 'selected'; echo '>3</option>

<option value="4" '; if($test[0][10] == '4') echo 'selected'; echo '>4</option>

<option value="5" '; if($test[0][10] == '5') echo 'selected'; echo '>5</option>

<option value="6" '; if($test[0][10] == '6') echo 'selected'; echo '>6</option>

<option value="10" '; if($test[0][10] == '10') echo 'selected'; echo '>10</option>

<option value="Усі" '; if($test[0][10] == 'Усі') echo 'selected'; echo '>Усі</option>

</select>

</div>';

for($i = 0; $i < count($question); $i++){

echo '<div class="course-materials-download">

<div action="" class="course-materials\_\_move">

<div class="">

<form action="../course/query.php" method="POST">

<input name="test" value="test" class="d-none">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="top" class="d-none">

<button class="course-materials\_\_to-top">

</button>

</form>

</div>

<div class="">

<form action="../course/query.php" method="POST">

<input name="test" value="test" class="d-none">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="bottom" class="d-none">

<button class="course-materials\_\_to-bottom">

</button>

</form>

</div>

</div>

<p class="test-edit\_\_num">

'.($i+1).'

</p>

<a href="add-test-mult.html" class="">

<img src="../img/';

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

echo 'one.svg" alt="" title="Одиничний вибір"';

}

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

echo 'multi.svg" alt="" title="Множинний вибір"';

}

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

echo 'vid.svg" alt="" title="Відповідність"';

}

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

echo 'short-answer.svg" alt="" title="Коротка відповідь"';

}

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

echo 'essay.svg" alt="" title="Есе"';

}

echo ' class="test-edit\_\_img" >

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

</a>

<div action="" class="course-materials-buttons">

<input type="number" class="test-edit\_\_input-num" id="'.$question[$i][0].'" value="'.$question[$i][3].'" step="0.1" min="0">

<a href="#" title="Редагувати">

<svg enable-background="new 0 0 512 512" viewBox="0 0 512 512" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</a>

<form action="../course/query.php" method="POST">

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

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

<button type="Sybmit" title="Видалити">

<svg height="20px" viewBox="-57 0 512 512" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</button>

</form>

</div>

</div>';

}

echo '<div class="course\_\_buttons">

<div class="course-add-material">

<button class="course-add-material-button">

+

</button>

<div class="course-add-material-menu d-none">

<a href="?c='.$\_GET['c'].'&test=one&id='.$\_GET['id'].'">Одиничний вибір</a>

<a href="?c='.$\_GET['c'].'&test=multi&id='.$\_GET['id'].'">Множинний вибір</a>

<a href="?c='.$\_GET['c'].'&test=short&id='.$\_GET['id'].'">Коротка відповідь</a>

<a href="?c='.$\_GET['c'].'&test=vid&id='.$\_GET['id'].'">Відповідність</a>

<a href="?c='.$\_GET['c'].'&test=essay&id='.$\_GET['id'].'">Есе</a>

</div></div></div></div></div>

</section>

';

}

else{

echo 'Error 404';

}

?>

<script>

document.querySelector('.course-add-material-button').addEventListener('click', function(event){

document.querySelector('.course-add-material-menu').classList.toggle('d-none')

})

</script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

document.querySelector('#questionnumber').addEventListener('change', function(){

$.ajax({

method: "POST",

url: "/course/query.php",

data: { pagecount: document.querySelector('#questionnumber').value,

test: <?php echo $\_GET['id'] ?>,

}

})

.done(function(result){

})

})

document.querySelectorAll('.test-edit\_\_input-num').forEach(num =>

num.addEventListener('change', function(){

$.ajax({

method: "POST",

url: "/course/query.php",

data: { points: num.value,

question: num.id,

}

})

.done(function(result){

})

})

)})

</script>

# ДОДАТОК Б ЛІСТИНГ КОДУ ДЛЯ РЕАЛІЗАЦІЇ СТОРІНКИ РЕДАГУВАННЯ КУРСУ

<?php

session\_start();

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

header("Location: ../");

die;

}

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

?>

<?php if ($\_SESSION["userType"] == 'teacher'): ?>

<section class="course course-edition">

<div class="container">

<h3 class="course-title">

<?php echo $subject[0][1] ?>

</h3>

<div class="course-author-info">

<div class="course-author-photo">

<img src="../img/teacher.png" alt="teacher">

</div>

<div class="course-description">

<p class="course-author">

<span>Вчитель:</span><?php echo "{$teacher[0][1]} {$teacher[0][2]} {$teacher[0][3]}"; ?>

</p>

<div class="course-subtitle\_\_wrap">

<p class="course-subtitle">

<?php echo $subject[0][2] ?>

</p>

<form action="" class="course-subtitle\_\_edit">

<button>

<svg height="20px" viewBox="-15 -15 484.00019 484" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</button>

</form>

</div>

</div>

</div>

<div class="course-materials">

<div class="course-materials-block">

<?php

for($i = 0; true; $i++){

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

if($i == $topic[$j][4]){

echo '<div class="course-materials-title">

<div class="course-materials-title\_\_wrap">

<div action="" class="course-materials\_\_move">

<div class="">

<form action="../course/query.php" method="POST">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="top" class="d-none">

<button class="course-materials\_\_to-top">

</button>

</form>

</div>

<div class="">

<form action="../course/query.php" method="POST">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="bottom" class="d-none">

<button class="course-materials\_\_to-bottom">

</button>

</form>

</div>

</div>

<p class="course-materials-title-name">'.$topic[$j][1].'</p>

</div>

<div action="" class="course-materials-title\_\_buttons">

<form action="../course/query.php" method="POST">

<input class="d-none" name="table" value="topic"><input class="d-none" name="id" value="'.$topic[$j][0].'">

<button title="Приховати">

'; if($topic[$j][5] == 1){

echo ' <input class="d-none" name="visibility" value="0">';

echo '<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 511.992 511.992" width="20px" xml:space="preserve">…</svg>';

}

echo '</button></form>

<a href="?c='.$\_GET['c'].'&t=topic&a=change&id='.$topic[$j][0].'" title="Редагувати">

<svg height="20px" viewBox="-15 -15 484.00019 484" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</a>

<form action="../course/query.php" method="POST">

<input class="d-none" name="action" value="delete"><input class="d-none" name="table" value="topic"><input class="d-none" name="id" value="'.$topic[$j][0].'">

<button title="Видалити">

<svg height="20px" viewBox="-57 0 512 512" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</button></form>

</div>

</div>';

goto cont2;

}

}

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

if($i == $file[$j][5]){

echo '<div class="course-materials-download">

<div action="" class="course-materials\_\_move">

<div class="">

<form action="../course/query.php" method="POST">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="top" class="d-none">

<button class="course-materials\_\_to-top">

</button>

</form>

</div>

<div class="">

<form action="../course/query.php" method="POST">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="bottom" class="d-none">

<button class="course-materials\_\_to-bottom">

</button>

</form>

</div>

</div>';

echo "<a href='../course/files/".$file[$j][2]."' class='' download='".$file[$j][1]."'>";

if(substr(strrchr($file[$j][2], "."), 1) == 'docx' || substr(strrchr($file[$j][2], "."), 1) == 'doc'){

echo '<svg version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 512 512" width="30px" xml:space="preserve">…</svg>';

}

elseif(substr(strrchr($file[$j][2], "."), 1) == 'pdf'){

echo '<svg version="1.1" id="Layer\_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"

viewBox="0 0 309.267 309.267" width="30px" style="enable-background:new 0 0 309.267 309.267;" xml:space="preserve">…</svg>';

}

elseif(substr(strrchr($file[$j][2], "."), 1) == 'ppt'){

echo '<svg version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"

viewBox="0 0 56 56" width="30px" style="enable-background:new 0 0 56 56;" xml:space="preserve">…</svg>';

}

else{

echo '<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 58 58" width="30px" xml:space="preserve">…</svg>';

}

echo $file[$j][1];

echo '</a>

<div action="" class="course-materials-buttons">

<form action="../course/query.php" method="POST">

<input class="d-none" name="table" value="file"><input class="d-none" name="id" value="'.$file[$j][0].'">

<button title="Приховати">

'; if($file[$j][6] == 1){

echo ' <input class="d-none" name="visibility" value="0">';

echo '<svg version="1.1" id="Capa\_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 511.992 511.992" width="20px" style="enable-background:new 0 0 511.992 511.992;" xml:space="preserve">…</svg>';

}

echo '</button></form>

<a href="#" title="Редагувати">

<svg height="20px" viewBox="-15 -15 484.00019 484" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</a>

<form action="../course/query.php" method="POST">

<input class="d-none" name="action" value="delete"><input class="d-none" name="table" value="file"><input class="d-none" name="id" value="'.$file[$j][0].'">

<button title="Видалити">

<svg height="20px" viewBox="-57 0 512 512" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</button></form>

</div>

</div>';

goto cont2;

}

}

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

if($i == $note[$j][5]){

echo '<div class="course-materials-download">

<div action="" class="course-materials\_\_move">

<div class="">

<form action="../course/query.php" method="POST">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="top" class="d-none">

<button class="course-materials\_\_to-top">

</button>

</form>

</div>

<div class="">

<form action="../course/query.php" method="POST">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="bottom" class="d-none">

<button class="course-materials\_\_to-bottom">

</button>

</form>

</div>

</div>';

echo "<a href='#' class=''>";

echo '<svg enable-background="new 0 0 497 497" viewBox="0 0 497 497" width="30px" xmlns="http://www.w3.org/2000/svg">…</svg>';

echo $note[$j][1];

echo '</a>

<div action="" class="course-materials-buttons">

<form action="../course/query.php" method="POST">

<input class="d-none" name="table" value="note"><input class="d-none" name="id" value="'.$note[$j][0].'">

<button title="Приховати">

'; if($note[$j][6] == 1){

echo ' <input class="d-none" name="visibility" value="0">';

echo '<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 511.992 511.992" width="20px" xml:space="preserve">…</svg>';

}

echo '</button></form>

<a href="?c='.$\_GET['c'].'&t=note&a=change&id='.$note[$j][0].'" title="Редагувати">

<svg height="20px" viewBox="-15 -15 484.00019 484" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</a>

<form action="../course/query.php" method="POST">

<input class="d-none" name="action" value="delete"><input class="d-none" name="table" value="note"><input class="d-none" name="id" value="'.$note[$j][0].'">

<button title="Видалити">

<svg height="20px" viewBox="-57 0 512 512" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</button></form>

</div>

</div>';

goto cont2;

}

}

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

if($i == $hw[$j][7]){

echo '<div class="course-materials-download">

<div action="" class="course-materials\_\_move">

<div class="">

<form action="../course/query.php" method="POST">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="top" class="d-none">

<button class="course-materials\_\_to-top">

</button>

</form>

</div>

<div class="">

<form action="../course/query.php" method="POST">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="bottom" class="d-none">

<button class="course-materials\_\_to-bottom">

</button>

</form>

</div>

</div>';

echo "<a href='?c=".$\_GET['c']."&view=task&id=".$hw[$j][0]."' class=''>";

echo '<svg xmlns="http://www.w3.org/2000/svg" height="30px" version="1.1" viewBox="0 -3 511.99922 511" width="30px">

<g id="surface1">…</svg>';

echo $hw[$j][1];

echo '</a>

<div action="" class="course-materials-buttons">

<form action="../course/query.php" method="POST">

<input class="d-none" name="table" value="homework"><input class="d-none" name="id" value="'.$hw[$j][0].'">

<button title="Приховати">

'; if($hw[$j][8] == 1){

echo ' <input class="d-none" name="visibility" value="0">';

echo '<svg version="1.1" id="Capa\_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 511.992 511.992" width="20px" style="enable-background:new 0 0 511.992 511.992;" xml:space="preserve">…</svg>';

}

echo '</button></form>

<a href="?c='.$\_GET['c'].'&t=task&a=change&id='.$hw[$j][0].'" title="Редагувати">

<svg height="20px" viewBox="-15 -15 484.00019 484" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</a>

<form action="../course/query.php" method="POST">

<input class="d-none" name="action" value="delete"><input class="d-none" name="table" value="homework"><input class="d-none" name="id" value="'.$hw[$j][0].'">

<button title="Видалити">

<svg height="20px" viewBox="-57 0 512 512" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</button></form>

</div>

</div>';

goto cont2;

}

}

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

if($i == $test[$j][5]){

echo '<div class="course-materials-download">

<div action="" class="course-materials\_\_move">

<div class="">

<form action="../course/query.php" method="POST">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="top" class="d-none">

<button class="course-materials\_\_to-top">

</button>

</form>

</div>

<div class="">

<form action="../course/query.php" method="POST">

<input name="position" value="'; echo $i.'" class="d-none"><input name="action" value="bottom" class="d-none">

<button class="course-materials\_\_to-bottom">

</button>

</form>

</div>

</div>';

echo "<a href='../course/?c=".$\_GET['c']."&view=test&id=".$test[$j][0]."' class=''>";

echo '<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:svgjs="http://svgjs.com/svgjs" version="1.1" width="30px" height="30px" x="0" y="0" viewBox="0 0 511 511.99998">…</svg>';

echo $test[$j][1];

echo '</a>

<div action="" class="course-materials-buttons">

<form action="../course/query.php" method="POST">

<input class="d-none" name="table" value="test"><input class="d-none" name="id" value="'.$test[$j][0].'">

<button title="Приховати">

'; if($test[$j][6] == 1){

echo ' <input class="d-none" name="visibility" value="0">';

echo '<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 511.992 511.992" width="20px" xml:space="preserve">…</svg>';

}

else{

echo ' <input class="d-none" name="visibility" value="1">';

echo '<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 512.001 512.001" width="20px" xml:space="preserve">…</svg>';

}

echo '</button></form>

<a href="?c='.$\_GET['c'].'&t=test&a=change&id='.$test[$j][0].'" title="Редагувати">

<svg height="20px" viewBox="-15 -15 484.00019 484" width="20px" xmlns="http://www.w3.org/2000/svg">… </svg>

</a>

<form action="../course/query.php" method="POST">

<input class="d-none" name="action" value="delete"><input class="d-none" name="table" value="test"><input class="d-none" name="id" value="'.$test[$j][0].'">

<button title="Видалити">

<svg height="20px" viewBox="-57 0 512 512" width="20px" xmlns="http://www.w3.org/2000/svg">…</svg>

</button></form>

</div>

</div>';

goto cont2;

}

}

break;

cont2:

}

?>

<div class="course\_\_buttons">

<div class="course-add-material">

<button class="course-add-material-button"> + </button>

<div class="course-add-material-menu d-none">

<a href="?c=<?php echo $\_GET['c']; ?>&t=topic&a=add">Тема</a>

<a href="?c=<?php echo $\_GET['c']; ?>&t=note&a=add">Запис</a>

<a href="?c=<?php echo $\_GET['c']; ?>&t=file&a=add">Файл</a>

<a href="?c=<?php echo $\_GET['c']; ?>&t=task&a=add">Завдання</a>

<a href="?c=<?php echo $\_GET['c']; ?>&t=test&a=add">Тест</a>

</div>

</div>

<div class="course-grades">

<a href="?c=<?php echo $\_GET['c']; ?>&view=grades" class="course-grades\_\_button">

Оцінки

</a>

</div>

</div>

</div>

</section>

<script>

document.querySelector('.course-add-material-button').addEventListener('click', function(event){

document.querySelector('.course-add-material-menu').classList.toggle('d-none')

})

</script>

<?php endif ?>