

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**Кафедра комп'ютеризованих систем управління**

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач кафедри

\_\_\_\_\_ Литвиненко О. Є.  
«\_\_\_\_\_» \_\_\_\_\_ 2021 р.

**ДИПЛОМНИЙ ПРОЄКТ  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ  
“БАКАЛАВР”**

**Тема:** «Інтерактивний додаток для автоматизованого ведення електронного журналу. Компонента роботи з серверною частиною»

**Виконавець:** \_\_\_\_\_ Сефер О. В.

**Керівник:** \_\_\_\_\_ Вавіленкова А.І.

**Нормоконтролер:** \_\_\_\_\_ Тупота Є. В.

**Київ 2021**

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет Кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютеризованих систем управління

Спеціальність 123 «Комп'ютерна інженерія»

(шифр, найменування)

Освітньо-професійна програма «Системне програмування»

Форма навчання денна

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Литвиненко О.Є.

«\_\_\_\_\_» \_\_\_\_\_ 2021 р.

## ЗАВДАННЯ

**на виконання дипломної роботи (проєкту)**

**Сефера Олександра Володимировича**

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема дипломної роботи (проєкту): «Інтерактивний додаток для автоматизованого ведення електронного журналу. Компонента роботи з серверною частиною»

затверджена наказом ректора від «4» лютого 2021 р. №135/ст

2. Термін виконання роботи (проєкту): з 17.05.2021 по 20.06.2021

3. Вихідні дані до роботи (проєкту): системи для автоматизованого ведення електронного журналу, технології проєктування, структура додатків для ведення електронного журналу

4. Зміст пояснювальної записки: \_\_\_\_\_

1) електронне оцінювання у закладах освіти;

2) технології та алгоритми обробки даних у інтерактивному додатку ведення електронного журналу;

3) результати роботи інтерактивного додатку для автоматизованого ведення електронного журналу.

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

1) схема алгоритму «Авторизація користувачів у електронному журналі»;

2) схема алгоритму «Взаємодії користувачів із базою даних»;

3) плакат «Інтерфейс додатку «Сторінка класу»»;

4) плакат «Інтерфейс користувача «Сторінка учня»».

6. Календарний план-графік:

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Аналіз літератури і технічної документації	17.05.2021-19.05.2021	
2	Затвердження плану дипломного проекту та проведення консультацій з науковим керівником, щодо наповнення пояснювальної записки	20.05.2021-21.05.2021	
3	Дослідження систем для інтерактивного додатку для автоматизованого ведення електронного журналу	21.05.2021-23.05.2021	
4	Розробка архітектури додатку для інтерактивного додатку для автоматизованого ведення електронного журналу	24.05.2021-28.05.2021	
5	Створення веб-додатку автоматизованого відслідковування задач у проектах	29.05.2021-05.06.2021	
6	Написання пояснювальної записки та проходження нормоконтролю	06.06.2021-09.06.2021	
7	Підготовка демонстраційного матеріал	10.06.2021-13.06.2021	

7. Дата видачі завдання: «17» травня 2021 р.

Керівник дипломної роботи (проекту) \_\_\_\_\_ Вавіленкова А.І.  
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання \_\_\_\_\_ Сефер О.В.  
(підпис випускника) (П.І.Б.)

## РЕФЕРАТ

Пояснювальна записка до дипломного проєкту «Інтерактивний додаток для автоматизованого ведення електронного журналу. Компонента роботи з серверною частиною» 54 с., 27 рис., 2 табл., 16 літературних джерел, 3 додатки.

**ІНТЕРАКТИВНИЙ ЖУРНАЛ, СИСТЕМИ ОБЛІКУ УСПІШНОСТІ ШКОЛЯРІВ, ДВОРІВНЕВА АРХІТЕКТУРА, МОНІТОРИНГ УСПІШНОСТІ, ВЕБ-ДОДАТОК.**

Об'єкт дослідження – процес автоматизованого оцінювання та ведення електронного щоденника

Предметом дослідження є способи організації автоматизованого ведення електронного журналу.

Метою даного дипломного проєкту є проєктування та розробка серверної частини інтерактивного додатку для моніторингу та аналізу успішності школярів.

Методи дослідження – технології обліку та моніторингу успішності, порівняльний аналіз, методи об'єктно-орієнтованого програмування, методи проєктування за дворівневою архітектурою.

Здійснено огляд існуючих систем для моніторингу та аналізу успішності учнів (студентів, школярів); здійснено порівняльний аналіз їх функціональних можливостей; проаналізовано структуру систем для моніторингу та аналізу успішності учнів; здійснено огляд принципів дворівневої архітектури; реалізовано веб-додаток для моніторингу та аналізу успішності учнів у проєктах за дворівневою архітектурою.

Матеріали дипломного проєкту рекомендується використовувати для спрощення документообігу у середніх навчальних закладах, а також у системах для автоматизації обліку успішності учнів.

Прогнозні припущення про розвиток об'єкту та предмету дослідження – застосування в якості системи для моніторингу та аналізу успішності школярів у середніх навчальних закладах та закладах вищої освіти.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ.....	7
ВСТУП .....	8
РОЗДІЛ 1 ЕЛЕКТРОННЕ ОЦІНЮВАННЯ У ЗАКЛАДАХ ОСВІТИ .....	10
1.1. Використання електронних журналів в освіті .....	10
1.2. Аналіз існуючих електронних журналів для оцінювання у закладах освіти .....	15
1.3. Висновки до розділу .....	20
РОЗДІЛ 2 ТЕХНОЛОГІЇ ТА АЛГОРИТМИ ОБРОБКИ ДАНИХ У ІНТЕРАКТИВНОМУ ДОДАТКУ .....	22
2.1. Клієнт-серверна архітектура інтерактивного додатку ведення електронного журналу .....	22
2.2. Структура бази даних інтерактивного додатку .....	25
2.3. Алгоритми авторизації користувачів та взаємодії з базою даних в інтерактивного додатку .....	27
2.4. Висновки до розділу .....	35
РОЗДІЛ 3 РЕЗУЛЬТАТИ РОБОТИ ІНТЕРАКТИВНОГО ДОДАТКУ ДЛЯ АВТОМАТИЗОВАНОГО ВЕДЕННЯ ЕЛЕКТРОННОГО ЖУРНАЛУ ..	37
3.1. Структура інтерактивного додатку для автоматизованого ведення електронного журналу .....	37
3.2. Апробація інтерактивного додатку для автоматизованого ведення електронного журналу .....	42
3.3. Кількісна оцінка параметрів роботи інтерактивного додатку для автоматизованого ведення електронного журналу .....	47
3.4. Висновки до розділу .....	49
ВИСНОВКИ.....	50
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51 <a href="#">3</a>

ДОДАТОК А..... **Ошибка! Закладка не определена.**

ДОДАТОК Б..... **Ошибка! Закладка не определена.**

ДОДАТОК В..... **Ошибка! Закладка не определена.**

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

<i>API</i>	–	<i>Application Programming Interface</i>
<i>IDE</i>	–	<i>Integrated Development Environment</i>
<i>SQL</i>	–	<i>Structured Query Language</i>
<i>PHP</i>	–	Мова програмування для веб-розробки
<i>JavaScript</i>	–	Мова програмування, яка працює у браузерях
<i>Framework</i>	–	Програмна платформа для побудови додатків
<i>Laravel</i>	–	<i>Framework</i> для мови програмування <i>PHP</i>
<i>HTTP</i>	–	<i>Hyper Text Transfer Protocol</i>
МОН	–	Міністерство Освіти та Науки

## ВСТУП

На поточний момент велику роль грає розповсюдження Інтернету. Завдяки цьому будь-яка людина має змогу знаходячись вдома, працювати із великими масивами інформації, а також взаємодіяти із іншими людьми.

Одночасно із розвитком Інтернету у нашому розпорядженні з'являються більш сучасні багатофункціональні та високопродуктивні технічні пристрої, починаючи із звичайних настільних комп'ютерів та закінчуючи смарт-годинниками. Також, особливе місце у нашому житті займають смартфони, планшетні комп'ютери та ноутбуки. Ця група пристроїв виділяється із загальної маси тим, що вони надають можливість скористатися пристроєм буквально «на ходу».

Попри всі досягнення у цій сфері, до пандемії Коронавірусу більшість процесів все ж не були оцифровані, а бо були не в повній мірі. Одним із таких процесів є процес оцінювання учнів. Більшість шкіл працювало із паперовими носіями інформації: щоденники, журнали успішності, протоколи, прикази та інші документи. Ці носії точно не забезпечують людину необхідною мобільністю, про яку йшла мова раніше. До цього ж, у паперових носіїв є ряд недоліків:

- при великій кількості документів, накоплених за певний час, мають велику вагу;
- займають багато місця;
- потребують певних умов зберігання.

Для спрощення роботи та забезпечення мобільності рекомендується переходити на електронні носії інформації. Використовуючи їх, вирішується багато проблем, адже вони (електронні носії):

- здатні зберігати великі об'єми інформації;
- не займають багато місця;
- дозволяють легко копіювати дані, вирішуючи проблему втрачання даних;
- надають можливість дистанційного доступу до них.

Актуальність даної роботи обумовлена необхідністю переходу шкіл з паперових носіїв на цифрові, використовуючи систему, яка буде спрощувати операції по контролю успішності, документообігу та іншими функціями.



Об'єкт дослідження – процес автоматизованого оцінювання та ведення електронного щоденника.

Предметом дослідження є способи організації автоматизованого ведення електронного журналу.

Метою даного дипломного проекту є проектування та розробка простого у використанні інтерактивного веб-додатку для автоматизованого оцінювання, документообігу та аналізу успішності учнів.

Методи дослідження дипломного проекту Новизна дипломного проекту полягає у використанні дворівневої архітектури, дало змогу реалізувати додаток з гнучкою структурою, даючи можливість подальшої модифікації та вдосконалення функціоналу без значних зусиль. Таким чином, отриманий додаток є менш ресурсозатратним та «легким», а отже – його обслуговування буде коштувати менше, ніж відповідний додаток створений, наприклад, за трьохрівневою архітектурою.

Практичне значення отриманих результатів. У дипломному проекті створено веб-додаток для автоматизованого оцінювання, документообігу та аналізу успішності учнів. Він надає базовий функціонал системи для автоматизованого оцінювання, документообігу та аналізу успішності учнів. За його допомогою можна відслідковувати зміни успішності учня, фіксувати кількість робочих годин вчителя, отримати аналіз успішності та динаміку зміни оцінок кожного учня або класу.

Особистий внесок випускника полягає у створенні серверної та клієнтської частини інтерактивного додатку.

Апробація отриманих результатів. Теоретичні аспекти отриманих у дипломному проекті результатів проходили апробацію на міжнародній науково-технічній конференції “Інтелектуальні технології лінгвістичного аналізу”.

Публікації. Сефер О.В. Проблема пошуку інформації, яка відрізняється від текстової: міжнар. науково-техн. конф. «Інтелектуальні технології лінгвістичного аналізу», 20 – 21 жовтня 2020 р.: тези доп. – К., 2020. – С. 13.

Прогнозні припущення про розвиток об'єкту та предмету дослідження – застосування в якості системи для спрощення операцій по контролю успішності, документообігу та інформування учнів.

## РОЗДІЛ 1

### ЕЛЕКТРОННЕ ОЦІНЮВАННЯ У ЗАКЛАДАХ ОСВІТИ

#### 1.1. Використання електронних журналів в освіті

Процес навчання у середньому навчальному закладі спрямований на вирішення низки освітніх завдань, кожне з яких характеризується дидактичною завершеністю. Обов'язковими компонентами в цьому процесі є контроль теоретичних знань, умінь та практичних навичок, тобто перевірка його результативності.

Головною метою контролю як дидактичного засобу управління навчальним процесом є забезпечення його ефективності, що в результаті приводить до системи теоретичних знань, умінь, практичних навичок, самостійного застосування здобутих знань при вирішенні тої чи іншої проблеми, стимулювання навчальної діяльності школярів, формування у них прагнення до самоосвіти, підготовка їх до вступу у вищі та середні спеціальні навчальні заклади.

Процес контролю знань школярі складається з трьох основних компонентів:

1. Перевірка – виявлення певного рівня теоретичних знань, умінь та навичок на практиці для подальшого їх оцінювання;
2. Оцінка – визначення рівня теоретичних знань, умінь і навичок школяра на практиці;
3. Облік – фіксування результатів роботи школяра під час навчання у вигляді оцінок у журналі академічної групи, відомостях, залікових книжках.

Під час контролю процесу навчання потрібно розв'язувати низку завдань: визначити наскільки школярі готові до сприйняття, усвідомлення та засвоєння

Кафедра КСУ				НАУ 21 18 12 000 ПЗ			
Виконав	Сефер О.В.			Електронне оцінювання у навчальних закладах освіти	Літера	Аркуш	Аркушів
Керівник	Васіленко А.М.				Методів і засобів	54	
Консульт.					СП-435 123		
Норм. контр.	Тупота Є.В.						
Зав. Каф.	Литвиненко О.Є.						

навчання; оцінити обсяг і глибини набутих школярами теоретичних знань, умінь та практичних навичок. Перелічені вище та багато інших завдань визначають зміст контролю процесу навчання, який змінюється із зміною освітніх завдань.

Процес контролю виконує такі функції:

– навчальна – суть даної функції полягає в тому, що вона являється корисною для кожного із школярів. Слухаючи змістовну відповідь однокласника, школярі звіряють з нею свої знання, ставлять запитання, доповнюють її, що сприяє повторенню та систематизації знань усіх однокласників. Навчальне значення для школярів має й те, що вони слухають доповнення неповних відповідей школярів самим викладачем. Освітня функція чітко виявляється також у тому, що школяр, який відповідає перед класом, повторює вивчений матеріал, виконує практичні завдання, закріплюючи знання, вміння і навички, краще усвідомлює навчальний матеріал;

– виховна – дана функція виходить з того, що очікування перевірки та отримання якомога вищого балу оцінювання спонукає школяра вчасно готуватися до здачі домашніх завдань лабораторних та курсових робіт. Крім того, перевірка та оцінювання теоретичних та практичних знань допомагають школяру самому оцінити рівень своїх знань та здібностей, що спонукає їх до поглибленого вивчення матеріалу. Все це сприяє формуванню позитивного ставлення школярів до навчання;

– діагностична – забезпечує виявлення успіхів та недоліків у теоретичних знаннях та практичних навичках школярів під час навчального процесу. Тоді ж встановлюються причини недоліків та шляхи їх усунення, визначаються заходи, що спрямовані на поліпшення успішності школярів;

– стимулююча – під час старших класів (10-й та 11-й) за відмінне навчання (усі відмітки не нижче 10-ти балів) учень має змогу отримати Золоту Медаль;

– оцінювальна – дана функція забезпечує об'єктивне оцінювання теоретичних та практичних знань, умінь та навичок школярів, а також сприяє кращому навчанню;

– управлінську – на основі контролю визначається стан успішності школярів, що дає змогу запобігти нерезультативного та неефективного процесу навчання або подолати його.

В такому випадку викладач може коригувати і свою власну діяльність: змінювати методику викладання, вдосконалювати навчальну діяльність школярів, застосовувати стимулюючі методи.

Під час навчального процесу проводиться контроль отриманих школярами знань. Розрізняють поточний, тематичний та підсумковий види контролю.

Поточний контроль проводиться у повсякденній навчальній роботі школяра. Він полягає в систематичному спостереженні викладача за навчальною діяльністю школярів під час занять та за навчальною роботою дома. Метою такого контролю є отримання оперативних даних про рівень знань та рівень підготовки школярів. Також даний вид контролю демонструє якість навчальної роботи під час занять та вирішення завдань управління навчальним процесом.

Періодичний (тематичний) контроль проводиться для оцінки рівня теоретичних знань та практичних навиків школярів, які були отримані на попередніх заняттях. Метою даного виду контролю є визначення, наскільки успішно школярі володіють системою знань, чи відповідають ці знання програмі. Даний вид контролю спрямований на перевірку знань школярів певної теми. Така система дає змогу усунути елементи випадковості при виведенні підсумкових оцінок, що трапляється, коли викладач орієнтується лише на поточний контроль. Якщо навчальним планом передбачено виконання школярами практичних, лабораторних робіт та інших обов'язкових практичних завдань, то їхнє виконання є обов'язковою умовою допуску школярами до тематичного контролю. Найчастіше, школярі пишуть так звану «тематичну» контрольну роботу та отримують оцінку за тему за результатом цієї контрольної роботи та набраних балів протягом теми

Підсумковий вид контролю проводиться в кінці навчального семестру у вигляді семестрової контрольної роботи. Семестрова контрольна робота, зазвичай, прирівнюється за вагомістю до «тематичної», але різниця полягає у тому, що вона містить у собі весь, пройдений за семестр, матеріал, що дозволяє повністю оцінити успішність школяра, а також, зробити певні висновки щодо розуміння школярами в цілому певної теми.

Ефективність контролю залежить від якості його організації: часу проведення контрольних занять, їх частоти й послідовності; характеру й форм самостійної роботи школярів (індивідуальна, групова); використання дидактичних і технічних

засобів навчання; поєднання методів контролю і самоконтролю (усна, письмова, графічна, практична, тестова, програмована перевірка); фіксування й оформлення даних контролю процесу навчання.

З огляду на виклики та нагальні потреби, які постали перед системою освіти України в умовах протидії пандемії *COVID-19*, та з метою дерегуляції та дебіюрократизації управління системою освіти, спрощення ведення документообігу закладів загальної середньої освіти (далі – ЗЗСО) Міністерством освіти і науки України було поставлено завдання ДНУ «Інститут освітньої аналітики» (наказ МОН № 781 від 09.06.2020) забезпечити розробку державних безкоштовних електронних журналів та щоденників в рамках розвитку Програмно-апаратного комплексу «Автоматизований комплекс освітнього менеджменту» (далі – ПАК «АІКОМ»), який є модернізованою системою ІТС «ДІСО». Впровадження даного програмного продукту є інновацією, що надасть для педагогічних працівників, батьків та учнів ЗЗСО зручний онлайн-інструмент, який цілодобово забезпечить доступ до оцінок, відвідуваності та інших ресурсів журналу. Впровадження такого інформаційно-технологічного рішення є особливо актуальним для тих ЗЗСО та громад, які не можуть собі дозволити закупівлю альтернативних комерційних програмних продуктів.

Електронний журнал успішно вирішує ряд завдань, що стоять перед викладачами, та нічим не поступається своєму попереднику – традиційному паперовому журналу, а навіть має свої переваги. В наш час все частіше і частіше спостерігається тенденція зростання реалізації інформаційних технологій в питаннях автоматизації проблем управління освітою. Тому створення електронного журналу успішності є актуальним завданням.

Електронний журнал надає велику кількість переваг над традиційним паперовим, але він вимагає певний мінімум технічних засобів для роботи з ним: вільний комп'ютер та працююча мережа. Також потрібен адміністратор, який буде заповнювати актуальні дані списку класів та предметів.

Перевагами електронного журналу для викладача є те, що журнали всіх класів, в яких він викладає, знаходяться на одному сервері, і для доступу до них достатньо комп'ютера, ноутбука, планшета чи телефона, які мають доступ до мережі. В електронному журналі дуже зручно знайти журнал з певного предмету

та певного класу на відміну від паперового. Достатньо великою перевагою є те, що з даними в електронному журналі нічого не трапиться: електронний журнал не можливо загубити, випадково порвати чи намочити. Також можливість зміни оцінок школярів з певного предмету має тільки викладач, який веде цей предмет, що надає захист від випадкової або зловмисної зміни оцінок сторонніми особами.

Перевагами електронного журналу для школярів (та їх батьків) є те, що кожен з них може зайти в систему і подивитися всі свої (або своєї дитини) актуальні оцінки з різних предметів. Школяр може бачити тільки свої оцінки, що забезпечує конфіденційність інформації яка стосується тільки його та викладача.

Створення альтернативи паперовому журналу та зберігання даних успішності школярів у комп'ютері – ідея не нова. Застосування електронного журналу значно полегшило б доступ до інформації про поточну успішність школярів, дозволило виконувати різні статистичні запити для наступного їх аналізу. Розробки електронних журналів проводяться як професійними компаніями, так і викладачами та студентами-ентузіастами.

Першими прототипами електронного журналу, були електронні таблиці *MS Excel*. Створення такого журналу не потребує спеціальних знань, його дуже просто заповнювати, а його структура й зовнішній вигляд нагадують свого паперового попередника. Але найбільшим недоліком такого журналу є орієнтованість електронних таблиць на одного користувача. Журнал, створений за допомогою таблиць *MS Excel* не можуть одночасно заповнювати кілька викладачів. Дані з нього важко викласти у відкритий доступ, тому що інформація в електронних таблицях погано структурована, що ускладнює одержання різної статистичної інформації.

В наш час створено велику кількість професійних продуктів, що виконують роль електронного журналу успішності, як українських так і закордонних. Деякі працюють як веб-додатки для роботи з якими окрім комп'ютера потрібний веб-браузер та доступ до мережі інтернет. Інші – як звичайні комп'ютерні програми, потребують встановлення на комп'ютер для роботи з ними. Веб-додатки отримали більшу популярність, так як вони можуть працювати на різних операційних системах, де є веб-браузер та доступ до мережі інтернет. Головною проблемою електронних журналів, які мають вигляд комп'ютерних програм, що потрібно

встановлювати є те, що їх потрібно розробляти окремо під кожну операційну систему, якщо це потребується.

## 1.2. Аналіз існуючих електронних журналів для оцінювання у закладах освіти

Представниками електронних журналів успішності у вигляді веб-додатків є: «МКР» – (Методичний комплекс розкладу) це програмний комплекс який представляє собою безліч пов'язаних між собою програм, які забезпечують управління вузом в єдиному інформаційному просторі, та включає в себе модулі, що працюють в середовищі *Windows* (навчальний модуль, деканат, абітурієнт, методичний відділ, відділ кадрів тощо) та веб-портал (відображення розкладу занять, успішності, навчальних планів, нарахувань оплати за гуртожиток, контроль оплати за навчання та гуртожиток, тестування студентів, запис студентів на вивчення дисциплін т.д.). Вся інформація зберігається в одній спільній базі даних (рис.1.1). Сервіс, який призначений для ведення журналу успішності студентів надає можливість викладачеві виставляти та редагувати оцінки студентів з предметів, які він в них викладає. Студент має можливість переглядати свої оцінки зі всіх предметів. Для викладача існує три режими доступу до даного модулю:

- викладач має доступ тільки до своїх журналів;
- викладач має доступ до всіх журналів на кафедрі;
- викладач має доступ до всіх журналів на факультеті.

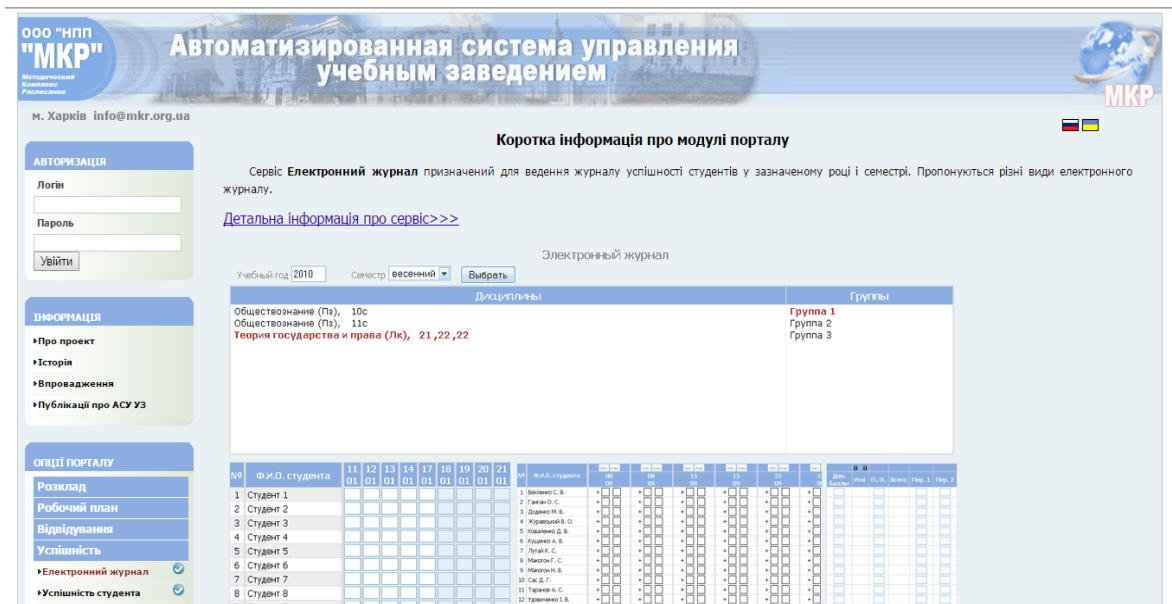


Рис. 1.1. Інтерфейс методичного комплексу розкладу

«Щоденник» – це всеукраїнська безкоштовна освітня мережа, формує унікальне електронне середовище для вчителів, учнів та їх батьків. Метою проекту є об'єднання всіх педагогів, учнів та батьків України в єдину спільноту, модернізація навчального процесу та впровадження сучасних комп'ютерних технологій у школах (рис. 1.2).

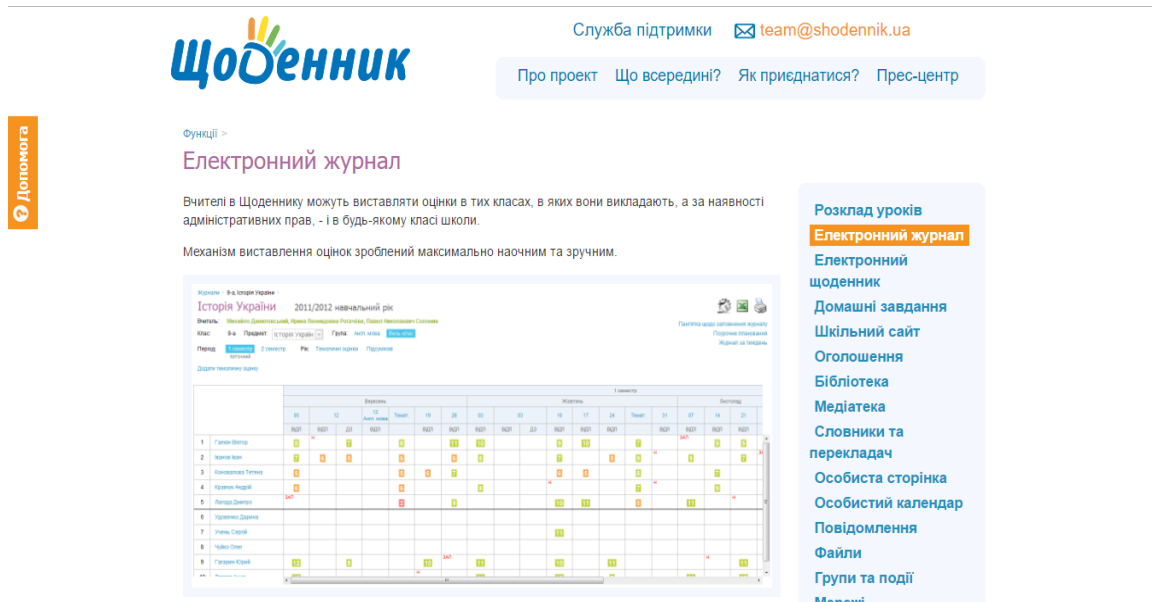


Рис. 1.2. Освітня мережа «Щоденник»

Основними функціями сервісу «Щоденник» є: розклад уроків, шкільний сайт, словники та перекладач, файли, електронний журнал, оголошення, особиста сторінка, групи та події, електронний щоденник, бібліотека, мій календар.



Також, особливо під час пандемії – значного розвитку набула система онлайн навчання від *Google* – «*Google Classroom*» (рис. 1.3).

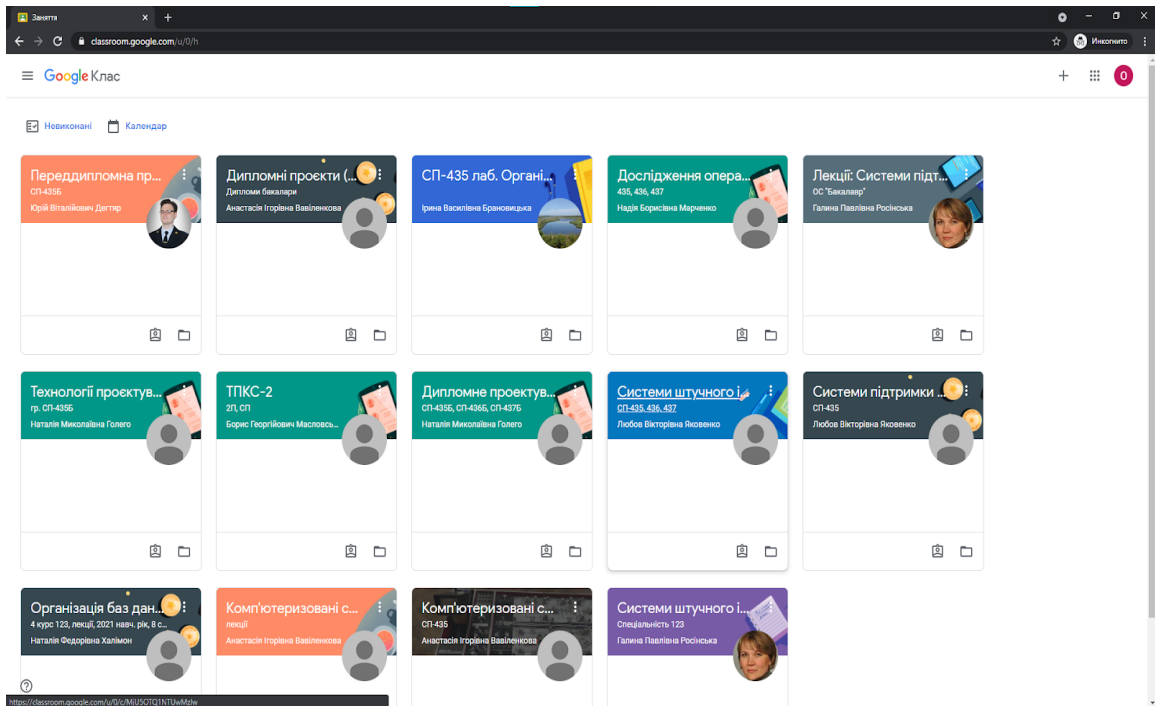


Рис. 1.3. Інтерфейс отримання інформації про активні *Google Classroom*

Серед переваг даної системи – величезні можливості корпорації *Google* із безліччю розробників, що дозволило інтегрувати навчальну платформу із вже існуючими сервісами в досить короткий час. Зокрема, із сервісом хмарного сховища *Google Drive* (використовується для зберігання робіт учнів), сервісом для відеозв'язку *Google Meet* (використовується для відеоуроків, лекцій, та ін.), *Google Calendar* (використовується для планування подій та занесення їх у календар користувача), *Gmail* (використовується для сповіщення користувача про відмітки, коментарі та ін.)

Серед недоліків – надскладний, інтуїтивно-незрозумілий інтерфейс, відсутність можливості переглянути усі оцінку в короткий строк. Наприклад, аби переглянути оцінки з певної дисципліни у ролі учня – потрібно зайти у відповідний курс, перейти на вкладку «завдання» для цього курсу, а потім перейти на підвкладку «Переглянути роботу». Таким чином можна отримати результати роботи по лише одній певній дисципліни. Відсутність можливості побачити усі результати на поточний момент (наприклад, за день або за поточний\минулий семестр).

Також, до недоліків можна віднести більшість інтегрувань із сервісами, адже коли освіта знов перейде у звичайний режим (а із завершенням пандемії так і відбудеться, тому, що із певних досліджень [5] «Назвіть проблеми, що найчастіше виникають в процесі реалізації дистанційного навчання»,— тут лише 1% респондентів стверджують, що не мають жодних проблем в процесі реалізації дистанційного навчання) – необхідність у використанні даних сервісів частково чи повністю зникне – а інформаційне нагромадження, яке давить на кожного користувача і знижує інтуїтивну зрозумілість – залишиться.

«Campus» - електронна система оцінювання у Національному Університеті «Київський політехнічний інститут» (рис. 1.4).

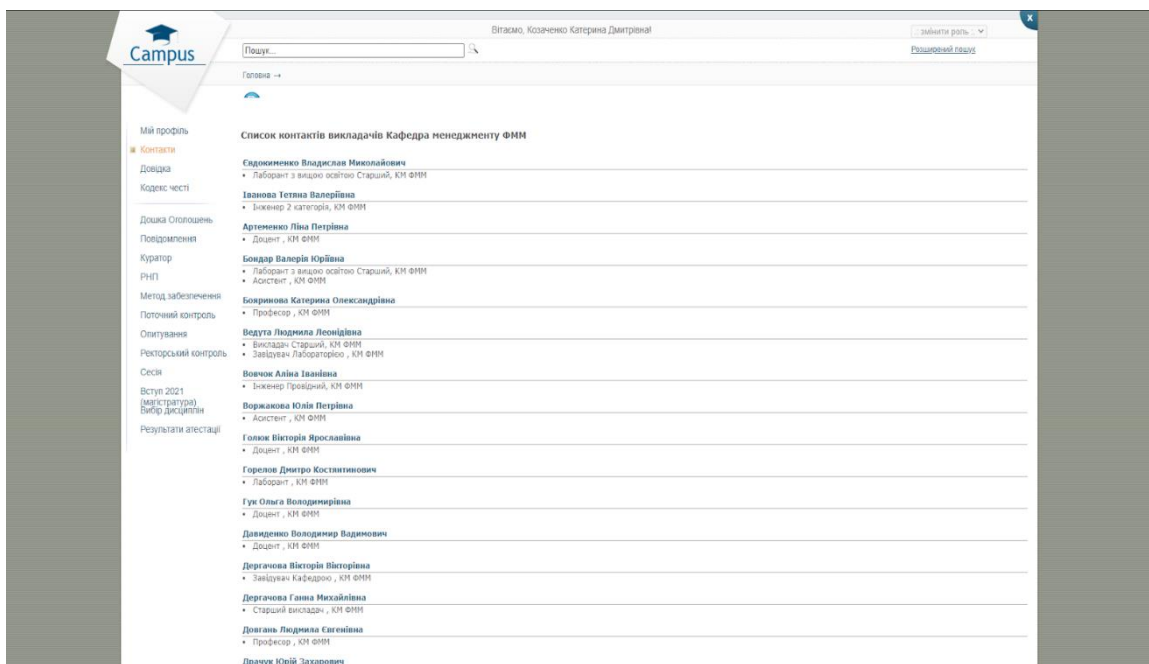


Рис. 1.4. Електронна система оцінювання «Campus»

Система моніторингу успішності. Відсутня можливість завантаження робіт. Присутні розподіл на категорії оцінювання, список поточних оцінок.

Серед вагомих недоліків – застарілий інтерфейс користувача.

Сьогодні – в час бурхливого розвитку інтернет технологій для освіти представлена велика кількість різних електронних журналів, як університетів, так і для шкіл. Особливою характеристикою таких систем є те, що вони використовують веб-технології та побудовані на «клієнт-серверній» архітектурі. Тому, як правило, для роботи з таким електронним журналом потрібен зв'язок з мережею Інтернет у момент безпосередньої роботи. Великими перевагами даного рішення є те, що для

роботи з таким електронним журналом достатньо мати планшет чи смартфон з активним підключенням до мережі Інтернету.

За надійне збереження даних в такому електронному журналі не варто хвилюватися, адже всі вони зберігаються на віддаленому сервері. І навіть, якщо щось трапиться з комп'ютером за допомогою якого безпосередньо працювали з електронним журналом, то дані залишаться збереженими та неушкодженими.

На сьогоднішній день кожна освітня установа стикається з великим потоком документообігу. Викладачі щодня виконують десятки маніпуляцій по відбору потрібної інформації, заповнюють журнал відвідування та успішності школярів.

В наш час інформаційні технології посідають почесне місце, так як більшість документів знаходяться у електронному вигляді. Саме тому новітні технології дозволяють розробити систему для контролю успішності школярів, автоматизувавши роботу журналу і зменшити роботу викладачів, чому і присвячено даний комплексний дипломний проєкт.

Для створення інтерактивного додатку для автоматизованого ведення електронного журналу необхідно:

- ознайомитися з особливостями електронного оцінювання у закладах освіти;
- проаналізувати роботу існуючих електронних журналів;
- проаналізувати архітектуру електронних журналів;
- дослідити технології обробки даних в електронних журналах для оцінювання роботи учнів;
- створити у команді інтерактивний додаток для автоматизованого ведення електронного журналу.

При заповненні журналу обліку проведення уроків класний керівник повинен надати таку інформацію:

- номер групи;
- прізвище, ім'я, по-батькові школяра;
- найменування предметів;
- прізвище, ім'я, по-батькові викладачів.

До обов'язків викладача входить:

- оцінювати рівень знань школярів;

- перевіряти присутність;
- виставляти оцінки за проведене заняття, тематичну чи семестрову контрольну роботу, курсову або реферативну роботу чи здачу індивідуальних завдань;
- занесення теми уроку до журналу;
- підведення підсумків успішності школяра.

При використанні електронного журналу успішності школярів всі вище перераховані дії класного керівника та викладачів можуть бути автоматизовані. Так як електронний журнал успішності – це інформаційна система, то список класу досить заповнити один раз, звільнивши тим самим викладачів від рутинної роботи, заповнювати кожного семестру списки школярів у журналі. Заповнивши всі дані про школярів один раз, не потрібно буде заповнювати їх кожного семестру, необхідно буде лише доповнювати, або виправляти їх при необхідності. Основною перевагою електронного журналу успішності є автоматизований процес виставляння семестрових оцінок, складання відомостей відвідуваності та успішності школярів.

При розробці електронного журналу успішності потрібно зробити акцент на те, щоб він був максимально простим та інтуїтивним у використанні. Головною метою при розробці інтерфейсу користувача є те, щоб зробити його зручним під час користування. Так як з даною системою будуть працювати люди різного віку, то інтерфейс повинен бути зрозумілий для кожного з них. Так як в електронному журналі буде зберігатися інформація про навчальну успішність школярів, то всі дані повинні бути захищені від несанкціонованого доступу та зміни.

### 1.3. Висновки до розділу

Електронне навчання надає можливість отримувати навички при різних причинах, які унеможливають очне навчання, збільшуючи мотивацію до отримання нових знань та задоволеність від процесу. Для фіксування досягнень раціонально використати онлайн журнал. Різні методи оцінювання дозволяють визначити, як слухачі засвоюють знання: лише поверхнево чи глибоко. Електронне

використання різних методів змушує слухачів вивчати явища з різних точок зору та розширює можливість ефективного навчання і ґрунтовного засвоєння навчального матеріалу. Таким чином, у процесі електронного навчання індивідуальне та групове оцінювання слід використовувати гармонійно, з урахуванням як альтернативних, так і класичних підходів до процесу оцінювання.

Аналіз додатків для урахування оцінок, показав, що більшість із них реалізовані у вигляді сайтів і мають занадто великий функціонал, або занадто простий. Більшість із них реалізовані для комерційного використання великих компаній і мають багато зайвої інформації такої, як реклама, вітання з днем народження користувачів цих інформаційних систем, конкретну базу даних для конкретного навчального закладу.

Вся логіка перевірки даних реалізована в серверній частині, що збільшує час відповіді від сервера, оскільки перед записом даних в базу даних, серверу потрібно відвести час на перевірку цих даних, а не лише записати дані в базу даних.

Недоліки, що наведені вище, були враховані при постановці задачі на розробку автоматизованої системи електронного журналу.

## РОЗДІЛ 2

### ТЕХНОЛОГІЇ ТА АЛГОРИТМИ ОБРОБКИ ДАНИХ У ІНТЕРАКТИВНОМУ ДОДАТКУ

#### 2.1. Клієнт-серверна архітектура інтерактивного додатку ведення електронного журналу

Під час роботи користувача з веб-додатком відбувається інтенсивна та постійна взаємодія частин проекту між собою. Інтерфейс користувача взаємодіє з серверною частиною. Серверна частина в свою чергу взаємодіє з базою даних. В такому процесі відбувається майже неперервний обмін інформацією між модульними частинами проекту. На рис. 2.1. представлені інформаційні потоки у веб-додатку.

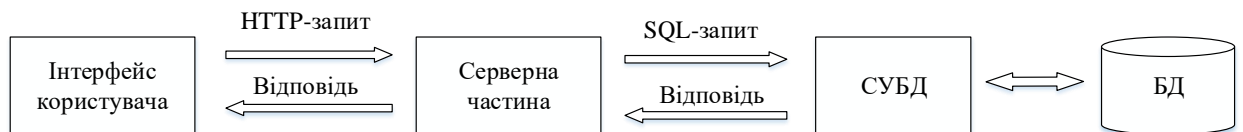


Рис. 2.1. Інформаційні потоки у веб-додатку

Користувач, під час взаємодії з інтерфейсом веб-додатку, взаємодіє з базою даних, що відбувається через серверну частину системи. З клієнтської частини робиться *HTTP*-запит до сервера на отримання або збереження певних даних. Після чого серверна частина додатку робить *SQL*-запит до системи керування базами даних (СКБД) на отримання або запис даних в базі даних. СКБД робить запис в базі даних або виконую пошук по ній та повертає результат пошуку до серверної частини, яка в свою чергу дає відповідь до клієнтської частини додатку. Такий спосіб забезпечує зручну роботу користувача з базою даних та ефективну роботу самої бази.

Кафедра КСУ				НАУ 21 18 12 000 ПЗ			
<i>Виконав</i>	Сефер О.В.			Технології та алгоритми обробки даних у інтерактивному додатку	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	Вавіленкова А.І.					22	54
<i>Консульт.</i>					СП-435 123		
<i>Норм. контр.</i>	Тупота Є.В.						
<i>Зав. Каф.</i>	Литвиненко О.Є.						

Для проектування електронного журналу була обрана архітектура клієнт-сервер. Загальна схема архітектури зображена на рис. 2.2.

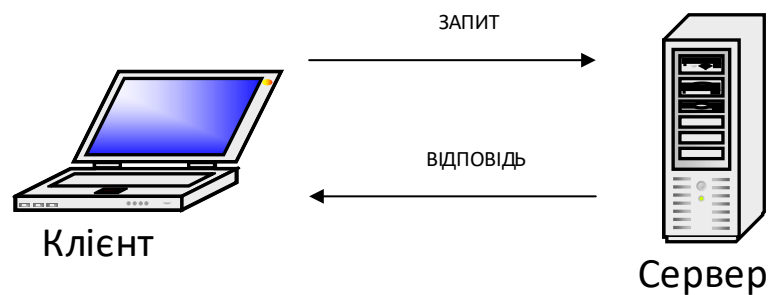


Рис. 2.2. Архітектура клієнт-сервер

Основна частина додатку при такій архітектурі знаходиться на сервері, що дозволяє зменшити вимоги до апаратної частини Клієнта, оскільки більшість обчислень та запитів на сторонні ресурси у такому випадку покладається на Сервер).

Клієнт – зазвичай браузер або мобільний додаток (рідше – інший сервер). Клієнт обмінюється даними із сервером за допомогою *HTTP* запитів. Сервер оброблює запит та повертає Клієнту відповідь. Такої архітектури вистачає лише для додатків, де Сервер – виступає у ролі проміжної ланки для обміну даними із іншими Серверами.

Веб-сервер – сервер, який приймає *HTTP* запити від Клієнтів, оброблює їх згідно заданої логіки та правил та повертає відповідь Клієнту. Веб-сервером називають програмне забезпечення, яке виконує функції веб-сервера, та комп’ютер, на якому це програмне забезпечення працює.

Електронний журнал має зберігати великі об’єми інформації (наприклад, оцінки учнів, їх персональні дані та ін.). У такому випадку доцільніше використати певну модифікацію даної архітектури, яка називається двохланкова архітектура (рис.2.3). Її особливість полягає у тому, що три базові компоненти розподіляються між двома ланками (клієнтом та сервером).

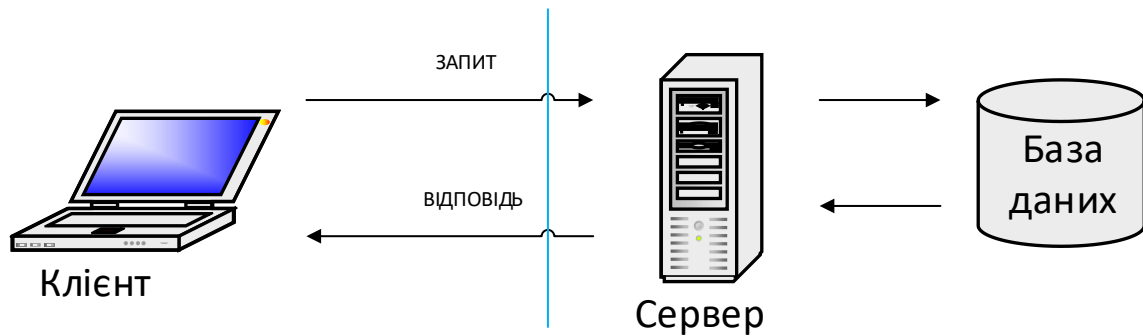


Рис. 2.3. Двохланкова архітектура клієнт-сервер

База даних – інформаційна модель, яка дозволяє впорядковано зберігати дані про об’єкти або групу об’єктів, які можна категоризувати. Фактично, база даних не є частиною веб-серверу, але більшість додатків зберігає у ній усі динамічні дані (облікові записи користувачів, дані користувачів, сесії та ін.). Для керування (запису, видалення, редагування та ін.) базою даних використовуються Системи Керування Базами Даних (СКБД). Бази даних поділяються на певні види, в залежності від цілей використання. Так існують реляційні бази даних, *NoSQL* бази даних, бази даних типу «ключ-значення» та інші. В залежності від виду бази даних – обирається СКБД для цієї бази. Найпопулярнішими наразі є *Oracle*, *MySQL* та *Microsoft SQL Server*. Графік популярності СКБД зображено на рис. 2.4.

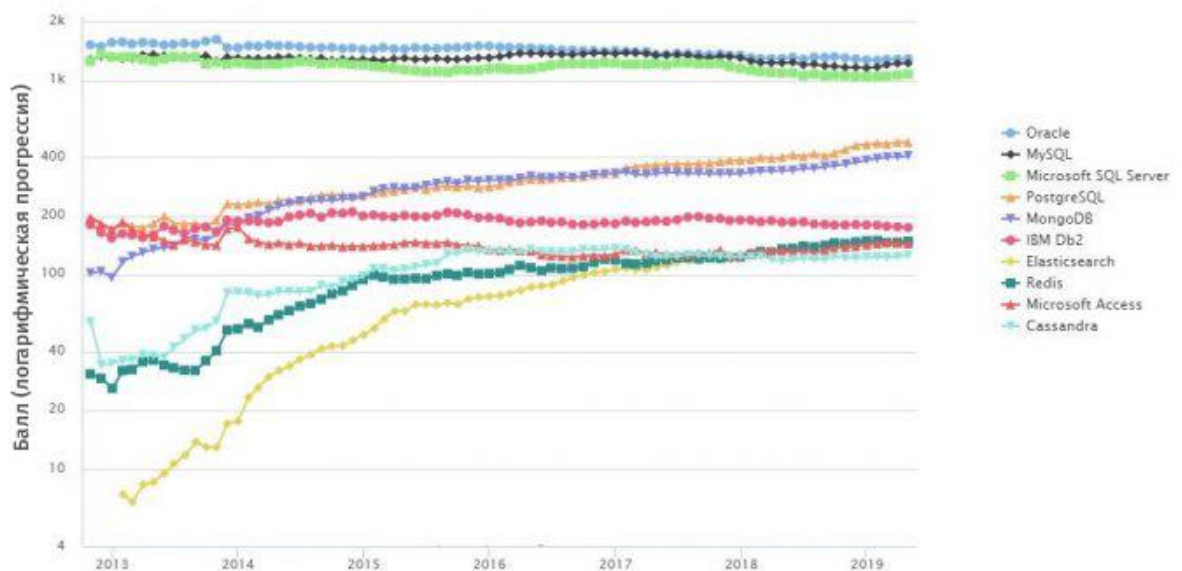


Рис. 2.4. Графік популярності СКБД

При двохланковій архітектурі вводиться поняття моделей «тонкого» та «товстого» клієнта. Модель «тонкого» клієнта характеризується тим, що вся логіка додатку та управління даними зосереджена на сервері. Клієнт у цьому випадку



виступає у якості рівень «представлення» (тобто користувацький інтерфейс). Модель «товстого» клієнта характеризується тим, що Сервер у такому випадку лише керує даними, а обробка інформація здійснюється на стороні Клієнта (при цьому, Клієнт і так само, як і при моделі «тонкого клієнта» виступає у ролі користувацького інтерфейсу).

## 2.2. Структура бази даних інтерактивного додатку

Проектування баз даних - процес створення схеми бази даних і визначення необхідних обмежень цілісності. Основні завдання проектування баз даних полягають в наступному:

- забезпечення зберігання в БД всієї необхідної інформації;
- забезпечення можливості отримання даних по всім необхідним запитам;
- скорочення надмірності і дублювання даних;
- забезпечення цілісності бази даних;

При створенні бази даних, необхідно прагнути впорядкувати інформацію по будь-яким ознаками, щоб після без проблем витягувати з неї необхідні дані в будь-якому їх поєднанні. Зробити це є можливим, тільки якщо дані структуровані.

Структурування – це певний набір узгоджень про різні способи представлення даних. Так само як класифікація речей може бути по декільком ознакам, наприклад, за кольором, розміром, функціоналу і т.д., так і структурувати дані можна по-різному. Розрізняють ієрархічну, мережеву, реляційну, об'єктно-орієнтовану і гібридну модель баз даних.

В даний час найпопулярнішою з усіх перерахованих вище моделей є реляційна, дана структура стала певним стандартом на поточний момент у процесі розвитку баз даних. На рис 2.5. зображено структуру бази даних Електронного журналу. База даних представляє собою набір таблиць, впорядкованих відповідно до певних обмежень та за певними критеріями. У якості СКБД було обрано *MySQL*.

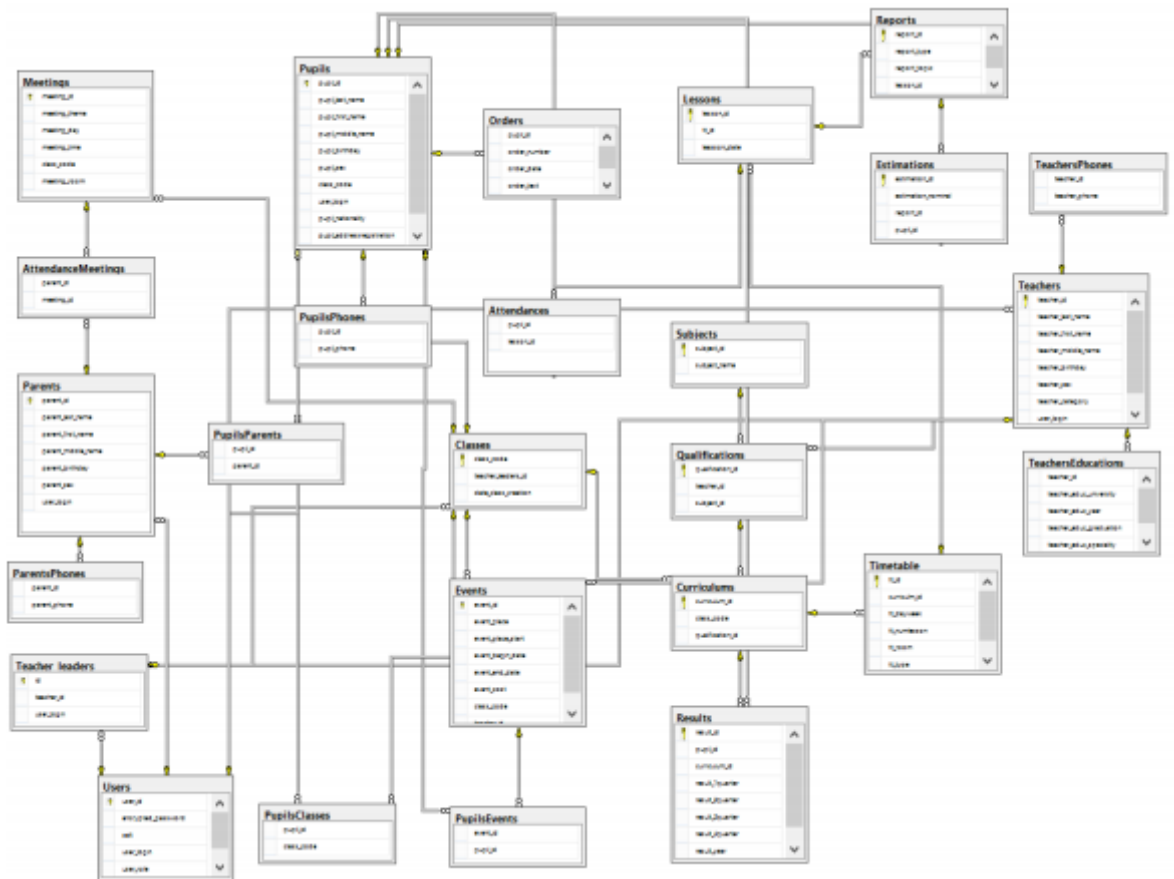


Рис. 2.5. Структура бази даних Електронного журналу

Було створено 23 таблиці, які забезпечують функціонування додатку (серед них таблиці Користувачів, предметів, учнів, вчителів, класів, розкладу та ін.). У якості прикладу далі додається розбір таблиці Користувачів *Users* (табл. 1.1).

Таблиця 1.1

Схема відношень Користувач (*Users*)

Зміст поля	Ім'я поля	Тип, довжина	Примітка
<i>id</i>	<i>user_id</i>	N(5)	ПК
Хеш паролю	<i>encrypted_password</i>	V(200)	Обов'язкове поле
Сіль для хешу	<i>salt</i>	V(200)	Обов'язкове поле
Логін	<i>user_login</i>	V(50)	Обов'язкове, унікальне поле
Роль	<i>user_role</i>	V(50)	Обов'язкове поле Значення: 'admin', 'teacher', 'pupil', 'class_head', 'school_head'

Позначки використані в таблицях:

- *PK* – первинний ключ;
- *C* – тип даних *Char*, у дужках розмірність;
- *V* – тип даних *Varchar*, у дужках розмірність;
- *T* – тип даних *Time*;
- *D* – тип даних *Date*;
- *N* – тип даних *Numeric*, у дужках розмірність.

Ця таблиця відповідає за ідентифікацію користувача у системі. Поле *ID* – унікальний ідентифікатор у системі. Хеш паролю – використовується для того, щоб зберігати пароль користувача у зашифрованому вигляді (це забезпечує додатковий рівень безпеки, адже якщо зловмисник і отримує базу даних – такий спосіб зберігання створить додаткові складнощі для розшифрування паролю Користувача). Сіль для хешу[8] – випадкова складова (слово), яка додається до алгоритму хешування для отримання різної строки із одного і того ж хеш-коду пароля. Логін – одна із складових для авторизації та автентифікації Користувача. Роль – змінна, яка відповідає за права Користувача, тобто в залежності від Ролі Користувача йому доступні або недоступні певні дії (наприклад, видалення іншого Користувача чи зміна оцінки).

### 2.3. Алгоритми авторизації користувачів та взаємодії з базою даних в інтерактивного додатку

Оскільки додаток має клієнт-серверну архітектуру було вирішено організувати авторизацію використовуючи *http* – запити, механізм *cookies*, та (обравши *framework* для серверної частини) – бібліотеку *Laravel Passport*.

*Cookies* - текстові або бінарні дані, отримані від веб-сайту на веб-сервері, які зберігаються у клієнта, тобто браузера, а потім відправлена на той самий сайт, якщо його буде повторно відвідано.

Таким чином веб-сервер помічає браузер користувача при відвідуванні. Куки створюються за ініціативою скриптового сценарію на стороні веб-браузера. При наступному візиті сервер буде знати, що користувач вже тут був. За допомогою

кукі-технології можна вивчити вподобання відвідувача. Кукі є одним із найточніших засобів визначення унікального користувача.

*Laravel Passport* - бібліотека для мови *PHP*, яка включає в себе імплементацію *OAuth 2-ої* версії (*OAuth* - це відкритий стандарт авторизації, який дозволяє користувачам відкривати доступ до своїх приватних даних, що зберігаються на одному сайті, іншому сайту, без необхідності вводу імені користувача та паролю) для *Laravel* додатків.

Схема алгоритму для авторизації користувачів у електронному журналі наведена на рис. 2.6.

1. Користувач вводить до форми входу свій *email* та пароль. На цьому кроці відбувається первинна валідація (перевірки на певний шаблон) вхідних даних. Наприклад, *email* користувача має відповідати наступному шаблону регулярного виразу:

$$\langle \backslash S + @ \backslash S + \backslash . \backslash S + / i \rangle$$

Необхідність первинної валідації обґрунтовується зменшенням кількості запитів до сервера. На етапі введення всіх даних через *JavaScript* перевіряється, чи можливий взагалі такий *email* (або пароль), адже у випадку, коли такі вхідні дані неможливі – набагато вигідніше навіть не відсилати запит до серверу, ніж очікувати його відповідь про помилку.

2. Наступним етапом є *HTTP-POST* запит до серверної частини додатку. Метод *POST HTTP* запиту дозволяє закодувати «тіло» повідомлення, на відміну, наприклад, від *GET* запиту

3. Наступним кроком сервер отримує дані (*email* та пароль) користувача. Шукає у базі користувача, якому би відповідав заданий *email*. Якщо користувача з таким *email* не було знайдено у базі – сервер повертає помилку із відповідним кодом (404 – *Not Found*). Користувач повертається на крок 1. Якщо користувача з таким *email* існує – переходимо до кроку 4.

4. Беремо хеш від паролю та зіставляємо його із записом у базі даних. Є два варіанти – співпадає або ні. Якщо хеш від паролю не співпадає – сервер повертає помилку 401 (*Not Authorized*). Якщо хеш співпадає – на сервері генерується токен (строка, яка служить ідентифікатором користувача для програми), який відправляється користувачу. Переходимо до кроку 5.

5. Коли користувач отримує токен - він (токен) зберігається у *cookies*.
6. До кожного наступного запиту автоматично додається *cookie* із токеном.

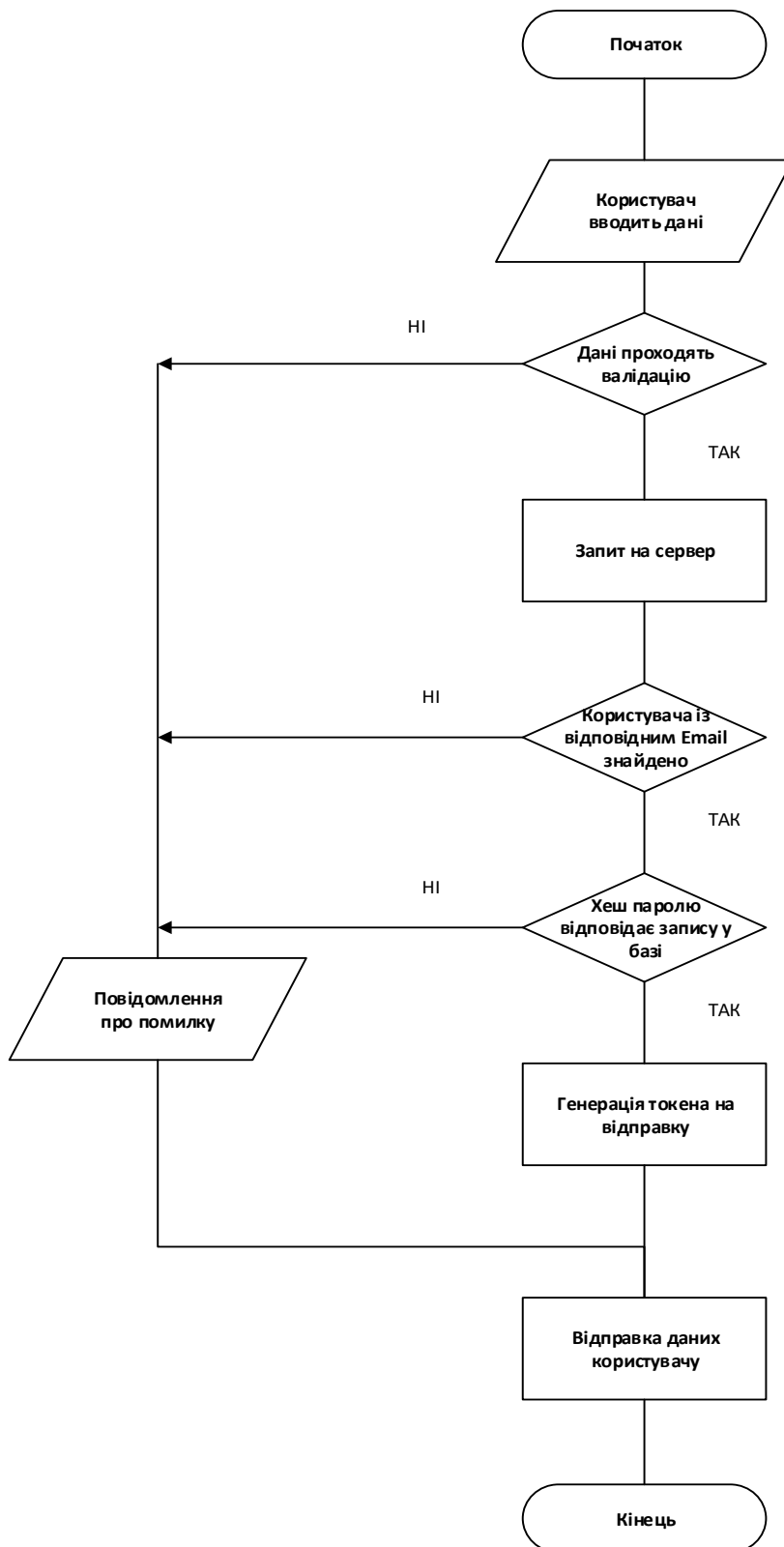


Рис. 2.6. Схема алгоритму авторизації користувачів у електронному журналі

Після вдалої авторизації у додатку Користувач отримує *JSON* із токеном для ідентифікації на сервері. Весь процес виглядає наступним чином.

Клієнт вводить свій логін та пароль у форму (рис.2.7), після цього натискає на кнопку увійти:

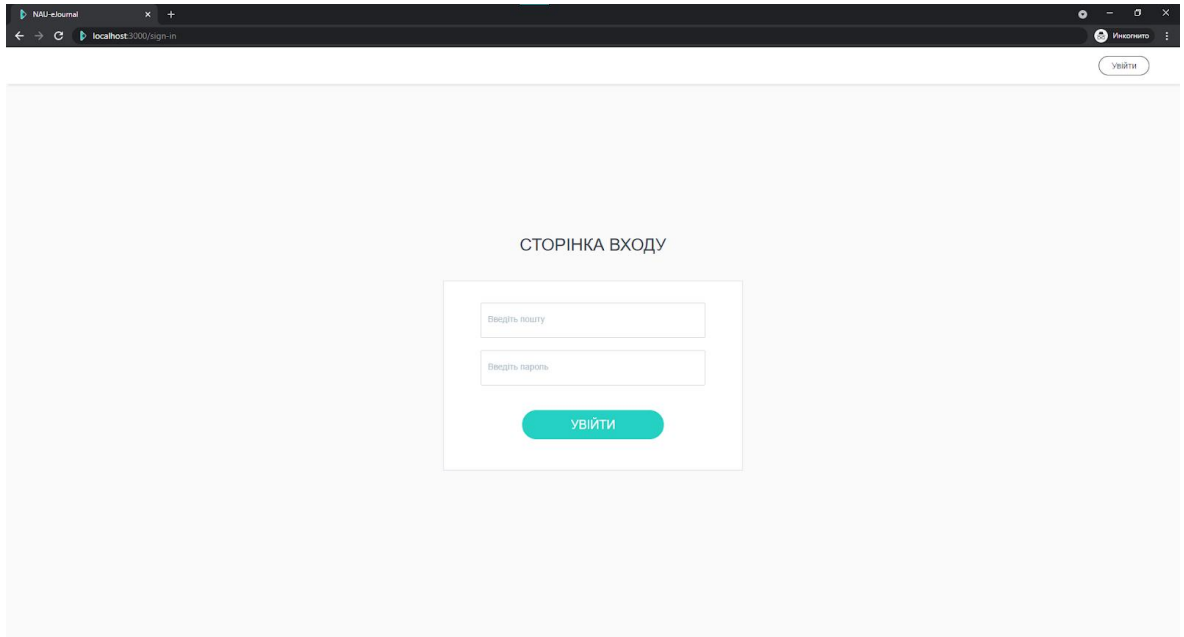


Рис. 2.7. Форма для входу у додаток

На Клієнті створюється запит на авторизацію:

```
export function auth(authData: AuthData): ThunkAction<any, any, any, any> {  
  return async (dispatch: ThunkDispatch<any, any, any>, {}, cookies: Cookies) =>  
  {  
    dispatch(authLoading(true));  
    let response = http.post('auth/admin-login', authData);  
    await response  
    .then((res: any) => {  
      const data = res.data;  
      cookies.set('token', data.accessToken, {path: '/'});  
      cookies.set('adminRead', data.adminRead, {path: '/'});  
      dispatch(authSuccess(data.accessToken, data.adminRead));  
    })  
    .catch((e: any) => {  
      const error = getErrorFromResponse(e.response);  
      dispatch(authError(error));  
    });  
  };  
};
```

```
}
```

Клієнт переходить у стан очікування відповіді.

Після чого, на серверній частині його оброблює спеціальний контролер:

```
<?php
```

```
namespace App\Http\Controllers\Api\Auth;
```

```
use App\Http\Controllers\Controller;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\Auth;
```

```
use Carbon\Carbon;
```

```
class LoginController extends Controller
```

```
{
```

```
/**
```

```
 * Handle the incoming request.
```

```
 *
```

```
 * @param \Illuminate\Http\Request $request
```

```
 * @return \Illuminate\Http\Response
```

```
 */
```

```
public function __invoke(Request $request)
```

```
{
```

```
    $credentials = $request->only('email', 'password');
```

```
    if (!Auth::attempt($credentials)) {
```

```
        return response()->json([
```

```
            'message' => 'You cannot sign with those credentials',
```

```
            'errors' => 'Unauthorised'
```

```
        ], 401);
```

```
    }
```

```
    $token = Auth::user()->createToken(config('app.name'));
```

```
    $token->token->expires_at = $request->remember_me ?
```

```
        Carbon::now()->addMonth() :
```

```
        Carbon::now()->addDay();
```





```
TOEGAs_yAA2hyND_rbYaIyg8b1C1-  
etLUMay1AsJMjGec6JD1mfW59vuvQLq9C91m1BOKV_hTORi5zt2X92KkBJVQINSwc  
8tahe5Oj0s_ppzBeH_escuN4VUQOHmQ4YkMykILxdCm8n98mEuQ1oe8NY-  
rs5VNzmN5HwLFC1YWct5i9ioSc7Ejm6l4AjS5v_Ux5tgpBbsyp1Tzpsive9JS-  
mmnUsPO_qL-bMZ2Bcmrr0MiXyWQ"
```

```
adminRead: true  
}
```

Після чого наступний код встановлює токен (та допоміжне поле) у *cookies*:

```
cookies.set('token', data.accessToken, {path: '/'});  
cookies.set('adminRead', data.adminRead, {path: '/'});
```

Надалі – при будь-якому запиті *cookies* будуть автоматично прив'язуватись до будь-якого запиту. Даний механізм описаний у коді Клієнту:

```
export const httpAuth = axios.create({  
  baseURL,  
  timeout,  
});
```

```
const cookies = new Cookies();  
httpAuth.interceptors.request.use(function (config: any) {  
  const token = cookies.get('token');  
  if (token == null) {  
    return config;  
  }  
}
```

```
const parsedToken = parseJwt(token);  
if (parsedToken == null) {  
  return config;  
}
```

```
const expDate = parsedToken.exp * 1000 - 60 * 1000 * 60; // treat as expired if  
less than 1h till the expiration  
if (Date.now() < expDate) {
```

```

config.headers = { Authorization: 'Bearer ' + token };
if (config.method === 'patch') {
  config.headers['Content-Type'] = 'application/merge-patch+json';
}

return config;
}

return issueToken()
.then((res: any) => {
  store.dispatch(tokenUpdate(res.data));
  config.headers = { Authorization: 'Bearer ' + res.data.token };

  return Promise.resolve(config);
})
.catch((e) => {
  return config;
});
});

```

Тобто будь-які запити будуть працювати згідно зі схемою, зображеною на рис.2.8.



Рис. 2.8. Схема запиту із прив'язкою *cookies*

Схема алгоритму взаємодії користувачів із базою даних, на прикладі отримання користувачем ролі «Учень» своїх оцінок за поточний день наведена на рис. 2.9.

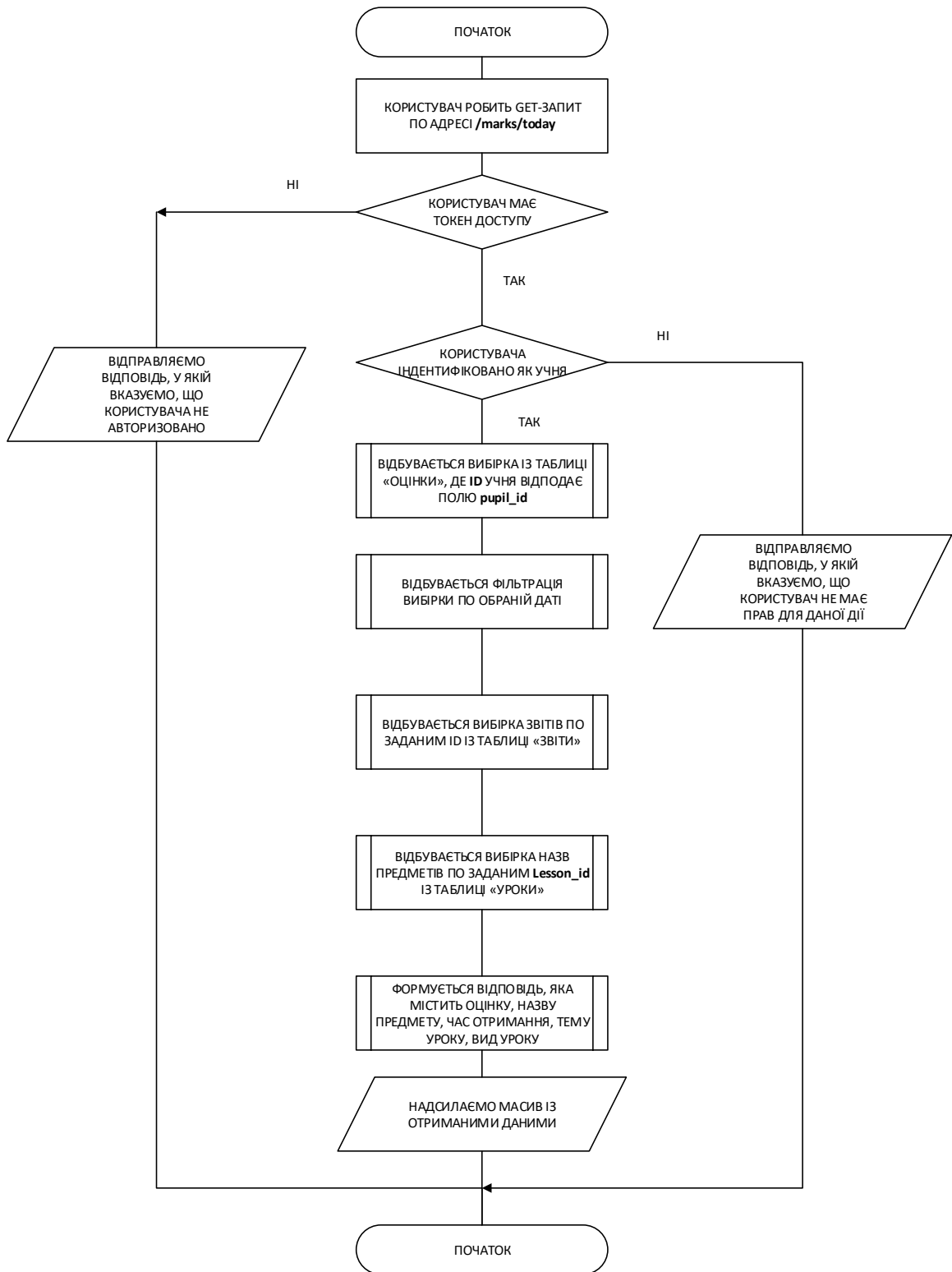


Рис. 2.9. Схема алгоритму взаємодії користувачів із базою даних, на прикладі отримання користувачем ролі «Учень» своїх оцінок за поточний день

## 2.4. Висновки до розділу

Після вираження порівняльної характеристики основних фігуруючих СУБД в проектах сьогодення було отримано інформацію про сильні сторони кожного

інструменту. Провівши аналіз було виділено три основних варіанти: *MSSQL*, *Oracle* та *MySQL* (яка і була обрана через ряд переваг).

Було розглянуто:

1. Архітектуру клієнт-серверного додатку, описана модель клієнта;
2. Розроблено, детально описано та застосовано алгоритм авторизації та автентифікації Користувачів у системі додатку;
3. Описано алгоритм взаємодії користувачів із базою даних.

### РОЗДІЛ 3

## РЕЗУЛЬТАТИ РОБОТИ ІНТЕРАКТИВНОГО ДОДАТКУ ДЛЯ АВТОМАТИЗОВАНОГО ВЕДЕННЯ ЕЛЕКТРОННОГО ЖУРНАЛУ

### 3.1. Структура інтерактивного додатку для автоматизованого ведення електронного журналу

Структурна схему інтерактивного додатку (рис.3.1) складається із наступних основних модулів:

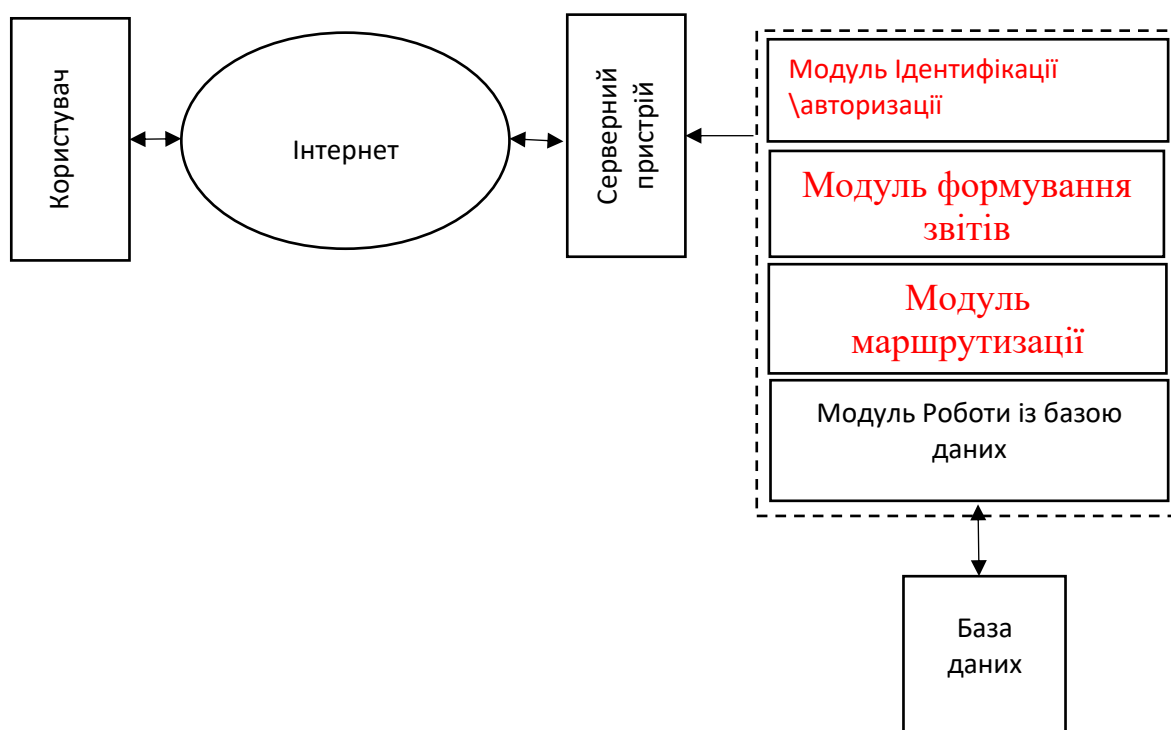


Рис. 3.1. Структурна схема інтерактивного додатку

1. Ідентифікації\авторизації
2. Формування звітів

Кафедра КСУ				НАУ 21 18 12 000 ПЗ				
Виконав	Сефер О.В..			Результати роботи інтерактивного додатку для ведення електронного журналу		Літера	Аркуш	Аркушів
Керівник	Вавіленкова А.І.						37	54
Консульт.				СП-435 123				
Норм. контр.	Тупота Є.В.							
Зав. Каф.	Литвиненко О.Є.							

3. *REST* Контролер, або модуль маршрутизації (відповідає за маршрутизацію додатку, відповідно до прав)

### 3.1.1 Модуль формування звітів

Основна робота модулю відбувається під час входу Користувача до системи (це процес описаний у пункті 2 другого розділу). Після цього цей модуль спрацьовує кожен раз, коли Користувач робить запит на захищений маршрут. Схема алгоритму роботи (рис. 3.2) цього модулю наступний:

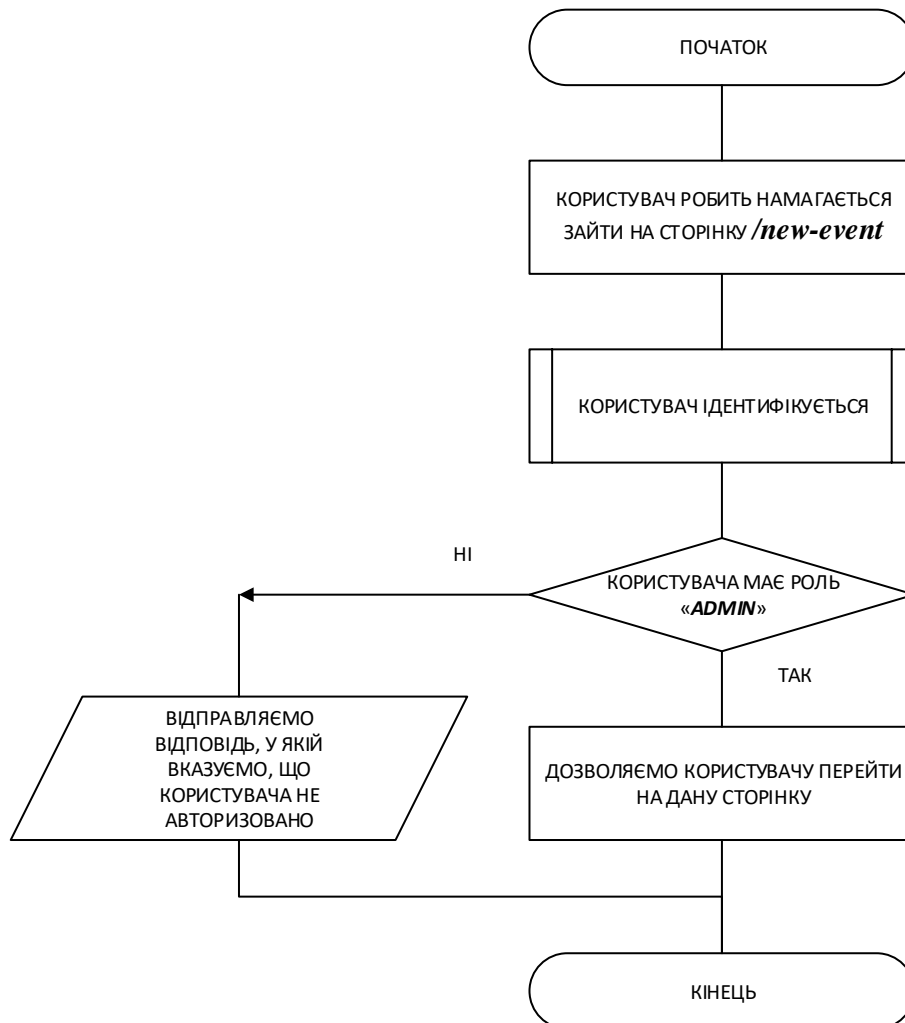


Рис. 3.2. Схема алгоритму роботи модулю ідентифікації

1. Користувач (нехай це буде «Учень») робить запит по маршруту /new-event, який відповідає за створення нового уроку.

2. Модуль розбирає *cookies* Користувача та присвоює йому відповідну роль (у даному випадку – роль Учень).

3. Маршрут /new-event є захищеним маршрутом. Для цього прописана так звана *middleware*:

```

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Api\Auth\RegisterController;
/*
|-----
| API Routes
|-----
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

```

```

Route::group(['middleware' => ['auth:api','role:admin']], function() {
    Route::post('/new-event', function(Request $request) {
        return $request->user();
    });
    Route::post('/new-event', [EventController::class, 'addEvent']);
});

```

```

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});

```

Доступ до нього має лише Користувач із роллю «*admin*».

4. Запит відхилено із кодом помилки «403 *Forbidden*» (Заборонено).

### 3.1.2 Модуль формування звітів

Даний модуль розташовується на клієнтській частині додатку. Таке розміщення даного модуля дозволяє зменшити навантаження на сервер, хоча в той самий час це збільшує навантаження на пристрій клієнта.

Схема алгоритму роботи модуля зображена на рис. 3.3. Даний алгоритм показує послідовність кроків для формування звіту успішність учня протягом останнього тижня:

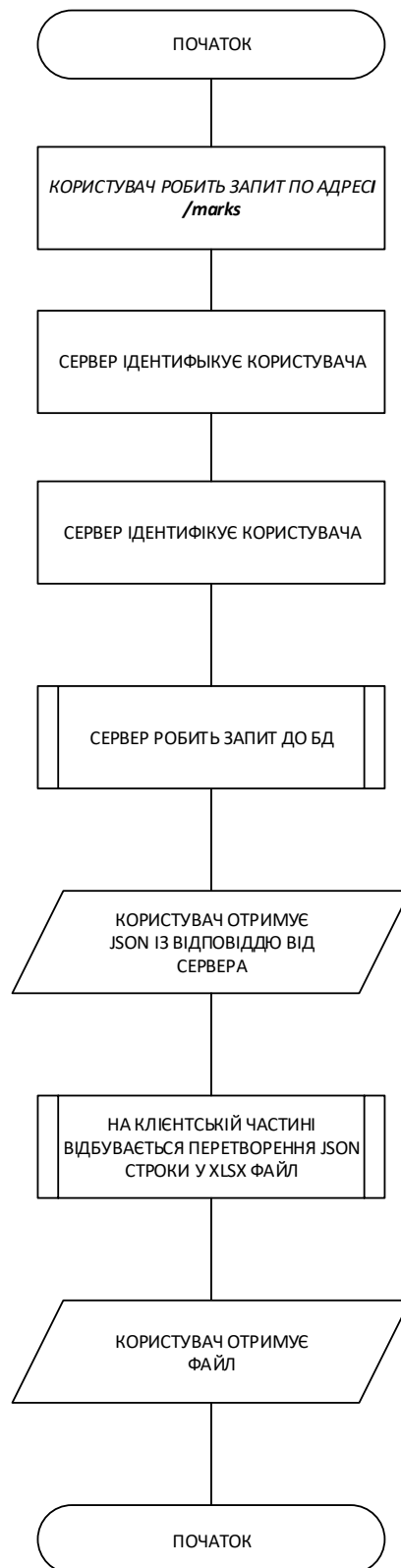


Рис. 3.3. Схема алгоритму роботи модулю формування звітів



### 3.1.3 Модуль маршрутизації

Даний модуль представляє собою набір маршрутів (роутів), які полегшують запити до сервера. Роути поділяються на захищені та публічні.

Одним із публічних роутів є роут *POST/ /login*:

```
Route::post('login', [RegisterController::class, 'login']);
```

При звертанні до нього викликається *RegisterController* (Додаток А), який обробляє відповідний запит.

Прикладом захищеного роуту є роут *GET/ /get-pupils*, який відповідає за видання всіх учнів певного класу:

```
Route::group(['middleware' => ['auth:api','role:teacher']], function() {  
    Route::get('/user', function (Request $request) {  
        return $request->user();  
    });  
    Route::get('/get-pupils', [FormController::Class, 'GetPupils']);  
});
```

Після виклику даного роуту викликається *FormController* (Додаток Б), який обробляє запит та викликає метод *GetPupils*:

```
public function GetPupils(Request $request){  
    return Pupil::where('class_code',$request->form_code)->get();  
}
```

Який, в свою чергу, робить *SQL* запит до бази даних аналогічний *SQL* коду:

```
SELECT pupil_last_name, pupil_first_name, pupil_middle_name, pupil_birthday,  
pupil_sex, class_code, user_login, pupil_nationality, pupil_address_of_registration,  
pupil_address_of_living,  
FROM pupils  
WHERE class_code = $request->form_code
```

### 3.2. Апробація інтерактивного додатку для автоматизованого ведення електронного журналу

Для перевірки працездатності програми зайдемо у додаток із акаунтом, який має прав адміністратора. Для цього заходимо на веб-сайт та вводимо відповідні дані у форму (рис.3.4):

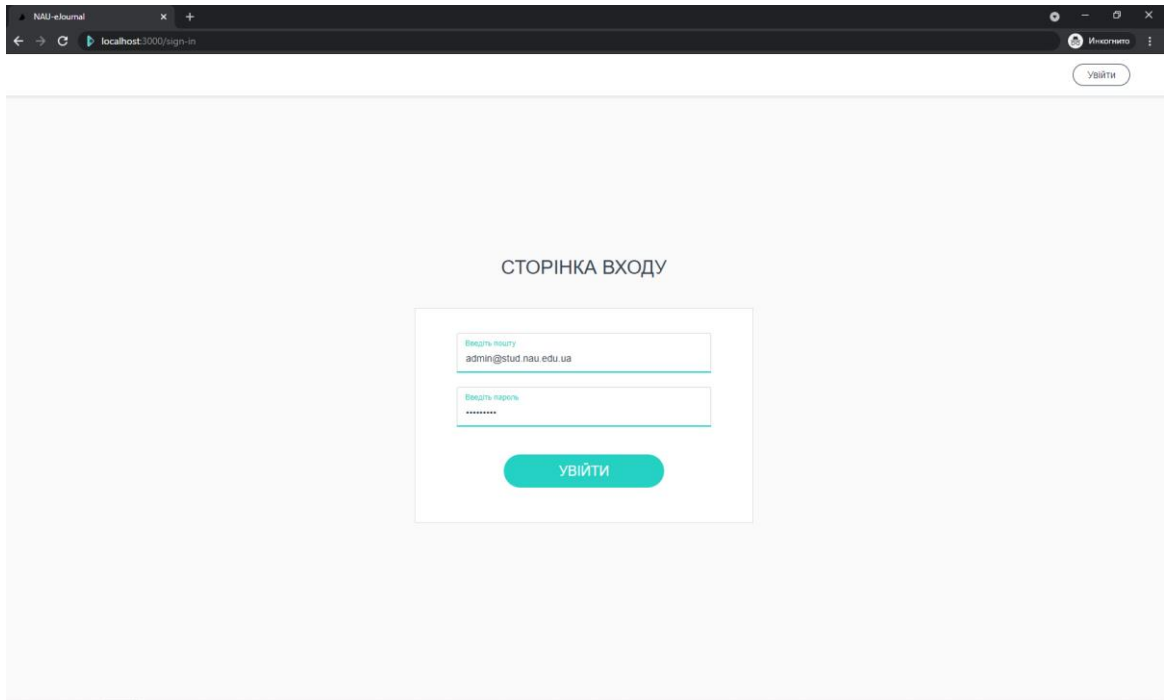


Рис. 3.4. Форма входу у додаток

Після натискання кнопки «Увійти» Користувача автоматично перенаправляє на відповідну домашню сторінку (рис.3.5) (у ролей «Батьки» та «Учні» – це оцінки за цей тиждень; у «Адміністратора» – сторінка «Класи»).

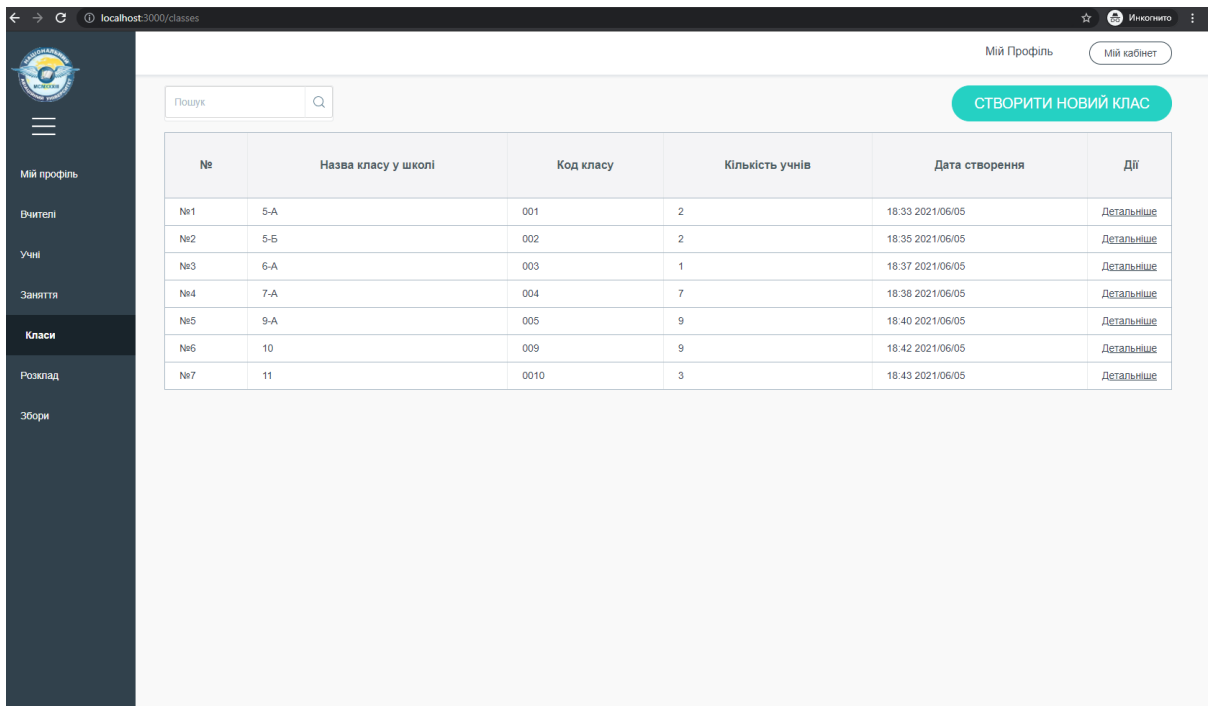


Рис. 3.5. Домашня сторінка адміністратора

Тут відображаються існуючі класи так короткий звіт по ним (кількість учнів у класі, дата створення, код класу у системі), також «Адміністратор» має можливість створити новий клас. Також «Адміністратор» має можливість перейти на сторінку класу (рис.3.6), для того, щоб побачити учнів цього класу. Для прикладу – перейдемо на сторінку 11-го класу.

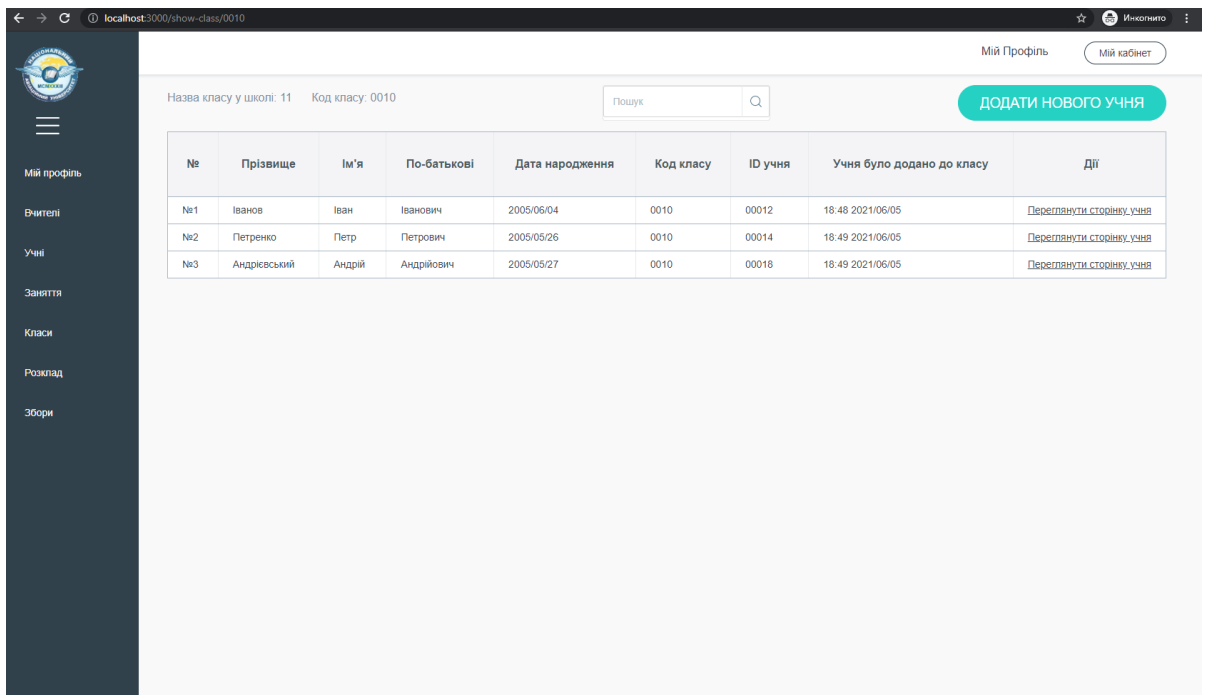
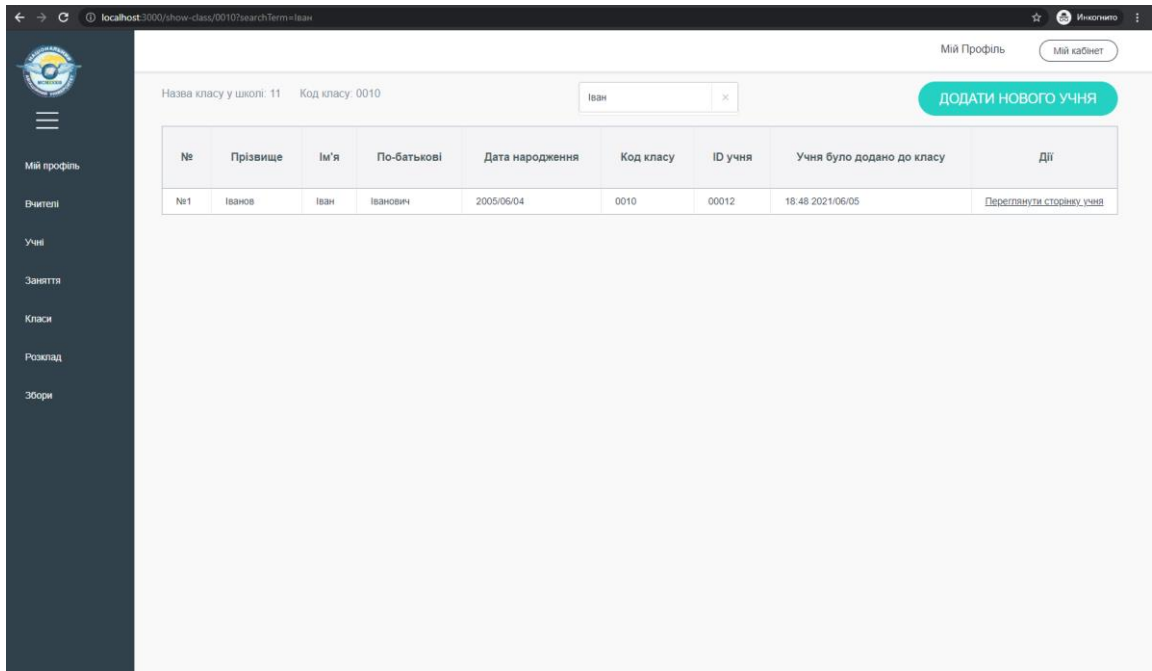


Рис. 3.6. Сторінка 11-го класу

На цій сторінці відображаються учні, їх дати народження, прізвища, ім'я та по-батькові, унікальні ідентифікатори у системі, код класу, до якого вони відносяться, посилання на їх сторінки у системі та дата додавання до класу. Також, тут знаходиться вікно пошуку за будь-яким критерієм. Для прикладу введемо слово «Іван» у пошуковий рядок та натиснемо «Enter» або на іконку пошуку.

У результаті пошуку (рис.3.7) отримуємо список учнів, які підходять під заданий критерій:

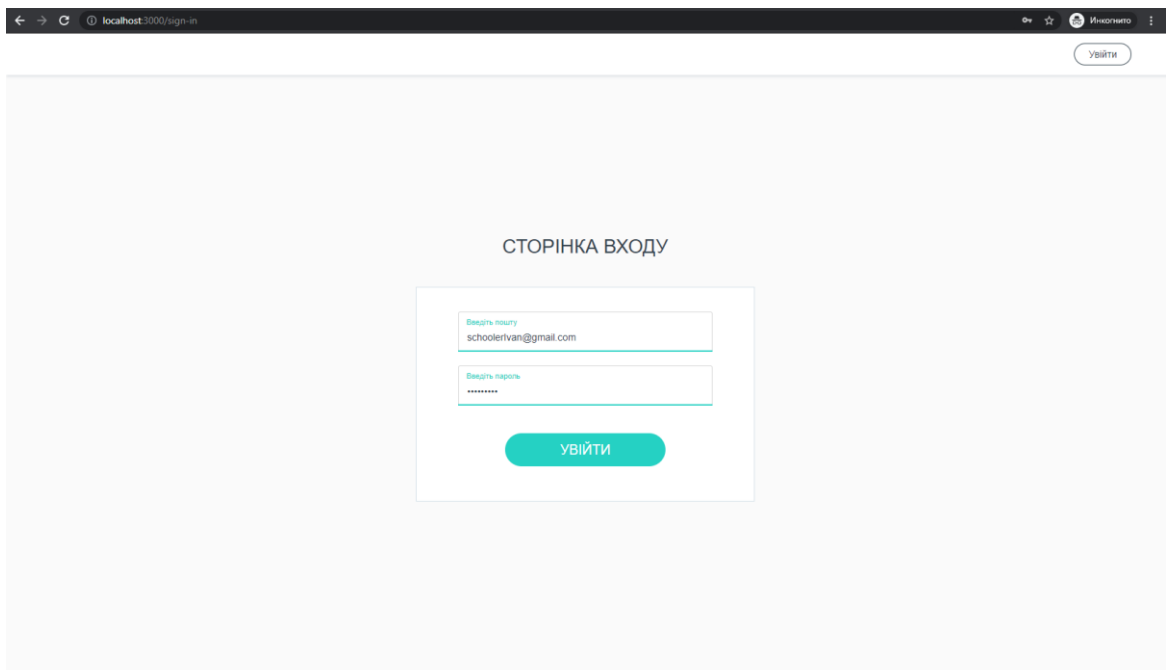


The screenshot shows a web browser window with the URL localhost:3000/show-class/0010?searchTerm=Іван. The page displays a search interface with a search bar containing 'Іван' and a 'ДОДАТИ НОВОГО УЧНЯ' button. Below the search bar is a table with the following data:

№	Прізвище	Ім'я	По-батькові	Дата народження	Код класу	ІД учня	Учня було додано до класу	Дії
№1	Іванов	Іван	Іванович	2005/06/04	0010	00012	18:48 2021/06/05	<a href="#">Переглянути сторінку учня</a>

Рис. 3.7. Результати пошуку за словом «Іван»

Перейдемо до тестування функцій Користувача. Для цього знов повертаємося на сторінку входу (рис.3.8) та вводимо відповідні дані:



The screenshot shows a login page titled 'СТОРІНКА ВХОДУ'. It features a central form with two input fields: 'Введіть ім'я' (Username) containing 'schoolerivan@gmail.com' and 'Введіть пароль' (Password) with masked characters. Below the fields is a teal 'УВІЙТИ' (Login) button. The browser address bar shows localhost:3000/sign-in.

Рис. 3.8. Форма входу у додаток

Після вдалої авторизації у системі Користувача автоматично пересилає на домашню сторінку (рис.3.9).

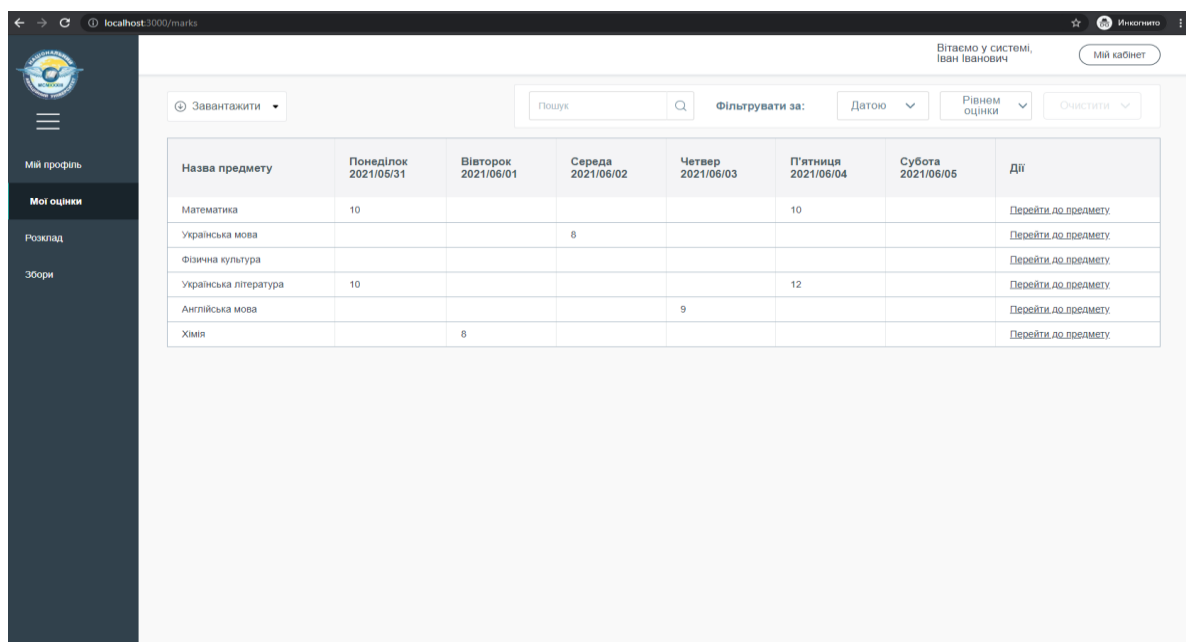


Рис. 3.9. Домашня сторінка учня

Користувач бачить свою успішність за цей тиждень. Також, Користувач має змогу перейти за відповідним предметом, натиснувши на посилання «Детальніше». Також, Користувач має змогу зробити пошук, наприклад за певною оцінкою. Користувачу доступна фільтрація оцінок за датою (рис.3.10) та за рівнем оцінки (рис.3.11)

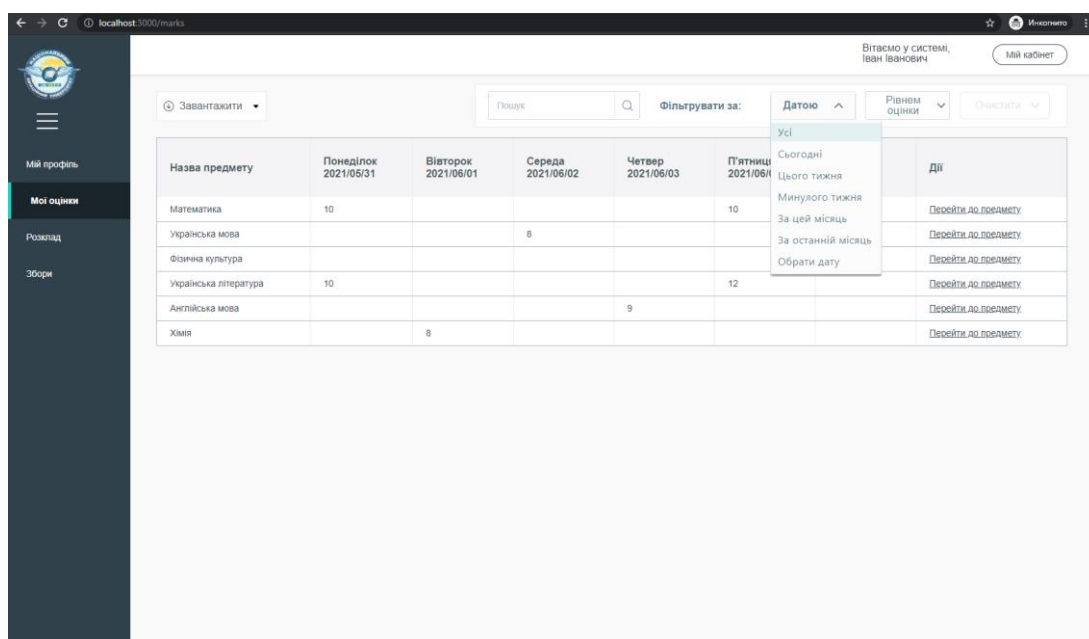


Рис. 3.10. Можливість фільтрації за датою

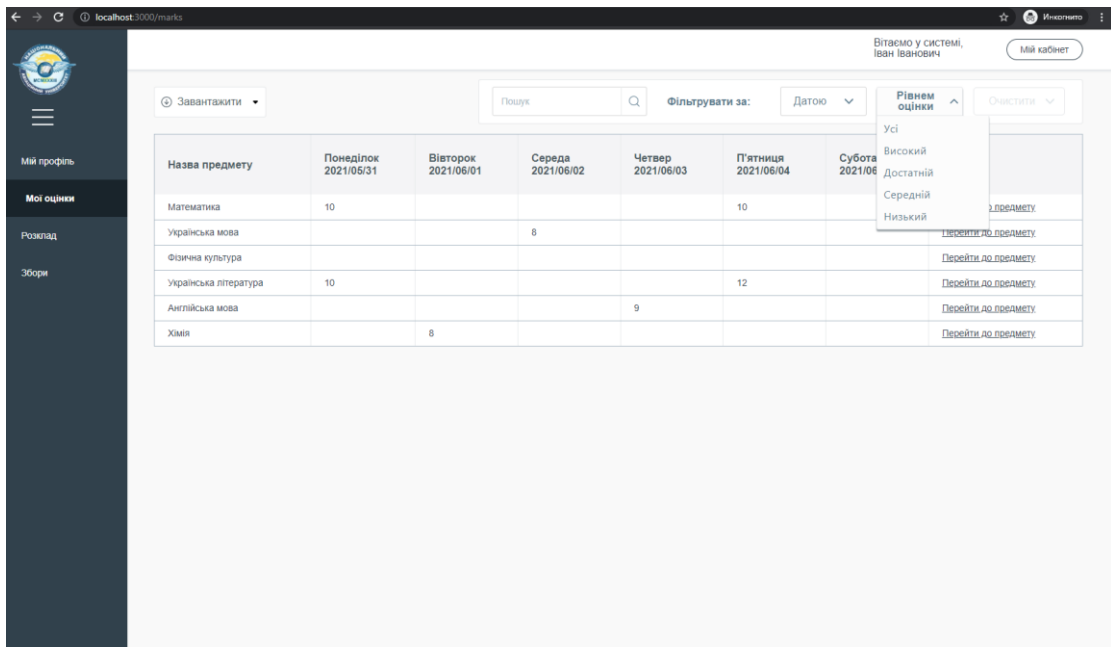


Рис. 3.11. Можливість фільтрації за рівнем оцінки

Також, Користувач має змогу завантажити свої оцінки за вибраний період часу, для цього натискаємо на кнопку завантажити (рис.3.12) та обираємо відповідний формат (наразі доступним є тільки *.xlsx*).

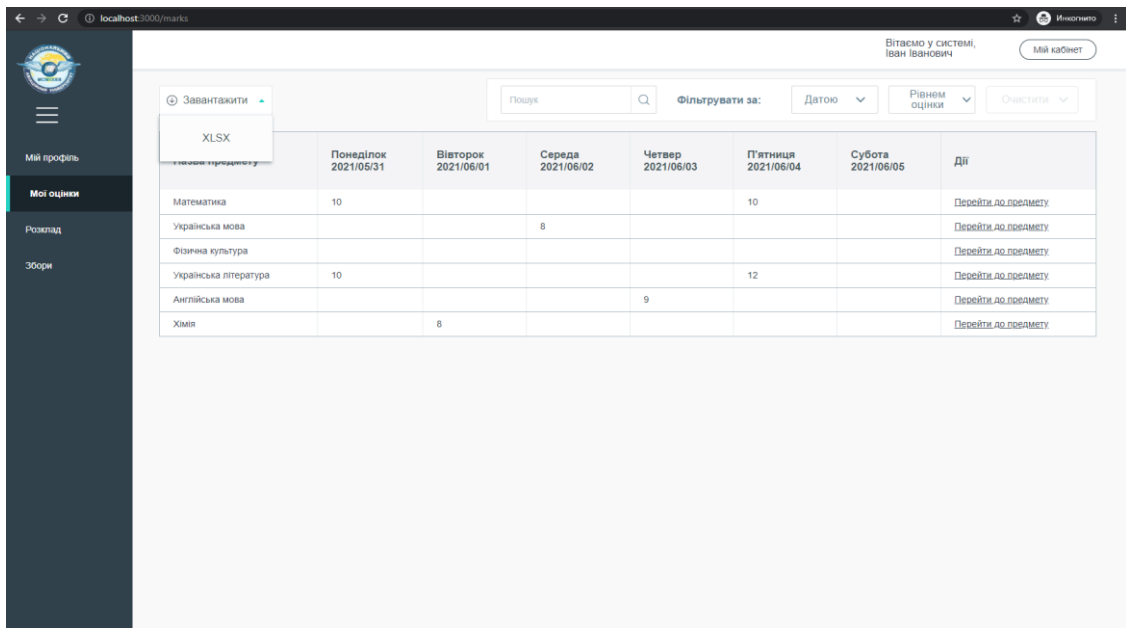


Рис. 3.12. Розкрите меню «Завантажити»

Після натискання на формат – починається процес завантаження (рис.3.13).

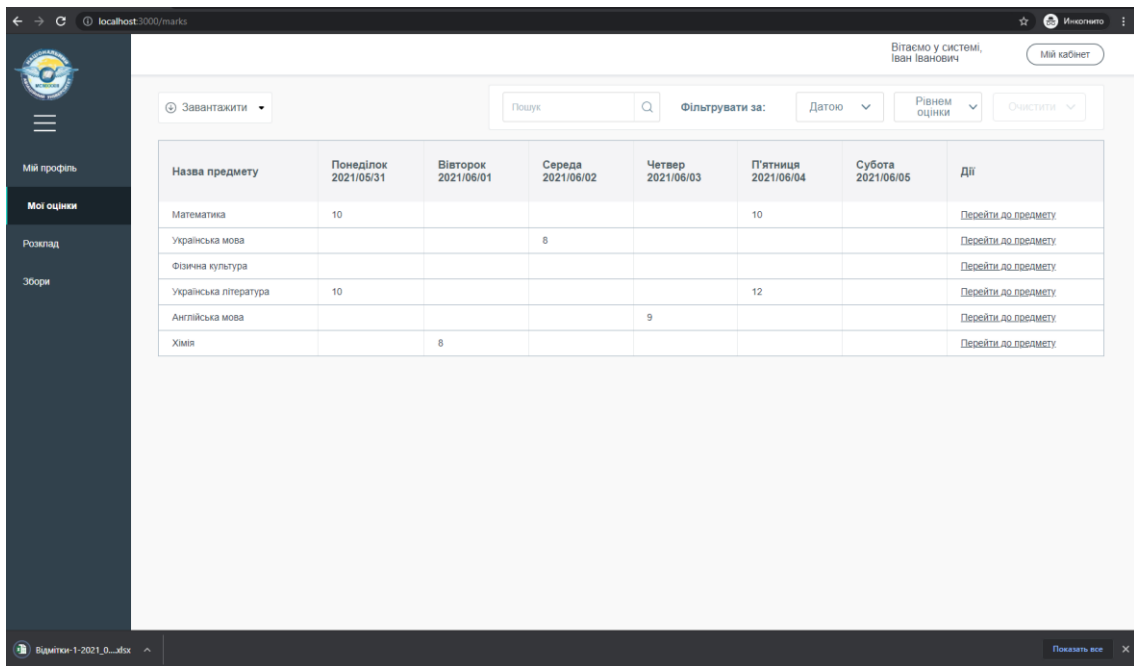


Рис. 3.13. Процес завантаження файлу

При закінченні процесу завантаження Користувач має змогу відкрити файл (рис. 3.14) для перегляду.

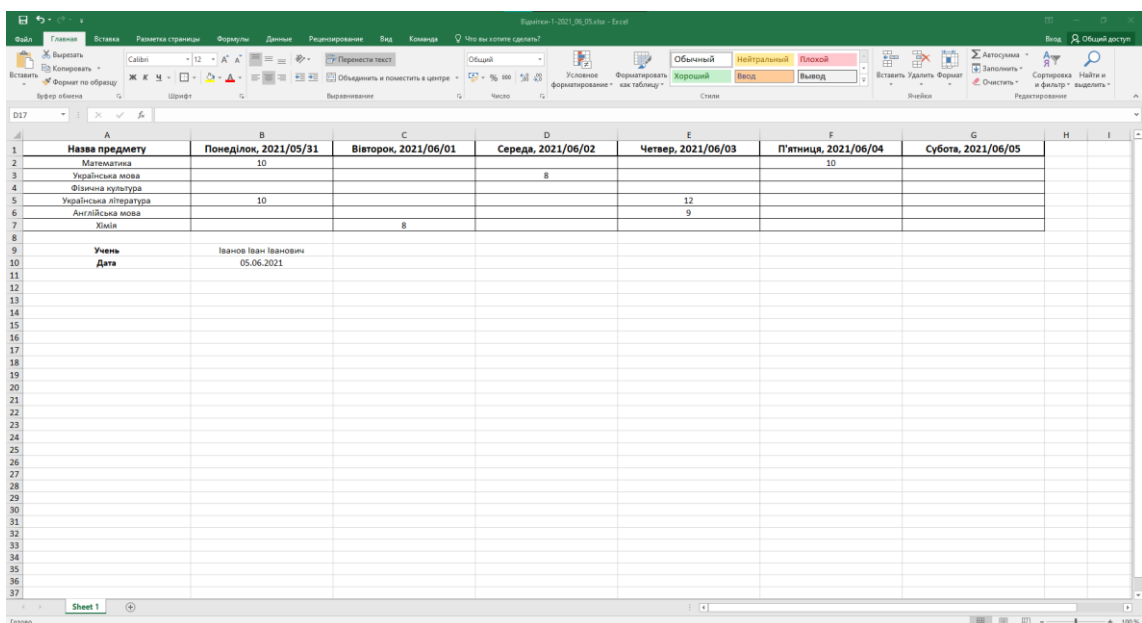


Рис. 3.14. Приклад файлу із оцінками

### 3.3. Кількісна оцінка параметрів роботи інтерактивного додатку для автоматизованого ведення електронного журналу

Важливим етапом розробки та реалізації додатку для автоматизованого ведення електронного журналу є оцінка його ергономічності, тобто визначення

ефективності даного продукту зі сторони зручності користування ним, що сприятиме створенню оптимальних умов роботи та підвищенню її продуктивності.

Для оцінки ергономічності розробленого проекту з боку його основних користувачів буде застосовано метод експертних оцінок. До проведення оцінювання було залучено 3 експертів, які є представниками основних трьох сегментів користувачів продукту: вчитель з досвідом роботи понад 25 років, учень 7 класу загальноосвітньої школи та один з батьків школяра.

Перед початком оцінювання було визначено основні показники ергономічності розробленого проекту, а саме широта функціоналу, розташування відомостей на сторінці, зручність навігації, оформлення та швидкість завантаження. Також було визначено коефіцієнти вагомості даних показників в діапазоні від 0 до 1 для отримання більш точних та коректних результатів.

Оцінювання проводилось за десятибальною шкалою, а результат визначався на основі середньозваженого значення. Результати оцінювання ергономічності даного проекту методом експертних оцінок наведено в таблиці 2.1.

Таблиця 2.1

#### Оцінка ергономічності проекту методом експертних оцінок

№	Назва показника	Вагомість показника	Середнє арифметичне значення оцінки	Середньозважена оцінка
1	Широта функціоналу	1,0	8,67	8,67
2	Розташування відомостей на сторінці	0,6	9,67	5,80
3	Зручність навігації	0,9	9,67	8,70
4	Оформлення	0,7	10,00	7,00
5	Швидкість завантаження	1,0	10,00	10,00
Всього				40,17

Отже, результат оцінювання ергономічності розробленого проекту шляхом застосування методу експертних оцінок продемонстрував позитивний результат, оскільки середньозважена оцінка ергономічності проекту склала 40,17 бали з 42 можливих. Варто зазначити, що серед усіх показників найнижче значення оцінки



було надане показнику широти функціоналу вчителем, що було обумовлене відсутністю можливості завантаження домашнього завдання для учнів.

### 3.4. Висновки до розділу

У третьому розділі було розглянуто роботу інтерактивного додатку для моніторингу та аналізу успішності школярів. Була розглянута структура додатку, описані основні структурні модулі проекту.

Також був проведений аналіз його ергономічності, тобто визначення ефективності даного продукту зі сторони зручності користування ним за методом експертних думок.

Були отримані оцінки експертів та враховані відповідні зауваження.

По результатам оцінок можна зробити висновок, що даний додаток дозволяє вирішити проблеми пов'язані із використанням паперових носіїв інформації, є зручним для використання у школах.

Програмна реалізація серверної частини виконана за допомогою мови *PHP*.

Модуль формування звітів для зменшення навантаження на сервер було перенесено на клієнтську частину додатку. Він був написаний на мові програмування *JavaScript*.

## ВИСНОВКИ

Під час виконання дипломного проекту було здійснено огляд систем моніторингу та аналізу успішності учнів, а також онлайн-платформ для навчання, визначено їх основні функціональні можливості, проведено їх порівняльний аналіз, визначено переваги та недоліки. Основні задачі, що вирішуються автоматизованими системами електронного оцінювання: зменшення паперового документообігу; підвищення мобільності усіх користувачів данної системи (батьки, учні, вчителі, адміністрація навчального закладу).

Під час виконання даної роботи були пройдені основні етапи розробки та проектування системи електронного журналу: аналіз предметної області, постановка задачі та вимог, проектування функціонуючого додатку, проектування користувацького інтерфейсу, проектування бази даних, інтеграції усіх модулів у єдину систему.

Окремим пунктом варто підкреслити аналіз результатів методом експертних думок. Було проведено опитання серед цільової аудиторії та визначено слабкі та сильні сторони даного проекту для кінцевого користувача.

Також під час розробки автоматизованого додатку були закріплені навички програмування на мовах, які були використані – *PHP, C#, JavaScript*.

Матеріали дипломного проекту рекомендується використовувати при проведенні наукових досліджень, у навчальному процесі середніх та вищих навчальних закладів.

## СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Поняття мови програмування [Електронний ресурс]. – URL: <https://ibrain.kz/informatika/ponyatie-yazyk-programmirovaniya>
2. Види мов програмування [Електронний ресурс]. – URL: <http://csaa.ru/vidy-jazykov-programmirovaniya/>
3. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.
4. ДСТУ ГОСТ 3008-95 «Документація. Звіти у сфері науки і техніки. Структура і правила оформлення».
5. Дослідження стану реалізації дистанційного навчання в Україні, *Prosvit*, [Електронний ресурс]. – URL: [https://nus.org.ua/wp-content/uploads/2020/05/Research2020\\_ProSvit\\_MF1.pdf](https://nus.org.ua/wp-content/uploads/2020/05/Research2020_ProSvit_MF1.pdf)
6. Макконнелл Дж. Основы современных алгоритмов – М.: Техносфера, 2004.– 368 с.
7. Зандстра, М. *PHP: объекты, шаблоны и методики программирования*, Третье издание = *PHP Objects, Patterns and Practice, Third Edition* / М. Зандстра — Москва: «Вильямс», 2010. — 560с.
8. Сіль для хешу [Електронний ресурс]. – URL: <https://www.internet-technologies.ru/articles/solenoe-heshirovanie-paroley-delaem-pravilno.html>
9. Суэринг, С. *PHP и MySQL. Библия программиста*, 2-е издание = *PHP 6 and MySQL 6 Bible* / С. Суэринг, Т. Конверс, П. Джойс — Москва: «Диалектика», 2010. — 912с.
10. Кузнецов, М. *PHP. Практика создания Web-сайтов*, 2 изд. / М. В. Кузнецов, И. В. Симдянов. – Санкт-Петербург: БХВ-Петербург, 2012. – 1264с.
11. Вуд, Л. *Web-графика*. / Л. Вуд – Санкт-Петербург: «Диалектика», 2001. – 488с.
12. Вагнер, Р. *JavaScript*. / Р. Вагнер, А. Вайк – Киев: ДияСофт, 2001. – 464с.
13. Дари, К. *AJAX и PHP. Разработка динамических веб-приложений* /
14. К. Дари, Б.Бринзаре, Ф. Черчерез-Тоза; Пер. с англ. – Санкт-Петербург: СимволПлюс, 2007. – 332с.

15. *Academic Writing Success | Academic Writing Coach Reviews*. (2018).  
[Электронный ресурс] <https://academiccoachingandwriting.org/dissertation-doctor/dissertation-doctor-blog/iv-the-structure-of-your-literature-review>

16. Джамса Крис. Эффективный самоучитель по креативному *Web*-дизайну. *HTML, XHTML, CSS, JavaScript, PHP, ASP, ActiveX*. Текст, графика, звук и анимация. Пер с англ./Крис Джамса, Конрад Кинг, Энди Андерсон - М.: ООО "ДиСофтЮП", 2005.- 672 с.