

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ ІНФОРМАЦІЙНОЇ СИСТЕМИ
ПРОЦЕСІВ АВТОМАТИЗАЦІЇ ДІЯЛЬНОСТІ КАФЕДРИ

(тема)

Виконав:
студент 2 курсу, групи ІНФМ-22-1

Пашук О.Г.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник к.т.н. Вечірська І.Д.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Кобилін О.А.
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Пашуку Олександрю Григоровичу
(прізвище, ім'я, по батькові)1. Тема роботи Методи та засоби побудови інформаційної системи процесів автоматизації діяльності кафедри

затверджена наказом по університету від 3 листопада 2023 року № 1280Ст

2. Термін подання студентом роботи до екзаменаційної комісії 24 грудня 2023 р.3. Вихідні дані до роботи літературні джерела, електронні ресурси, теоретичні основи та технічна документація з використання програмних засобів для побудови інформаційних систем.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз існуючих інформаційних систем та їх вплив на освітню сферу2. Дослідження можливих рішень та підходів для створення ефективної інформаційної системи.3. Проектування, реалізація та впровадження модуля ведення обліку та успішності студентів для системи автоматизації процесів діяльності кафедри «Електронна кафедра».

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми обробки зображень, постановка задачі, аналіз існуючих інформаційних систем, аналіз можливих варіантів рішень та стратегій для розробки інформаційної системи, вибір методології та реалізація модуля ведення обліку успішності студентів, результати дослідження та впровадження.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	03.11.2023	
2	Аналіз завдання, підбір літератури	03.11.23-05.11.23	
3	Аналіз літератури з досліджуваної проблеми	05.11.23-06.11.23	
4	Дослідження архітектури інформаційних систем	06.11.23-07.11.23	
5	Розробка програмного модуля	07.11.23-08.11.23	
6	Програмна реалізація	08.11.23-10.11.23	
7	Оформлення пояснювальної записки	11.11.23-12.12.23	
8	Перевірка на плагіат	10.12.2023	
9	Рецензування	18.12.2023	
10	Підготовка презентації та доповіді	20.12.2023	
11	Занесення роботи в електронний архів	03.01.2024	
12	Попередній захист кваліфікаційної роботи	03.01.2024	

Дата видачі завдання 3 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

_____ к.т.н. Вечірська І.Д.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 64 с., 8 табл., 16 рис., 50 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, АВТОМАТИЗАЦІЯ, ТЕХНОЛОГІЯ, ФРЕЙМВОРК.

Об'єктом дослідження є процеси автоматизації діяльності кафедри.

Метою дослідження є підвищення ефективності ведення обліку успішності студентів шляхом розробки та впровадження модуля «Електронний журнал» для системи автоматизації процесів діяльності кафедри «Електронна кафедра».

Використано методи порівняння та аналізу автоматизованих інформаційних систем. Проведено дослідження засобів та принципів побудови інформаційних систем автоматизації діяльності кафедри.

У результаті дослідження здійснено реалізацію системи автоматизації процесів кафедри «Електронна кафедра».

INFORMATION SYSTEM, AUTOMATION, TECHNOLOGY, FRAMEWORK.

The object of research is the processes of automation of the department.

The aim of this research is to increase the efficiency of accounting for student performance by developing and implementing the module «Electronic Journal» for the system of automation of processes of the department «Electronic Department».

Methods used for comparison and analysis of automated information systems. A study of the means and principles of building information systems for automation of the department's activities was conducted.

As a result of the analysis, the implementation of the process automation system of the department «Electronic Department» was carried out.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	6
Вступ.....	7
1 Аналіз існуючих інформаційних систем та їх вплив на освітню сферу..	9
1.1 Характеристика інформаційної системи та етапи розвитку.....	9
1.2 Принципи проектування інформаційних систем.....	16
1.3 Аналіз сучасних інформаційних систем в освіті.....	18
1.4 Автоматизація в освітній сфері.....	25
1.5 Постановка задачі дослідження.....	28
2 Дослідження можливих рішень та підходів для створення ефективної ІС.....	30
2.1 Дослідження архітектури інформаційних систем.....	30
2.2. Аналіз технологій клієнтської частини.....	36
2.2.1 HTML.....	36
2.2.2 CSS.....	37
2.2.3 JavaScript.....	38
2.2.4 Vue.js.....	38
2.3 GraphQL.....	40
2.4 Node.js.....	42
3 Проектування, реалізація та впровадження модуля ведення обліку та успішності студентів в інформаційній системі «Електронна кафедра».....	45
3.1 Вимоги до системи.....	45
3.2 Розробка програмного модуля.....	50
3.3 Тестування та опис роботи.....	53
3.4 Інструкція до використання.....	57
Висновки.....	59
Перелік джерел посилання.....	60

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

IC – інформаційна система

AIC – автоматизована інформаційна система

DPS – Data Processing System система обробки даних (система обробки даних)

MIS – Management Information System управлінські інформаційні системи
(управлінські інформаційні системи)

RPA – Robotic Process Automation роботизована автоматизація процесів
(роботизована автоматизація процесів)

AI – Artificial Intelligence штучний інтелект (штучний інтелект)

ВСТУП

Розвиток автоматизації, навіть у сфері освіти, є неминучою реальністю. Досі, багато чого не зрозуміло з теми впливу автоматизації на сферу освіти в майбутньому, але є також ряд значних переваг автоматизації, які вже можна побачити в сфері освіти. Найцінніше, що можна отримати від автоматизації – це додатковий час для взаємодії студентів та їх наставників – викладачів.

Такі поняття як автоматизація, штучний інтелект та робототехніка здатні проявити себе та внести вагомий внесок у сферу освіти в майбутньому. Це дозволить університетам адаптуватися до потреб студентів і забезпечити більш індивідуальне навчання, дозволяючи учням різного рівня кваліфікації працювати разом в одній аудиторії. Відстежуючи рівні та моделі знань учнів, автоматизація зможе допомогти викладачам виявити та усунути прогалини в їхньому викладанні. На додаток до оцінювання заповнених пробілів та оцінювання з кількома варіантами (що вже стало можливим завдяки автоматизації), ці технології зможуть допомогти педагогам оцінити запитання та есе з короткими відповідями.

Актуальність дослідження полягає у тому, що доступність та якість освітніх послуг при теперішніх умовах є достатньо відкритим питанням в розвитку української освіти світового рівня. Дані питання викликані потребою внесення глобальних змін, направлених на сприйняття розвитку освіти та підвищення її якості.

Розвиток інформаційних технологій в сучасних кондиціях стає одним із основних шляхів отримання вищої освіти, зручнішим та якіснішим транспортуванням даних до студента. Створити прямо зараз ефективну систему управління навчальним процесом та забезпечити комп'ютерний документообіг не можна досягти без впровадження та розробки єдиної системи управління освітою в закладі, як б містила в собі повний набір програмних методів та спільну базу даних.

Популярні системи управління навчальним процесом направлені на підтримку головних функцій навчання та не в усіх випадках мають в собі такі комунікаційні процеси як навчання, розробка, планування, аналізу, контролю та звітності. Моніторингові системи якості та системи дистанційного навчання виокремлюють серед основного списку відомих систем. На мою думку, тільки комплексні електронні інформаційні системи можуть бути ефективним інформаційним підґрунтям для різних видів користувачів системи – починаючи зі звичайного студента і закінчуючи ректором. Вибір та актуальність роботи була зумовлена саме цим.

1 АНАЛІЗ ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ ТА ЇХ ВПЛИВ НА ОСВІТНЮ СФЕРУ

1.1. Характеристика інформаційної системи та етапи розвитку

Інформаційна система – структурований набір елементів, які вирішують такі задачі як збір, обробку, передачу, зберігання та надання даних. Основа інформаційної системи включає в себе людей, обладнання, процеси, процедури, дані та операції. З наступних компонентів складається кожна інформаційна система:

- структура системи;
- функціонал елементів системи
- система в загалому, виходи та входи кожного окремого елементу;
- обмеження та мета системи та її складових.

Будова інформаційної системи містить в собі множину окремих її складових – підсистем. Окрема одиниця системи, яка виділена за конкретною ознакою називається підсистема. Набір певних підсистем являється структурою інформаційної системи та виконують функції технічного, інформаційного, математичного, програмного, організаційного і правового забезпечення [3].

Основу інформаційного забезпечення являє комплекс одиниць єдиної системи класифікації та заміну тексту повідомлень спеціальними символами, уніфікованих систем документації, схем інформаційних потоків, активно працюючих в організації, а також методологію побудови баз даних.

Сукупність технічних засобів, які використовуються при роботі інформаційної системи та їх документація і технологічні процеси називається технічним забезпеченням.

Для ефективної роботи усіх разом технічних засобів застосовується математичне і програмне забезпечення, яке складається з математичних

моделей, алгоритмів, методів та програм для вирішення задач та досягнення мети інформаційної системи.

За опис регламенту використання працівникам технічних засобів та взаємодії між собою при самій розробці та використанні інформаційної системи відповідає організаційне забезпечення.

Набір правових норм, що визначають початок роботи інформаційних систем, їх юридичний статус та діяльність, які регулюють порядок отримання, трансформації і використання даних називається правовим забезпеченням.

Існує класифікація інформаційних систем за певними критеріями: за різновидом задач або за функціональною ознакою; за вектором застосування (управління технологічними процесами, організаційного управління, автоматизованого проектування); за рівнем автоматизації (автоматизовані, автоматичні і ручні); за типом використання даних (інформаційно-управлінські та інформаційно-пошукові) [3].

Навчальні ІС застосовують для підготовки фахівців у сфері освіти, при навчанні або підвищенні кваліфікації працівників різних галузей.

Існує наступний перелік питань, які можна вирішити інформаційними системами за типом задач, які показано на рисунку 1.1:

- задачі, які можна відобразити у формі математичної моделі, яка має конкретний алгоритм розв'язку – структуровані;
- задачі, які не мають можливості складання в них математичної моделі – неструктуровані;
- задачі, які мають дані лише про декілька елементів та зв'язок між ними, а про залишок елементів дані додає вже людина – частково структуровані.



Рисунок 1.1 – Інформаційні системи за типом задач

Розподіл систем за функціональною ознакою виконується в залежності від мети системи, її основних функцій та завдань та загального призначення. Сукупність функціональних підсистем представляє собою головну структуру системи. Для прикладу, інформаційні системи фахівців можуть бути розділені на такі групи за функціональною ознакою: інформаційні системи обробки знань та системи офісної автоматизації, які показано на рисунку 1.2.



Рисунок 1.2 – Інформаційні системи за функціональною ознакою

Завдяки своїй простоті та універсальності інформаційні системи автоматизації офісу активно використовуються працівниками будь-якого

організаційного рівня [1]. Головною метою є обробка даних, підвищення ефективності роботи та спрощення діловодства. Дані системи виконують наступні функції:

- методами комп'ютерних технологій обробка відомостей на комп'ютерах;
- розробка нових видів та форм даних;
- збереження ресурсів та їх архівування;
- електронні календарі й зошити для організації обліку та ведення поточних даних;
- відеопошта (електронна, аудіо та відео);
- телеконференції та відео.

Експертні системи, у складі інформаційних систем обробки знань, в основі мають знання, потрібні при розробці або створенні людиною нового інформаційного додатку. Створення нових матеріалів та нових знань містяться в їх основній роботі.

Окрім того що інформаційна система відображає модель функціонування об'єкта управління, вона ще має вплив на органи управління. Це набір інформаційних процесів, які задовольняють потреби в інформації на різних рівнях прийняття рішень. Мета системи є виробленням інформації для споживання управлінською установою. Вона забезпечує нагромадження, передачу, збереження, оброблення та узагальнення інформації «знизу вгору», а також специфікацію інформації «зверху донизу» [5].

Головною метою інформаційної системи в освітній діяльності є збір та аналіз даних, пов'язаних з її управлінням та формування звітності. Органи управління завжди можуть покластись на своєчасно подану від неї інформацію для прийняття рішень, якість яких впливає на продуктивну діяльність об'єкта управління та його підрозділів. Головними завданнями є такі: знаходження нових джерел інформації; реєстрація, збір, обробка, передача інформації, що описує стан виробництва та управління; розмежування інформації між керівниками, їх підрозділами та виконавцями відповідно до їх ролі в

управлінні. Одними із ключових елементів організації є персонал, структура, робочі процедури, політика і культура. Інформаційна система являється також невід'ємним інструментом для вирішення управлінських питань.

Головними компонентами у системі обробки інформації є дані та обчислення. В кожній з них існують власні великі описи своїх змістів в тій чи іншій формі. Ці описи необхідні для інтерпретації і для коректного використання наданої інформації. Відсутність повного опису передбачає те, що користувачі можуть ознайомитись з ним з інших джерел [2].

Для головних компонентів інформації таких як дані та обчислення, важливе значення має така характеристика, як надмірність. Означення надмірності суттєво залежить від одиниці інформації. Коли одиниця вибрана, то надмірність – це просто копіювання цієї самої одиниці в системі. Важливим у виборі одиниці інформації є її розмір. Вибір занадто малої одиниці призводить до високого рівня самостійності блоків інформації, але в той же час і до росту накладних витрат на їх підтримку; якщо ви берете велику одиницю, не можна виключити кілька повторів блоків інформації.

В період створення та розвитку інформаційних систем організаційного типу структура і надмірність даних і обчислень значно змінювалась, за допомогою чого і відрізняли покоління цих систем.

В інформаційних системах першого покоління, які мають назву Data Processing System – DPS («Системи обробки даних»), або – «Автоматизовані системи управління – задачний підхід» – для кожної задачі спеціально готувалися дані і створювалася математична модель. Такий підхід викликав надмірність інформації (одні й ті ж дані можуть бути використані для вирішення різноманітних задач) і математичну надмірність (моделі вирішення різних проблем мали загальні блоки).

Системи обробки даних були вузько застосовуваними та зосередженими на автоматизацію робіт з паперами шляхом комп'ютеризації великих масивів і потоків даних на операційному рівні. Особливістю розпізнавання цих систем є ефективна обробка запитів, використання інтегрованих файлів для

зв'язування завдань один з одним, а також генерація зведених звітів для управлінського штату. Так як кожна система була спрямована на конкретну реалізацію, то опис її функцій (як правило, у вигляді друкованих посібників до процедур або у вигляді стандартів) подавався мінімально і призначався для фахівця в цій предметній області. Крім того, передбачалось, що користувачам притаманний відповідний досвід не тільки у прикладній галузі, а також при роботі із системами, які відповідають за обслуговування застосування.

Інформаційні системи другого покоління відомі під назвою Management Information System – MIS (управлінські інформаційні системи) або «АСУ – концепція баз даних». Забезпечення інформацією керівництва є головною функцією таких систем. Типова інформаційна система управління характеризується структурованим потоком інформації, інтеграцією завдань обробки даних, генеруючи запити та звіти.

При створенні великих розподілених баз даних основна проблема полягає у складності опису даних, що має на меті об'єктивне, незалежне від окремих прикладних додатків, спрощення колективного використання даних різними прикладними програмами. Словники даних та моделі широко використовуються для опису даних. До самостійних галузей досліджень додалась семантика даних.

АСУ показала свою економічну ефективність. Крім безпосереднього економічного ефекту впровадження АСУ справило чималий вплив на зміну характеру діяльності управління персоналу. Підвищення ефективності, наукової обґрунтованості та об'єктивності управлінських рішень; стало можливим вирішувати нові економічні проблеми, які до впровадження ІС не вирішувались керівником до впровадження ІС; збільшення часу на творчу роботу співробітників за рахунок зменшення обсягу рутинної кількості операцій; в результаті автоматизації процесів інформаційного обслуговування значно збільшилась інформованість управлінського персоналу.

Усвідомлення важливості розподілу обчислень в автоматизованих розрахунках виникло тоді, коли було помічено, що багато систем

використовують подібні обчислення, а окремі фактори, які реалізовані в додатках для допомоги визначеному користувачу, мають невелику різницю. Також, спостерігалось значне дублювання дій і процедур під час розробки, впровадження та тестування цих функцій.

АІС (автоматизована інформаційна система) – програмна система, що реалізує інформаційні технології у сфері управління у спільній роботі управлінського персоналу та набору технічних засобів.

Мета створення АІС – підвищення ефективності управління бізнесом. АІС реалізує типові функції управління: прогнозування, планування, облік, контроль, аналіз, регулювання. АІС призначена для автоматизованого збору, реєстрації, зберігання, пошуку, обробки та видачі інформації на запит користувачів. Він складається з сукупності взаємопов'язаних компонентів: технічних засобів обробки та передачі даних, методів і алгоритмів обробки у вигляді відповідного програмного забезпечення, наборів даних на різних носіях, персоналу та користувачів, об'єднаних організаційними, структурними, тематичними, технологічними чи іншими характеристиками для виконання автоматизованої обробки даних для задоволення інформаційних потреб споживачів інформації. Визначальним є принцип орієнтації розробленого технічного, програмного, інформаційного забезпечення на потреби конкретних користувачів – управлінського персоналу, який є учасниками процесу управління. АІС – це людино-машинна система, вона дозволяє підвищити якість управління за рахунок оптимального розподілу праці між людиною і комп'ютером на всіх етапах управління. Увесь комплекс операцій з обробки інформації, включаючи збір, введення, запис, реєстрацію, перетворення, зчитування, збереження, знищення, коригування, обмін по каналах зв'язку, в АІС здійснюється за допомогою апаратних і програмних засобів.

1.2. Принципи проектування інформаційних систем

В даний час інформаційна індустрія стала новою галуззю технологій, що приносить користувачам велику вигоду. Тому у сучасних умовах керівник організації повинен володіти знаннями методичних основ створення ІС. Знання методичних основ створення і використання ІС тісно пов'язано з розвитком і вдосконаленням управлінських процесів [14].

Основними принципами єдиних методичних підходів є:

– принцип системності, який є найважливішим при створенні, функціонуванні і розвитку ІС. Він розглядає досліджуваний економічний об'єкт, як єдине ціле. При цьому, встановлює напрями виробничо-господарської діяльності організації і конкретні функції, що реалізуються нею; виявляє різноманітні типи зв'язків між його структурними елементами забезпечуючи цілісність системи. Принцип системності припускає проведення в організації двох аспектного аналізу, а саме макро- і мікроаналізу. При макроаналізі система та її елементи розглядаються як частина системи вищого порядку. Особлива увага приділяється інформаційним зв'язкам: встановлюються їх напрями руху, виділяються і аналізуються ті зв'язки, які обумовлені метою функціонування і дослідження об'єкту, а потім вибираються найбільш переважні, що враховуються в процесі проектування ІС. При мікроаналізі вивчаються всі аспекти діяльності організації, аналізуються її структурні складові (включаючи діяльність на кожному робочому місці) з метою їх функціональних характеристик, що виявляються через зв'язки з іншими елементами і зовнішнім середовищем [12];

– принцип розвитку, який полягає в тому, що ІС створюється з урахуванням можливості постійного поповнення і оновлення функцій системи і видів її забезпечення. Його суть в тому, що розвиваються виробничі і управлінські процеси, ускладнюються і перебудовуються організаційні структури економічних об'єктів – це викликає необхідність нарощування обчислювальних потужностей ІС, оснащення їх новими технічними і

програмними засобами для постійного поповнення і оновлення вирішуваних завдань, розширення інформаційного фонду, що створюється у вигляді баз і сховищ даних, баз знань;

– принцип інформаційний, який направлений на детальне і всебічне вивчення інформації і інформаційних процесів, що супроводжують процеси управління в економічному об'єкті. Інформація вивчається в семантичному, синтаксичному і прагматичному аспектах. Крім цього, вивчення інформації необхідне для проектування автоматизованих робочих місць, систем передачі, зберігання і обробки даних, захисту інформації, де основними є знання об'ємів, змісту, і корисності відомостей. В теперішній час на інформаційному підході ґрунтується об'єктно-орієнтований метод моделювання інформаційних процесів і автоматизації проектувальних робіт для аналізу управлінських процесів і проектування інформаційних електронних потоків;

– у забезпеченні взаємодії інформаційних систем різних видів, цілей, рівнів при функціонуванні об'єкта полягає принцип сумісності. Єдність методичних підходів відображено в нормативних документах, що регламентують процес розробки, документування, приймання та експлуатації ІС. Це міжнародні та вітчизняні стандарти (ДСТУ), галузеві та відомчі положення, положення, протоколи, стандарти організацій. Добре поширені стандарти, що регулюють мову обробки інформації, технологію комунікацій та організацію обчислень;

– принцип стандартизації і уніфікації, який полягає в необхідності застосування типових, уніфікованих і стандартизованих елементів функціонування ІС. Це перш за все відноситься до складових інформаційного, технічного, програмного і інших забезпечувальних підсистем ІТ. Цей принцип дозволяє скоротити часові, трудові і вартісні витрати на створення ІС при максимально можливому використанні накопиченого досвіду у формуванні проектних рішень і впровадженні автоматизації проектувальних робіт, забезпечує багатоаспектну взаємодію ІС;

– принцип декомпозиції, який заснований на розділенні системи на частини і виділенні окремих комплексів робіт, що створює умови для ефективнішого аналізу існуючого стану управлінської діяльності, вивчення особливостей вирішення функціональних завдань для подальшого моделювання конкретних аспектів управлінської діяльності і перекладу їх на автоматизовану технологію. Принцип використовується як при вивченні особливостей властивостей елементів і системи в цілому, так і при створенні ІС на новій інформаційно-технологічній базі;

– принцип ефективності, ціль якого досягається за допомогою раціонального співвідношення між витратами на створення ІС та цільовим результатом, що отримується в ході її функціонування.

На основі аналізу поняття інформації, ролі та призначення інформаційних ресурсів, взаємодії інформаційних систем на різних рівнях управління, існуючих теоретичних положень, методичних підходів та практичних рекомендацій щодо проектування, експлуатації та удосконалення автоматизованих систем, загальні тенденції, розробив систему принципів трьох груп: побудова, експлуатація та розвиток інформаційних систем у державному управлінні. Ці принципи забезпечують теоретичні положення та методологічні підходи до процесів створення, функціонування та розвитку інформаційних систем на всіх етапах їх життєвого циклу [7].

1.3. Аналіз сучасних інформаційних систем в освіті

Дуже великий вплив на формування особистості та розвиток в сучасному світі має середовище, в якому вона живе, здобуває освіту та працює. Тому актуальною проблемою для будь-якого вищого навчального закладу є необхідність використання сучасних інноваційних технологій для створення інформаційного освітньо-наукового середовища з комунікаційними можливостями. Це дасть змогу створити інформаційні системи, в яких студент

матиме змогу отримувати необхідну для освіти інформацію, консультації викладачів, спілкуватись з колегами по навчанню в зручний для себе час. Під час даного дослідження було розглянуто декілька з існуючих таких систем, першою була інформаційна система під назвою «Курсор».

Система «Курсор» тримає в собі інформацію про студентський та професорсько-викладацький склад університету, дані про виконання календарних планів, результати модульних контролю, розклад занять, навчальні плани, журнали відвідування. Система «Курсор» університету адаптована до умов модульної системи контролю та оцінювання знань студентів [19].

Система забезпечує оперативний контроль навчального процесу і повну інформацію по кожному студенту починаючи з особистих даних, фотографії, місця проживання, результатів здачі всіх контрольних термінів, заліків та іспитів за весь період навчання і закінчуючи напрямком і місцем роботи.

Система адаптована до нових умов кредитно-модульної системи контролю та оцінювання знань студентів.

Для кожного працівника є інформація про основне місце роботи, про сумісництво, про ступені та звання, про викладацький графік роботи, про адресу проживання та телефон. Також є телефонний довідник університету. Основними джерелами даних нової системи є діючі системи контролю самостійної роботи студентів старше тридцяти років, «КУРС» та «ДЕКАН», бази даних систем «Абітурієнт» та «СДР», бази даних «Кадрова форма», «Мед. Страхування», «Зарплата», «Гуртожитки», «Житло». Цими базами займаються в основному співробітники бухгалтерії, обчислювального центру (ОЦ) університету та інших підрозділів [13].

Додатково нова система завантажена базами даних «Розклад занять» (кафедра комп'ютерних інформаційних технологій), «Кадри», «Телефонний довідник», «Професійне тестування». Для беззатримного оновлення даних налаштовано передачу даних зазначеними підрозділами по ОЦ.

Технічно нова система базується на університетському сервері і доступна кожному, хто має до нього доступ. Можлива автономна установка системи, оновлення даних для автономних систем здійснюється через флешки (оновлюються тільки поточні дані). Оновлення даних на сервері виконується оперативною групою ОЦ щодня. Система має можливість надавати зворотній зв'язок, користувачі можуть надсилати дані для коригування бази даних [20].

Система має три рівні доступу до даних: повний доступ (адміністративний по іншому), деканський (обмежений доступ) та частковий доступ. Наразі до інтегрованої системи підключено понад двісті користувачів, у середньому щомісяця здійснюється понад тисячі доступів до бази даних.

Програма має багатовіконний інтерфейс, кожна екранна форма розташована в окремому незалежному вікні. Кожен новий екземпляр форми буде розміщуватися у тому же вікні, якщо воно було створене. Більшість форм мають інтерфейс сторінок із закладками, сторінки відкриваються мишею.

Форми: «Список учнів» та «Викладачі та співробітники» можна розгорнути на весь екран, розмір інших форм не змінюється. Згортайте та закривайте форми за допомогою значка «Згорнути» або «Закрити» в рядку заголовка вікна форми.

Список студентів є основною формою для перегляду та пошуку даних студентів. При відкритті форми відображається список усіх студентів, які зараз навчаються, кожен список супроводжується сортуванням за групами. Для пошуку студентів організовується вікно зі списком факультетів, вибираючи потрібний факультет, отримуємо на екрані список студентів. Так само студенти відбираються за фільтруючими курсами та студентськими групами. Також можна вказати умову вибору відповідних категорій студентів, наприклад військових, також можна вибрати тип фінансування (бюджет або контракт).

У нижній частині форми розташоване вікно пошуку за прізвищем. Досить набрати декілька перших літер прізвища студента, і курсор встановлюється на рядок, що починається з цих букв. У цьому ж вікні можна

набрати номер навчального плану студента (номер залікової книжки) і натиснути кнопку (пошук).

Список студентів можна сортувати за групами (за замовчуванням), за кількістю боргів за поточний семестр, за кількістю пропущених занять, за рейтингом, за алфавітом. Для сортування необхідно або вибрати заголовок відповідного стовпця за допомогою миші, або відкрити контекстне меню, клацнувши правою кнопкою миші на списку або за допомогою лівої кнопки миші на кнопці «Сортувати». При сортуванні за кількістю відпрацьованих годин графа «Кількість заборгованості» замінюється графою «Кількість відпрацьованих годин».

За допомогою миші виберіть кнопку «Група», отримаєте форму зі списком учнів групи, номер групи визначається вибраним рядком списку студентів, як показано на рисунку 1.3.

The screenshot shows a web application window titled 'Студенти ДНУЗТ'. At the top, there are dropdown menus for 'факультет', 'курс', 'група', and 'бюджет чи контракт'. Below these is a search bar with a 'Розширений пошук' button and a 'фото' checkbox. A status bar indicates 'Всього 3728 студентів'. The main area is a table with the following columns: 'Студент', 'Група', 'Ном.уч.пл.', 'Кільк.боргів', and 'Рейтинг'. The first row is highlighted in yellow and shows 'Бондаренко Ольга Володимирівна' in group '1011-A' with ID '130296' and a rating of '0.00'. Other rows list students like 'Віджаясекара Наяна Даммі', 'Данилюк Аліна Вадимівна', etc. At the bottom, there are buttons for 'пошук по прізвищу', 'пошук', 'Студент', 'Група', 'Сортувати', and checkboxes for 'показувати закінчивших навчання і відрахованих' and 'показати оцінки останньої сесії'.

Студент	Група	Ном.уч.пл.	Кільк.боргів	Рейтинг
Бондаренко Ольга Володимирівна	1011-A	130296		0.00
Віджаясекара Наяна Даммі	1011-A	130297		0.00
Данилюк Аліна Вадимівна	1011-A	130299		0.00
Єфремова Марія Віталіївна	1011-A	130300		0.00
Іващенко Ганна Сергіївна	1011-A	130304		0.00
Крюков Станіслав Сергійович	1011-A	130305		0.00
Кузнецова Олеся Олексіївна	1011-A	130306		0.00
Манько Анастасія Миколаївна	1011-A	130308		0.00
Паламарчук Ярослав Юрійович	1011-A	130314		0.00
Сокол Інна Володимирівна	1011-A	130320		0.00
Гурський Єгор Сергійович	1011-Б	130298		0.00
Завальнюк Дар'я Ігорівна	1011-Б	130302		0.00
Золотаревська Марина Юріївна	1011-Б	130303		0.00
Кущенко Анастасія Денисівна	1011-Б	130307		0.00
Молодан Діана Володимирівна	1011-Б	130310		0.00
Остролицька Ольга Андріївна	1011-Б	130313		0.00
Репп Ліна Вікторівна	1011-Б	130317		0.00

Рисунок 1.3 – Форма списку студентів

З наступної форми, можна обрати за допомогою миші потрібні кнопки та відкрити форми календарного плану групи, результати модульних робіт, як показано на рисунку 1.4.

№п/п	Ім'я	Контракт	Кільк. боргів	Годин пропусків	Ритм	Оцінки	Рейтинг
1	Байдужий Олександр Ігорович	контракт	0	0.00	3 3 3 3 Н	63.21	
2	Баталов Вадим Сергійович	контракт	6	0.00	3 Н Н Н 3	60.90	
3	Василевська Марина Олегівна		0	0.00	4 5 4 5 5	90.99	
4	Вовк Олексій Юрійович		2	0.00	3 4 3 3 3	73.05	
5	Запорожченко Денис Васильович	контракт	0	0.00	4 3 3 3 3	64.46	
6	Мельников Данііл Валерійович		0	0.00	3 3 3 3 4	64.48	
7	Мотиленко Володимир Артемович	контракт	0	0.00	3 3 3 Н 3	62.59	
8	Новіков Микита Максимович	контракт	4	0.00	3 3 3 Н 3	61.97	
9	Пашковський Станіслав Юрійович	контракт	0	0.00	3 3 3 4 3	66.71	
10	Петрук Тетяна Олексіївна		10	0.00	4 4 4 5 4	88.17	
11	Ребров Владислав Миколайович	контракт	2	0.00	3 3 3 3 4	64.03	
12	Свистун Юрій Дмитрович	контракт	0	0.00	4 3 3 4 3	71.40	
13	Семенець Артем Сергійович		8	0.00	4 4 4 3 4	76.15	
14	Солонина Денис Євгенович		0	0.00	4 3 3 3 3	66.59	
15	Тиква Антон Андрійович		0	0.00	4 3 4 Н 3	67.90	
16	Ткачик Олексій Сергійович	контракт	2	0.00	3 Н Н Н 3	61.10	
17	Ференець Артем Едуардович		0	0.00	3 3 4 3 3	64.66	
18	Циганок Олег Ігорович		2	35.33	4 3 3 3 4	66.49	
19	Шаповалов Гліб Євгенійович		4	0.00	3 3 3 Н 3	61.03	

Рисунок 1.4 – Форма інформації даних по групі

Портал «ІСУО. Україна» (інформаційна система управління освітою).

Система розрахована на всі навчальні заклади та дозволяє працівникам місцевих органів управління освітою автоматично формувати статистичні звіти, починаючи з 2016 року звіти в системі подаються лише з накладенням ЕЦП, що відповідає всім нормативним актам. Для потреб органу управління освітою та закладів освіти розроблено додатковий функціонал «Конструктор форм», який дозволяє створювати будь-які внутрішні звіти (кількість комп'ютерного ПЗ, паспорт закладу освіти тощо) та автоматично отримувати інформацію від закладів. в електронному вигляді.

Портал «ІСУО. Україна» має чітку структуру, побудовану на принципі підпорядкування, прийнятому в державі:

- Міністерство освіти і науки України;
- обласні управління / відділи освіти;
- районні та міські відділи / відділи освіти;
- загальноосвітніх навчальних закладів.

Відповідно до підпорядкування відбувається розподіл прав доступу користувачів до функціональності та інформації. Так, незареєстровані користувачі можуть бачити лише реєстр закладів освіти із загальною відкритою інформацією, директори шкіл та уповноважені спеціалісти мають можливість працювати з детальною інформацією лише про свій заклад, спеціалісти районного відділу освіти мають можливість працювати з інформацією про всі заклади свого району. регіонального рівня – у межах свого регіону, а спеціалісти міністерства – з інформацією по всій країні.

Портал «ІСУО. Україна» забезпечує захист інформації відповідно до чинних вимог і норм КСЗІ, що є сертифікатом Державної служби спеціального зв'язку та захисту інформації України.

На порталі можливий пошук у межах області чи району/міста за учнями, викладачами (без обробки персональних даних), закладами за різними критеріями (клас, вік, мова навчання, стать, посада, кваліфікація тощо). Наприклад, знайдіть усіх учнів 1-го класу на місцевості або вчителів І категорії. Масштабованість системи дозволяє включати й інші дані, зокрема про фінансування ЗЗСО та потребу в навчальній літературі.

ІСУО має масштабовану архітектуру, яка не залежить від архітектури обладнання та типу операційної системи. Це дозволяє збільшити кількість користувачів і обсяг даних, не втрачаючи працездатності під час важких робочих навантажень.

Портал «Україна. Інформаційна система управління освітою» включає набір інструментів та інструментів, призначених для автоматизації управління органами управління освітою всіх рівнів, загальноосвітніми, дошкільними закладами. Система дозволяє створити єдину систему збору, обробки та зберігання інформації в умовах загальної середньої та дошкільної освіти. При цьому проблема забезпечення органів управління освітою повною оперативною та достовірною інформацією про діяльність навчальних закладів, що дозволяє здійснювати управління освітою на вищому рівні, ефективніше планувати бюджети на всіх рівнях при виконанні освітніх програм,

раціонально здійснювати управління освітою. та розумні корективи для подальшої освіти окремих регіонів та держави в цілому, а з іншого боку – забезпечення належного рівня контролю за діяльністю навчальних закладів.

Зарахування студентів є одним із найбільш трудомістких процесів, якими займаються навчальні заклади. Зазвичай він містить численні завдання на паперовій основі, які, якщо виконуватись вручну, все частіше схильні до людських помилок. Затвердження бланків заяв студентів, підтвердження документів та вручну підтвердження відповідності студентів на даний курс є втомлюючим. На даному етапі розвитку з цим питанням добре справляється ЄДЕБО (Єдина Державна електронна База питань Освіти). Ця система вирішує багато питань в освіті, але стосується більше до глобального управління та регулювання питань в освітній сфері. Інформаційна система Тритон використовується для внутрішнього документообігу та дозволяє студентам вчитись дистанційно, маючи в собі індивідуальний план занять для всіх груп, навчальні посібники та лекції, практичні, лабораторні роботи т.д. Наша задача полягає у застосуванні системи на вузких кругах, а конкретніше – автоматизація діяльності роботи кафедри в університетах [21].

Вона дозволить безпроблемно вести облік студентів та слідкувати за їх успішністю в електронному варіанті, що дозволить позбавитись від великої кількості макулатури та постійні клопоти з ними. Крім користі педагогам, студенти також зможуть безперешкодно перегляди свої поточні оцінки, не виходячи з дому.

Складові частини системи: електронні журнали та щоденники, електронна реєстрація в заклади дошкільної освіти, електронна атестація педагогічних працівників, електронна реєстрація в заклади загальної середньої освіти, електронний деканат.

1.4. Автоматизація в освітній сфері

Сучасні технології активно змінюють спосіб автоматизації освіти. Фактична роль автоматизації в освіті та в повсякденному житті студентів залишає бажати кращого. В оточенні гаджетів студенти переважно використовують їх для спілкування та розваг. Водночас сама система освіти залишається вперто стійкою до трансформацій. Для більшості університетів підхід до навчання все ще є єдиним.

Автоматизоване навчання може дати ряд переваг, але також викликає ряд проблем. Для належного впровадження цих переваг та компромісів необхідно зважити.

Якщо замислитись, немає нічого незвичайного в тому, що нові технології повільно, але наполегливо проникають у наше повсякденне життя. Якщо раніше тенденція для багатьох винаходів, що змінюють світ, виглядала трішки смішною, то в сьогоденніх реаліях є достатньо умістною [4].

Говорячи про освіту, є електронні калькулятори, комп'ютери, діапроектори, «розумні» дошки та інші дивовижні винаходи, яких не було ще пару десятиліть тому. Для їх прийняття часто потрібен час, при цьому попередні користувачі отримують перевагу над іншими.

Те саме можна очікувати і від автоматизації. Зростаюча потреба у впровадженні RPA та AI для того, щоб йти в ногу з темпами швидкозмінних робочих місць, може стати порушенням, необхідним для стимулювання змін у освіті. Рішення відмовитися від цих технологій більше не є варіантом, оскільки навчальні заклади, які не впроваджують RPA та AI (Штучний інтелект), швидше за все, за короткий час виявляться застарілими.

Академічні адміністрації часто виявляються заваленими повторюваними завданнями, що забирають багато часу. Керувати розкладом занять, відстежувати відвідування студентами, обробляти оцінки, приймати нових студентів – усі ці процеси настільки ж нудні, як і цінні.

У цьому випадку автоматизація – єдиний спосіб підтримувати роботу адміністративних процесів, що дозволяє людям, які відповідають за ці процеси, заощадити час.

Розгорнувши програмне забезпечення для автоматизації в діяльності університетів, більшість процедур ручного оформлення документів можна повністю усунути. У довгостроковій перспективі це означає, що академічний персонал може стати більш креативним. В результаті чого вони зможуть зосередити свої погляди та ідеї на тому, що дійсно потребує освітня сфера.

Автоматизація освіти заощаджує час та підвищує продуктивність праці. Це також дозволяє студентам краще взаємодіяти з викладачами, отримуючи більш швидкий і точний відгук про свої домашні завдання, проекти та іспити. Після того, як їх університет запровадить методи автоматизації, студентам більше не доведеться стикатися з помилками при оцінці чи тижнями чекати, щоб отримати відгук про свою роботу.

Відповідно, вищі навчальні засоби повинні використовувати методи автоматизації, щоб навчити своїх студентів, як краще працювати з машинами – роблячи те, чого машини не можуть. Це означає приділяти більше уваги розвитку цифрових навичок студентів, а також посилювати наголос на навчанні за допомогою досвіду.

Автоматизація – найкращий спосіб для навчальних закладів зняти це навантаження зі своїх співробітників, збираючи точно заповнені форми, документи, заяви та афідевіти від студентів віддалено. Готове рішення RPA від airSlate з легко налаштованими ботами, які діють автоматично відповідно до простої логіки «якщо-то» керують усім процесом. Це дає можливість викладачам дати час для перевірки документів, оцінити відповідність критеріям та вирішити, чи потрібно зараховувати студента чи ні [6].

Планування та підготовка до зустрічей мають вирішальне значення для організації навчальних процесів. Зазвичай організатор перевіряє наявність відвідувачів перед тим, як відповідно запланувати зустріч. Потім вони повідомляють кожного учасника окремо, передаючи інформацію про дату, час

та місце зустрічі. Кожного разу, коли відбувається зміна, організатор повинен повторити всю процедуру заново.

Однак за допомогою рішення RPA весь процес планування може бути повністю автоматизований, щоб звести до мінімуму участь людини. Все, що потрібно зробити організатору, – це налаштувати простий робочий процес, який перевірятиме розклад кожного відвідувача та сповіщає учасників.

Відстеження відвідуваності – це ще один рутинний, але обов'язковий процес, який більшість навчальних закладів мають виконувати щодня. Це стосується не тільки студентів. Не менш важливо вести точний облік факультетів та адміністрації школи, щоб забезпечити справедливу компенсацію для кожного члена команди. Приймавши відповідний інструмент автоматизації для щоденного відстеження відвідуваності, можна, наприклад, збирати інформацію для робочого журналу своїх співробітників. Такий підхід дозволяє проводити більш справедливу оцінку та дозволяє кожному працівнику отримати належну винагороду відповідно до присутності в рутинних процесах установи.

Окрім складання іспитів, більшість викладачів зобов'язані заповнювати звіти про успіхи кожного студента своїх груп. Зайве говорити, що над цим потрібно багато нудної роботи. Автоматизація тут може стати чудовою підмогою. Після того, як оцінки учнів будуть збережені в електронному вигляді, вчителі можуть використовувати спеціальне програмне забезпечення для автоматизації освіти для вилучення необхідних даних з електронної таблиці. Вони також можуть попередньо заповнити картки прогресу і навіть надіслати звіт про прогрес зацікавленим, наприклад батькам. З виконанням цього, вони звичайно ж позбавляють себе від виснажливої роботи та зберігають багато часу, який далі можна присвятити більш корисним та продуктивним задачам.

Найбільш значним стимулом для створення додаткової вартості, який автоматизація принесе до столу, є додатковий час взаємодії, який він дає студентам та тим, хто їх навчає. Автоматизація дозволить університетам

адаптуватися до потреб студентів та забезпечити більш індивідуалізоване викладання, дозволяючи учням різного рівня кваліфікації працювати разом в одній групі.

Завдяки технології, яка постійно відстежує прогрес учнів і моделі навчання, автоматизація зможе допомогти вчителям виявити та усунути недоліки в їхньому викладанні. Окрім оцінювання заповнених пробілів та оцінок із кількома варіантами, це допоможе вчителям швидко оцінити запитання та есе з короткими відповідями.

Хоча більшість із цих переваг ще не впроваджено, є одне, що ми точно знаємо: коли люди не вводять дані, освітні процеси можна спростити та модернізувати, вимагаючи набагато менше часу та зусиль для доопрацювання.

1.5. Постановка задачі дослідження

Таким чином, оскільки автоматизація роботи ВНЗ, як правило, відбувається одночасно за різними напрямками, то це необхідно враховувати під час створення автоматизованих систем організації освітніх процесів на кожній кафедрі.

Об'єктом дослідження є процеси автоматизації діяльності кафедри.

Метою магістерської роботи підвищення ефективності ведення обліку успішності студентів шляхом розробки та впровадження модуля «Електронний журнал» для системи автоматизації процесів діяльності кафедри «Електронна кафедра».

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих інформаційних систем та їх вплив на освітню сферу;
- дослідити можливі рішення та підходи для створення ефективної інформаційної системи;

– проектування, реалізація та впровадження модуля ведення обліку та успішності студентів для системи автоматизації процесів діяльності кафедри «Електронна кафедра».

2 ДОСЛІДЖЕННЯ МОЖЛИВИХ РІШЕНЬ ТА ПІДХОДІВ ДЛЯ СТВОРЕННЯ ЕФЕКТИВНОЇ ІС

2.1. Дослідження архітектури інформаційних систем

Інфраструктура інформаційних технологій включає в себе компоненти обладнання, програмні компоненти та послуги, які надає співробітник організації, інтегрованих разом.

Інфраструктуру інформаційних технологій можна умовно поділити на 5 елементів:

- апаратні компоненти;
- компоненти програмного забезпечення;
- люди;
- база даних та пристрої зберігання;
- мережі та комунікаційні компоненти.

Вона також має в собі відповідну операційну інтеграцію вищезгаданих компонентів з їхньою документацією, обслуговуванням та управлінням.

Архітектура інформаційної системи (Information System Architecture) відноситься до бізнес-процесів і політик, структури системи, технічної структури і технологій продукту, необхідних для бізнесу або інформаційної системи.

Архітектура складається з детального опису конструкції, змісту, списку поточного обладнання, програмного забезпечення та мережевих можливостей комп'ютеризованої системи. Ця архітектура включає як апаратне, так і програмне забезпечення, що використовується для надання рішення замовнику. Також містить в собі докладну інформацію про довгострокові плани, такі як оновлення або заміна старого обладнання та програмного забезпечення.

Архітектуру інформаційної системи будь-якої компанії можна розглядати як схему, що показує, як включено систему обробки даних,

телекомунікаційні мережі та дані. У ній дається узагальнена відповідь на такі питання:

- Де зберігати дані?
- Які дані збираються?
- Як і як збиралися дані?
- Яка програма використовує дані і як вони пов'язані з системою в цілому?
- Як передаються дані?

Компоненти інформаційних систем:

- апаратні ресурси. Апаратні ресурси відносяться до всіх фізичних пристроїв, що використовуються для обробки інформації. Сюди входять не лише комп'ютери та інше обладнання, а й будь-який матеріальний об'єкт, дані про який можуть зберігатися, від простого аркуша до магнітних або оптичних дисків.

Декілька видів апаратних ресурсів такі:

- комп'ютерна периферія. Ці пристрої використовуються в комп'ютері для введення даних, зберігання даних та виводу. Наприклад: клавіатура використовується для введення даних, принтер для виведення інформації та магнітні стрічки для зберігання інформації;

- комп'ютерні системи. Він складається з центрального процесора, що містить мікропроцесори та безліч взаємопов'язаних периферійних пристроїв. Наприклад: портативні пристрої, портативні або настільні мікрокомп'ютерні системи, мейнфрейми середнього та великого розміру;

- кадрові ресурси. Люди грають вирішальну роль ефективному функціонуванні всієї інформаційної системи.

Існують наступні типи людських ресурсів:

- фахівці з ІБ. Тих, хто розробляє та експлуатує інформаційну систему, називають спеціалістами з ІБ. Це можуть бути системні аналітики, розробники програмного забезпечення, системні оператори та інший управлінський, технічний або технічний персонал ІБ;

– кінцеві користувачі (користувачі або клієнти). Користувачі, які використовують інформаційну систему або інформацію, яку вона виробляє, відомі як кінцеві користувачі. Можна сказати майже всі люди, є користувачами інформаційних систем;

– ресурси даних. Це цінний організаційний ресурс, яким необхідно ефективно управляти, щоб користуватися всім кінцевим користувачам в організації. Поняття ресурсів даних було розширено менеджерами та фахівцями з інформаційних систем, щоб вкласти в нього набагато більше, ніж вихідний матеріал інформаційної системи.

Дані – це ресурси, які організовані, зберігаються та доступні за допомогою технологій управління ресурсами даних у таких формах:

– база знань, що містить знання у різних формах, таких як факти, правила та приклади успішної ділової практики;

– база даних, що містить оброблені та організовані дані. Наприклад: дані про транзакції продажів збираються, обробляються та зберігаються в базі даних продажів із підключенням до Інтернету, яка використовується менеджерами або фахівцями з маркетингу для звітів з аналізу продажів. База знань використовується системою управління знаннями та експертними системами для обміну знаннями з метою надання експертних рекомендацій у певній галузі;

– програмні ресурси. Збірник інструкцій відомий як програмне забезпечення. Інструкції, що використовуються для керування комп'ютерним обладнанням, відомі як програми, а інструкції, що використовуються для обробки даних, відомі як процедури.

Програмне забезпечення буває двох типів:

– прикладне програмне забезпечення. Це програмне забезпечення для безпосереднього керування комп'ютерами та підтримки їх виконання потрібно кінцевим користувачам. наприклад MS Word, MS Excel, система управління бібліотеками тощо;

– системне програмне забезпечення. Програмне забезпечення контролює та спрямовує операції комп'ютерної системи. Наприклад: операційна система.

Електронний бізнес та електронна комерція всіх типів організацій та їх інформаційних систем будуть успішними, якщо будуть впроваджені телекомунікаційні технології та мережі, такі як Інтернет, Інтранет та Екстранет. Телекомунікаційна мережа складається з комп'ютера, комунікаційних процесів та інших пристроїв, пов'язаних між собою комунікаційними середовищами та керованих ним програмним забезпеченням.

Основними компонентами ресурсів усієї інформаційної системи є комунікаційні технології та мережа. Основні мережеві ресурси такі:

- мережева інфраструктура. Це стосується обладнання, програмного забезпечення та технологій передачі даних, що використовуються для підтримки та управління організаціями мережі зв'язку. наприклад Модеми, міжмережеві процесори та програмне забезпечення для керування зв'язком;
- засоби масової інформації. Вони використовуються для доставки та отримання інформації або даних.

Класифікація архітектури інформаційних систем:

- клієнт-серверна архітектура. Модель клієнт-сервер для обчислення окремих завдань між постачальниками ресурсів або послуг, які називають серверами, і ініціаторами запитів на послуги, званими клієнтами. Архітектура клієнт/сервер – це загальна модель у будь-якій програмній організації, яка може бути реалізована декількома способами. Наприклад: взаємодія між клієнтською прикладною програмою, що виконується на робочій станції, та системою управління базами даних (СУБД), що виконується у більшій комп'ютерній системі, як показано на рисунку 2.1.

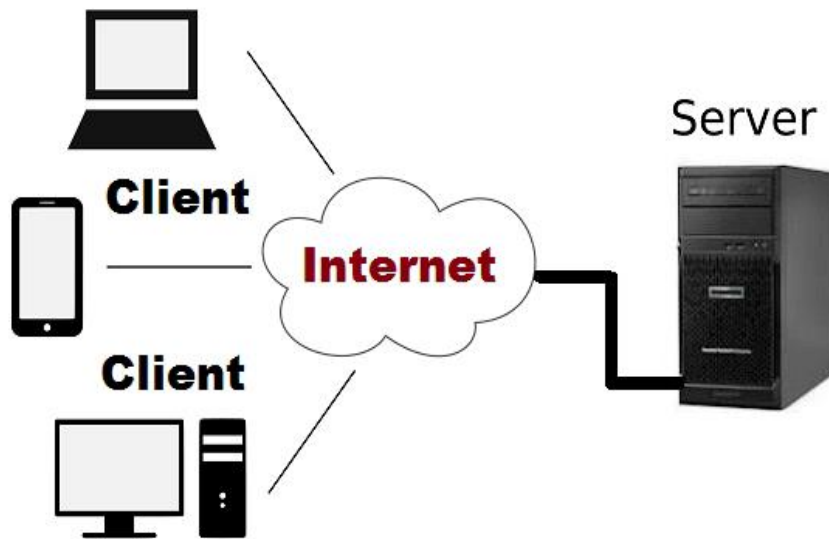


Рисунок 2.1 – Клієнт-серверна архітектура

Запит на доступ до бази даних надсилається прикладною програмою в систему управління базою даних через мережу. СУБД буде звертатися до даних від імені програми та надасть відповідь у вигляді результату операції пошуку для звіту про успіх / невдачу операції оновлення.

Архітектура програмного забезпечення клієнт/сервер є гнучкою модульною інфраструктурою на основі повідомлень. Він спрямований на підвищення зручності використання, взаємодії, гнучкості та масштабованості порівняно з централізованими обчисленнями з розподілом часу на мейнфреймах.

Компонентами цієї архітектури є клієнт, що запитує машина та сервер, ця машина-постачальник. Вони пов'язані один з одним або локальною мережею (LAN) або глобальною мережею (WAN).

Сервер виконує такі функції:

- обробка запиту;
- повернення результату клієнту;
- записує запит клієнта.

На клієнта покладається така відповідальність:

- декодування запиту користувача у певні протоколи, щоб дозволити обробку;
- зручний інтерфейс користувача;
- подання результату користувачеві;
- очікування відповіді від серверу;
- надсилання запиту на сервер;
- перетворення відповіді на доступний для читання формат.

Архітектура мейнфрейму: у мейнфреймі термінал приймає запити від користувачів, але не перевіряє їх. Це виконується сервером під час обробки запиту. Цей тип архітектури був поширений декілька десятиліть тому, зараз сценарій змінився. Така архітектура зараз не використовується через її негнучкість і погане співвідношення ціни та якості.

Як розширення цієї парадигми виникла концепція інтелектуального терміналу. Тут мейнфрейм об'єднаний з декількома ПК як інтелектуальні термінали, кожен з яких містить клавіатуру та екран, а також дисковий накопичувач, який виконує обмежені завдання обробки. Це зменшило залежність від постійної взаємодії із центральним комп'ютером.

Архітектура мейнфрейму включає:

- користувачі терміналу вводять запити, на яких відображаються результати;
- сервер, що обробляє дані у мережі.

Користувач надсилає запит даних через термінал. Сервер обробляє запит і виводить результат на екран терміналу.

Архітектури інформаційних систем можна обговорювати далі і далі, та внаслідок детального аналізу існуючих архітектур було прийнято рішення використовувати сучасну, розвинену та широко використовувану клієнт-серверну архітектуру, що дозволить дуже зручно розробляти та керувати компонентами системи за допомогою своєї гнучкості та простоти використання.

2.2. Аналіз технологій клієнтської частини

Для розробки клієнтської частини було проаналізовано сучасні веб-технології та технології зв'язку клієнта з сервером, було обрано такі технології: Vue.js, GraphQL, Node.js. Далі буде проаналізовано кожен з них.

При використанні Vue.js ми насамперед користуємося його складовими технологіями, такими як HTML, CSS та JS. Опишемо коротко кожен з них.

2.2.1 HTML

HTML означає «мова розмітки гіпертексту». Це мова кодування, яка використовується для створення сторінок, які можуть відображати веб-браузер. Більшість веб-сторінок, які ви знайдете в Інтернеті, включаючи ту, на якій ви читаєте цю статтю, зберігаються у вигляді HTML-файлу. Веб-сайти – це набір пов'язаних HTML-сторінок, що зберігаються на сервері. Ось чому цю мову часто називають «основою Інтернету». Щоразу, коли ви переходите на сторінку в Інтернеті, ви, по суті, вимагаєте HTML-файл, що зберігається на сервері. Потім браузер, наприклад Chrome або Firefox, проаналізує HTML-код і відобразить його так, як задумано. Як і будь-яка мова програмування, HTML виглядає як набір команд та текстових блоків, перш ніж він перетвориться на візуальну складову.

Якщо вам цікаво подивитися, як виглядає HTML-код на якійсь конкретній сторінці, і ви знаходитесь в інтернеті через комп'ютер чи ноутбук, спробуйте клацнути правою кнопкою миші в будь-якому місці цієї сторінки і вибрати «Переглянути вихідний код сторінки» (Inspect). Це має продемонструвати вам величезну стіну коду. Важливою частиною створення HTML сторінок є використання каскадних таблиць стилів (CSS). Це документи, які визначають, як мають виглядати певні елементи сторінки. Наприклад, наскільки великими мають бути зображення, які шрифти мають відображатися на сторінці і як веб-сторінка має реагувати на зміну розміру або розтягування. Все це має вирішальне значення для створення привабливих,

цілісних та стильних веб-сайтів. Якщо ви помітили, що останнє десятиліття веб-сайти стали краще виглядати, то зростання використання CSS є головною причиною [13].

2.2.2 CSS

CSS – це спеціальна мова, що використовується для відображення зовнішнього вигляду документа, написаного з використанням мови розмітки даних. Простіше кажучи, мова CSS призначена для того, щоб надавати потрібний зовнішній вигляд документам, написаними мовами розмітки.

Додавання зовнішнього вигляду документам типу HTML – це хоч і найпопулярніший, однак лише окремий випадок застосування мови CSS, тому що з його допомогою можна надавати вид і документам інших типів: XHTML, SVG і XUL.

Отже, метою створення CSS було відділити опис зовнішнього вигляду від самої розмітки. Як ви вже знаєте, для опису структури використовується HTML, для опису ж того, як ця структура буде виглядати, відповідає саме CSS. Роздільне керування структурою та зовнішнім виглядом надає гнучкості в керуванні документом та зменшити шанс написати однаковий код, який би точно виникав при використанні HTML для стилізування.

Використання CSS дає можливість задавати для сторінки та її елементів різні типи та розміри шрифту, кольору елементів, внутрішні та зовнішні відступи елементів один від одного, розташування окремих блоків чи стрічкових елементів на сторінці і т.д. Щоб HTML документ усвідомив про те, що за позиціонування та оформлення його елементів буде відповідати .css файл, треба це вказати. Для цього існують різні способи підключення CSS до документа, які дають браузеру зрозуміти, що на сторінку в цілому, або до якихось окремих її елементів повинно бути застосовано стильове оформлення.

Ці правила оформлення можуть знаходитися в самому html документі в потрібному тезі або винесені в окремий файл та підключено також за допомогою тегу вказавши ім'я цього файлу.

2.2.3 JavaScript

JavaScript – це мова програмування, яка широко використовується в браузерах для додавання інтерактивності веб-сторінок. Але крім цього, JS може добре впоратися з реалізацією будь-якого функціоналу на стороні сервера та взятий за основу у багатьох сучасних фронтенд фреймворках. Один із цих фреймворків був використаний саме при розробці нашого модуля, тож розглянемо його детальніше.

2.2.4 Vue.js

Vue.js – це бібліотека джаваскрипту що допомагає створювати веб-інтерфейсів з використанням паттерну MVVM (Model-View-ViewModel).

Оскільки Vue працює лише на представницькому рівні і не використовується для проміжного програмного забезпечення та бекенда, він може легко інтегруватися з іншими проектами та бібліотеками. Vue.js містить широку функціональність на рівні продуктивності і може використовуватися для створення потужних односторінкових веб-додатків.

Основні можливості Vue:

- реактивні інтерфейси;
- декларативний рендеринг;
- зв'язування даних;
- директиви;
- логіка шаблонів;
- компоненти;
- обробники подій;

- властивості;
- переходи та анімація CSS;
- фільтри.

Vue.JS створювався з огляду на найкращі практики таких технологій як React.JS, JQuery та Angular. З React.JS команда Vue запозичила ідею віртуального DOM. Такий підхід не дозволяє взаємодіяти напряму з вузлами інтерфейсу. Спочатку Vue працює зі своєю віртуальною копією DOM. Після того, як зміни відбудуться у копії, все застосується на реальних вузлах. Віртуальне дерево постійно порівнюється паралельно з реальним. Таким чином знаходиться різниця та інтерфейс перемальовується тільки там, де відбулися якісь зміни.

Переваги Vue.js:

- бібліотека досить проста та функціональна. Щоб розібратися в ній, потрібен мінімальний багаж знань;
- вимоги до стеку відсутні, тому Vue.JS можна використовувати на будь-якому проєкті;
- фреймворк зовсім небагато важить. Ми заощаджуємо час завантаження сторінок та отримуємо цілу купу плюшок: плюс до конверсії, плюс до UX, плюс пошукове ранжування тощо;
- досить висока швидкість розробки. Завдяки використанню будь-яких шаблонів і доступності документації, більшість проблем вирішуються досить швидко. У тому числі в порівнянні з React, так як у більшості додатків, які не мають складних інтерфейсів, вся міць цього фреймворку є трохи надмірною;
- можливість знайти та підключити до проєкту практично будь-якого розробника, хто хоч трохи знайомий з фронтенд розробкою. Низький поріг входження дозволяє працювати з фреймворком як фронт розробникам, так і бек.

Vue дозволяє створювати функціональні програми, що відповідають усім сучасним стандартам, при мінімальному підключенні нових ресурсів і, власне, дешевше.

Недоліки Vue.js:

- робота над станом програми відбувається «під капотом». У масштабних проєктах це може вилитися в неочевидність роботи компонентів, що часто створює проблеми з налагодженням та ефективною розробкою;
- компонентний підхід у Vue.JS не такий гнучкий і очевидний, як у React;
- система рендерингу React є більш функціональною. Вона надає більше можливостей для налагодження;
- гірша шаблонізація.

Vue.JS реалізує всі сучасні підходи до розробки WEB UI та є легким у освоєнні, гнучким та високо інтегрованим із сторонніми технологіями фреймворком. Безперечно, у плані створення складних інтерфейсів React його перевершує. Але для вирішення щодо простих завдань використання Vue виглядає не тільки виправданим, а й кращим.

2.3. GraphQL

Перш ніж розглядати GraphQL, треба приділити увагу історичній основі та знайти відповіді на питання що ж таке sql взагалі.

SQL – декларативна мова програмування, яка застосовується для створення, зміни та керування даними в базах даних. Ця мова підтримує чотири базові оператори запитів: SELECT, INSERT, UPDATE та DELETE. За допомогою SQL ми можемо запросити з бази даних (БД) те, що нам необхідно.

Наприклад, коли необхідно «дістати» з БД всіх користувачів з ім'ям Maria, це можна зробити за допомогою запиту, який представлено на лістингу 2.1.

Лістинг 2.1 Реалізація перевірки запиту на ім'я користувача

```
SELECT * FROM users WHERE first_name = «Lena»
```

Вирішити це завдання за допомогою REST можна кількома способами:

1. Визначити endpoint на сервері, який віддаватиме з бази даних користувачів з first_name Lena.
2. Визначити загальний endpoint для отримання всіх користувачів та фільтрувати отриманий список на стороні клієнта.

Лістинг 2.2 Реалізація перевірки запиту на ім'я користувача за допомогою REST

```
users.filter(user => user.first_name === «Lena»);
```

При цьому в кожному варіанті є свої мінуси. Перший підхід не можна масштабувати, тому що немає можливості створити endpoint для кожного користувача. Якщо ми використовуємо другий підхід, підвищується навантаження на мережу, а також виникає потреба у постобробці на стороні клієнта.

А тепер уявіть інструмент, який поєднує можливості SQL та REST на стороні клієнта. Ви вже здогадалися, що він називається GraphQL. Цей інструмент бере ідеї, розроблені для маніпуляції даними в БД, і використовує в Інтернеті. Тому за допомогою одного запиту GraphQL можна отримати одразу всі необхідні дані.

Query: запити у GraphQL

За допомогою запитів GraphQL отримує необхідні дані із сервера. Тип запиту Query GraphQL – аналог GET в REST. Запити – рядки, які надсилаються в HTTP POST-запиті. Query описує дані, які потрібно отримати з сервера. У відповідь на запит сервер надсилає дані у форматі JSON. Структура відповіді відповідає структурі запиту [16].

Успішні операції повертають JSON з ключем «data» та з ключом «error», а неуспішні повертають JSON з ключем та повідомленням про помилку. Завдяки цьому зручно обробляти помилки на стороні клієнта.

Mutation: мутації у GraphQL

Mutation – ще один root types. З його допомогою можна додавати дані до БД. Mutation – аналог POST і PUT в REST.

Subscription: передплати GraphQL

Subscription – третій тип операцій у GraphQL. З його допомогою клієнт слухає зміни БД в режимі реального часу. Під капотом передплати використовують вебсокети. За допомогою таких запитів можна отримувати списки з даними щоразу, коли вони змінюються.

Робота із сервером GraphQL завжди починається з розробки схеми (Schema). Вона складається з двох взаємопов'язаних об'єктів: TypeDefs та Resolvers.

Вище описані основні типи GraphQL. Щоб сервер міг із нею працювати, ці типи необхідно визначити. Об'єкт typeDef визначає список типів, доступних у проекті.

Після визначення типів необхідно додати їхню логіку. Це необхідно, щоб сервер знав, як відповідати на запити клієнта. Це завдання вирішується за допомогою Resolvers.

Resolver або розпізнавач – це функція, яка повертає дані для певного поля. Резолвери повертають дані того типу, що визначено у схемі. Розпізнавачі можуть бути асинхронними. З їх допомогою можна отримувати дані з API REST, бази даних або іншого джерела.

2.4. Node.js

Термін «сервер» у комп'ютерній термінології може означати як окремий комп'ютер, так і програму. Основною особливістю в обох випадках є здатність машини або програми більшу частину часу працювати автономно, без втручання людини у відповідь на зовнішні події відповідно до встановленого програмного забезпечення. Сервер (як комп'ютер) – це комп'ютер у локальній або глобальній мережі, що забезпечує функціонування мережі, а також усі або частину її функцій. Сервер (як програма) – програма, яка надає деякі послуги

іншим програмам або клієнтам. Зв'язок між клієнтом і сервером зазвичай здійснюється через обмін повідомленнями і використовує відповідний протокол для кодування запитів клієнта та відповідей сервера [7].

Як середовище асинхронних подій JavaScript, Node.js розроблено для створення масштабованих мережевих додатків. Функція зворотного виклику викликається для кожного з'єднання, але коли немає з'єднань, Node.js засинає.

Це контрастує з більш загальною моделлю, яка використовує паралельні потоки ОС. Цей підхід відносно неефективний і дуже складний у використанні. Крім того, користувачам Node.js, можливо, не доведеться турбуватися про блокування процесів, оскільки немає обмежень. Майже жодна з функцій в Node.js не працює безпосередньо з вводом-виводом, тому процес ніколи не блокується. Оскільки на Node.js нічого не блокується, розробити масштабовані системи легко.

Завдяки підвищеній доступності, мова JavaScript на даний час активно розвивається, а сучасні бізнес-процеси, зазнали значних змін в порівнянні з минулим. Ті дії, котрі ми на сьогоднішній день можемо робити в Інтернеті за допомогою JavaScript, який функціонує як на сервері, так і в браузері, було складно собі уявити ще кілька років тому. Node.js – це серверна платформа, розроблена для створення масштабованих розподілених серверних додатків, які використовують архітектуру, орієнтовану на події, і неблокуючу асинхронну взаємодію. Основна ідея Node.js полягає у використанні неблокуючих подій вводу-виводу, щоб бути простим і ефективним під час роботи з додатками, які обробляють великі обсяги даних в режимі реального часу і запускаються на розподілених пристроях. Все це означає, що Node.js не є платформою для всіх випадків, яка буде домінувати у світі веб-розробки. Навпаки, він призначений лише для вирішення певних типів проблем. Якщо у вашому додатку використовуються важкі обчислення, тоді Вам слід відмовитися від використання Node.js, оскільки ці операції анулюють всі переваги цієї платформи. Та якщо Ви створюєте систему, яка підтримує взаємодію з багатьма одночасними підключеннями, причому із

двунаправленою взаємодією, то рано чи пізно Вам знадобиться організація асинхронного доступу. Так, саме доступу – під асинхронністю розуміють асинхронний доступ до ресурсів – файлів, сокетів, даних. Тому Node.js добре підходить для вирішення нашої задачі на серверній стороні, так як модуль має постійні та у великій кількості запити на отримання та оновлення даних.

3 ПРОЕКТУВАННЯ, РЕАЛІЗАЦІЯ ТА ВПРОВАДЖЕННЯ МОДУЛЯ ВЕДЕННЯ ОБЛІКУ ТА УСПІШНОСТІ СТУДЕНТІВ В ІНФОРМАЦІЙНІЙ СИСТЕМІ «ЕЛЕКТРОННА КАФЕДРА»

3.1 Вимоги до системи

В рамках кваліфікаційної роботи головною метою було створення модуля обліку та успішності студентів на кафедрі. Розроблений модуль повинен функціонувати в загальному ритмі з основною системою. Розроблена система призначена для автоматизації процесів кафедри, покращення та спрощення керування навчальним процесом в університеті та повний перехід від журналів в документному форматі до електронних. Опишемо вимоги до системи. Ознайомимось з основними функціями модуля за допомогою юзкейсів нижче. В таблицях 3.1 – 3.8 опишемо вимоги до модуля.

Дія діючої особи, якою є викладач, полягає в постановці оцінки студенту з певного предмету. Передумовою є відсутність оцінки у студента перед цією подією. Успішний сценарій включає в себе вибір викладачем групи та предмету, обрання дати проведення оцінювання, постановку оцінки, а також занесення цієї оцінки в базу даних. Результатом виконання цих кроків є те, що студент отримує відповідну оцінку за вказаним предметом, як показано у таблиці 3.1.

Таблиця 3.1 – Виставлення оцінки студенту

Діюча особа	Викладач
Ціль	Поставити оцінку студенту
Передумова	У студента ще немає оцінки
Успішний сценарій	1. Викладач вибирає групу 2. Викладач вибирає предмет 3. Викладач обирає дату

Продовження таблиці 3.1

	4. Викладач ставить оцінку 5. Оцінка заноситься в базу
Результат	Студент отримав оцінку

В даному контексті викладач має за мету додати нову колонку для оцінок до таблиці. Зазначено, що передумовою для цього є відсутність колонки з вказаною датою. У успішному сценарії викладач вибирає групу та предмет, додає нову дату та забезпечує внесення цієї дати в базу даних. Результатом виконання цих кроків є успішне створення колонки для додавання нових оцінок, як показано в таблиці 3.2.

Таблиця 3.2 – Додавання нової колонки до таблиці

Діюча особа	Викладач
Ціль	Додати нову колонку для оцінок до таблиці
Передумова	Колонка з такою датою не існує
Успішний сценарій	1. Викладач вибирає групу 2. Викладач вибирає предмет 3. Викладач додає нову дату 4. Дата заноситься в базу
Результат	Створена колонка для додавання нових оцінок

В рамках цього завдання викладач спрямовує зусилля на зміну оцінки у студента. Передумовою для цього є наявність вже існуючої оцінки у студента. У вдалому сценарії викладач вибирає групу та предмет, визначає оцінку, після чого здійснює необхідні зміни. Змінена оцінка вноситься до бази даних, і в результаті студент отримує оновлену оцінку, як показано в таблиці 3.3.

Таблиця 3.3 – Редагування оцінки студенту

Діюча особа	Викладач
Ціль	Змінити оцінку студенту
Передумова	У студента вже є оцінка
Успішний сценарій	1. Викладач вибирає групу 2. Викладач вибирає предмет 3. Викладач оцінку 4. Викладач змінює оцінку 5. Оцінка змінюється в базі
Результат	У студента оновила оцінка

У цьому завданні викладач має за мету змінити тип оцінок для конкретної групи студентів. Передумовою є те, що тип оцінок ще не заданий для цієї групи. У вдалому сценарії викладач вибирає групу та предмет, обирає необхідну дату, змінює тип оцінки, і ця зміна відображається в базі даних. В результаті виконання цих кроків ефективно змінюється тип оцінок для вказаної групи, як показано в таблиці 3.4.

Таблиця 3.4 – Зміна типу оцінки у групі

Діюча особа	Викладач
Ціль	Змінити тип оцінок групі
Передумова	Тип не заданий
Успішний сценарій	1. Викладач вибирає групу 2. Викладач вибирає предмет 3. Викладач обирає дату 4. Викладач змінює тип оцінки 5. Тип заноситься в базу
Результат	Змінений тип оцінок

У даному завданні викладач має за мету завантажити список студентів для конкретної групи. Передумовою є обрана група, для якої потрібно

отримати список студентів. У успішному сценарії викладач вибирає групу, завантажує відповідний файл, і інформація про студентів ефективно вноситься до бази даних. В результаті виконання цих кроків отримується завантажений список студентів, доступний у базі даних, як показано в таблиці 3.5.

Таблиця 3.5 – Завантаження списку студентів

Діюча особа	Викладач
Ціль	Завантажити список студентів
Передумова	Обрана група
Успішний сценарій	1. Викладач вибирає групу 2. Викладач завантажує файл 3. Студенти заносяться в базу
Результат	Завантажений список студентів в БД

В даному завданні викладач має за ціллю завантажити дані з таблиці до програми Excel. Передумовою є наявність даних у вибраній таблиці. У успішному сценарії викладач вибирає групу та предмет, після чого ефективно вивантажує дані. Як результат цих кроків, створюється Excel файл, що містить витягнуті дані з обраної таблиці, як показано в таблиці 3.6.

Таблиця 3.6 – Завантажити дані з таблиці

Діюча особа	Викладач
Ціль	Завантажити дані з таблиці в Excel
Передумова	Таблиця не повинна бути пустою
Успішний сценарій	1. Викладач вибирає групу 2. Викладач вибирає предмет

Продовження таблиці 3.6

	Викладач завантажує дані
Результат	Excel файл з даними

В рамках цього завдання викладач має на меті додати нову групу. Успішний сценарій передбачає три основних кроки: викладач натискає на кнопку, вводить назву нової групи, і ця інформація ефективно додається до бази даних. В результаті виконання цих кроків створюється нова група, яка тепер доступна для подальшого використання, як показано в таблиці 3.7.

Таблиця 3.7 – Додавання нової групи

Діюча особа	Викладач
Ціль	Додати нову групу
Передумова	-
Успішний сценарій	1. Викладач натискає на кнопку 2. Викладач вводить назву групи 3. Група додається до бази
Результат	Створена нова група

В даному завданні викладач має за мету додати новий предмет. У успішному сценарії викладач натискає на відповідну кнопку, вводить назву нового предмета, і інформація успішно додається до бази даних. Як результат виконання цих кроків, в системі створюється новий предмет, який тепер може бути використаний у навчальному процесі, як показано в таблиці 3.8.

Таблиця 3.8 – Додавання нового предмету

Діюча особа	Викладач
Ціль	Додати новий предмет

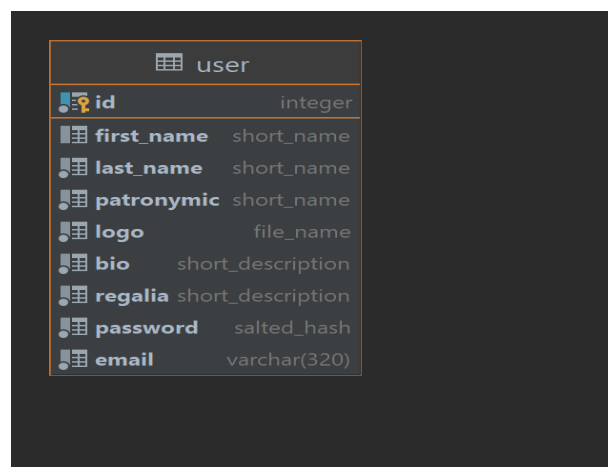
Продовження таблиці 3.8

Ціль	Додати новий предмет
Передумова	-
Успішний сценарій	1. Викладач натискає на кнопку 2. Викладач вводить назву предмета 3. Предмет додається до бази
Результат	Додано новий предмет

3.2. Розробка програмного модуля

Безпосередня розробка продукту була реалізована за допомогою технології Vue.js та GraphQL. Для роботи програми була задіяна база даних, яка виступає фундаментом та головним критерієм у модулі, тому почнемо з опису таблиць, які напряду пов'язані та задіяні у основних функціональних можливостей програми.

Нижче на рисунку 3.1. наведена схема таблиці користувачів системи.



user	
id	integer
first_name	short_name
last_name	short_name
patronymic	short_name
logo	file_name
bio	short_description
regalia	short_description
password	salted_hash
email	varchar(320)

Рисунок 3.1 – Схема таблиці користувачів в базі

В даній таблиці знаходяться повністю всі користувачі системи. Далі приведемо схему таблиці студентів, їх привязку до групи та освітньої програми, як показано на рисунку 3.2.

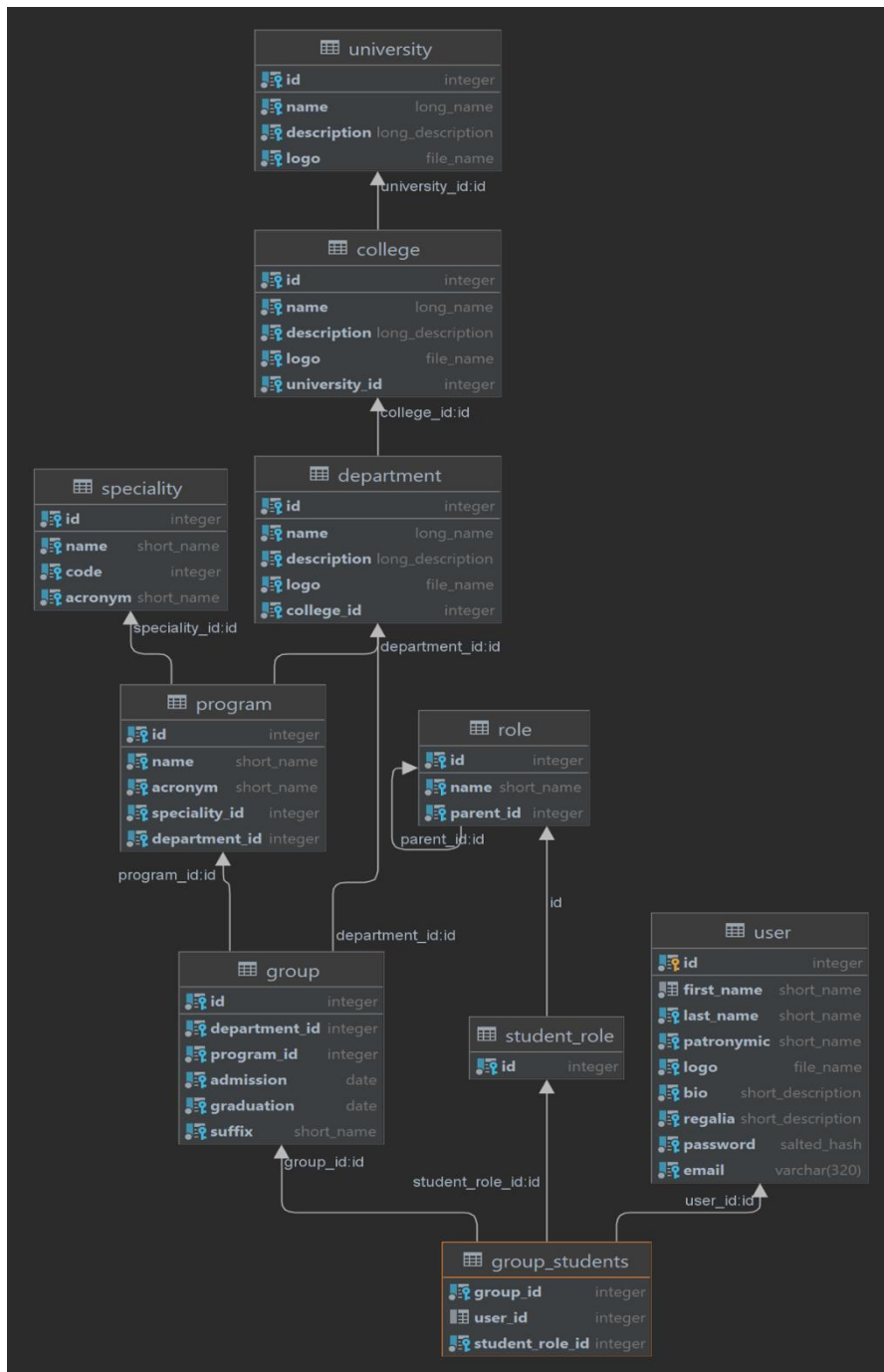


Рисунок 3.2 – Схема зв'язку між студентами та університетом

На схемі показано повна логіка зв'язку студентів між ними та їхніми групами, спеціальностями, навчальними програмами та університету.

Для ведення обліку та успішності було розроблено ще одну таблицю окремо для оцінок, як показано на рисунку 3.3.

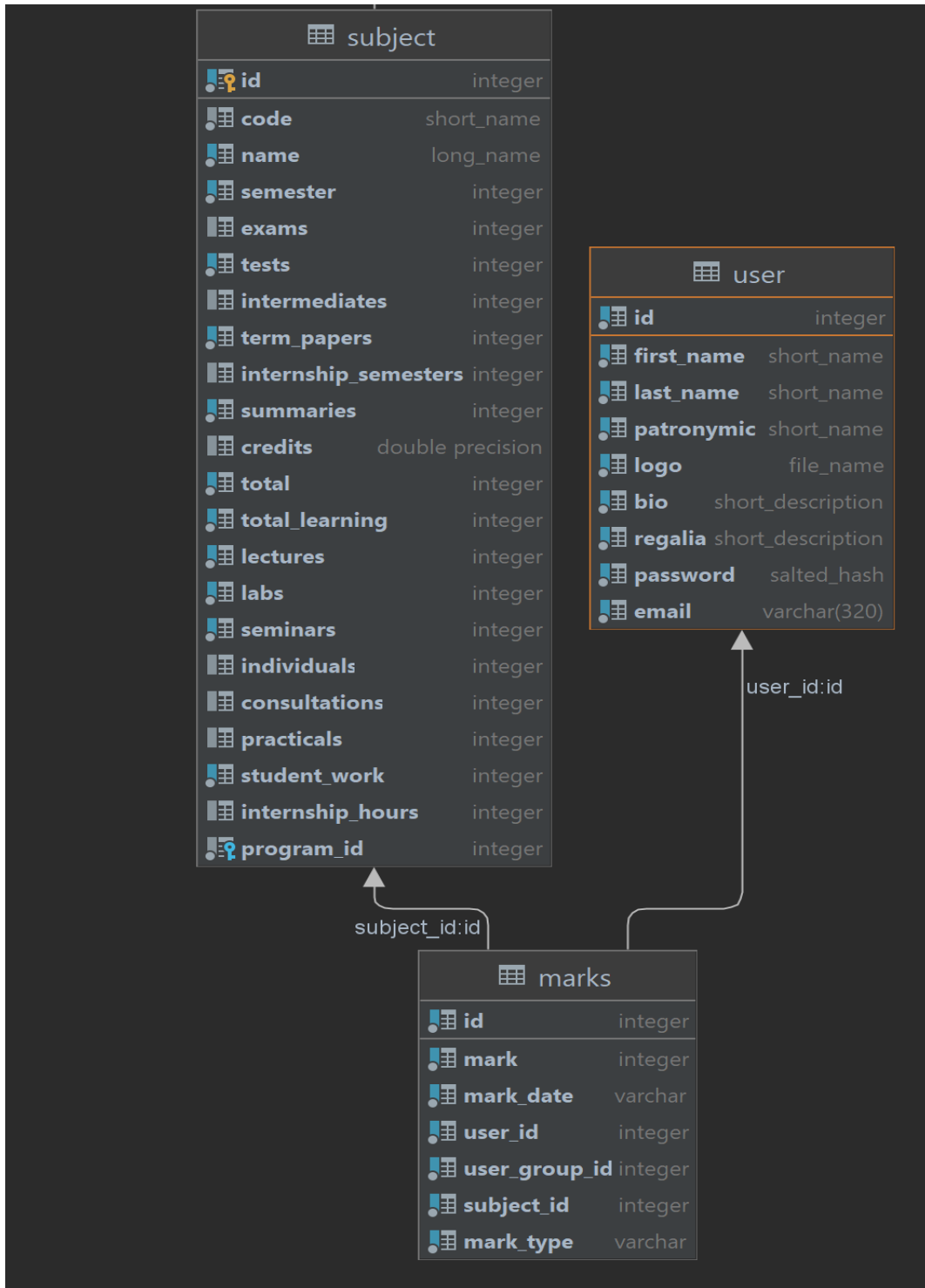


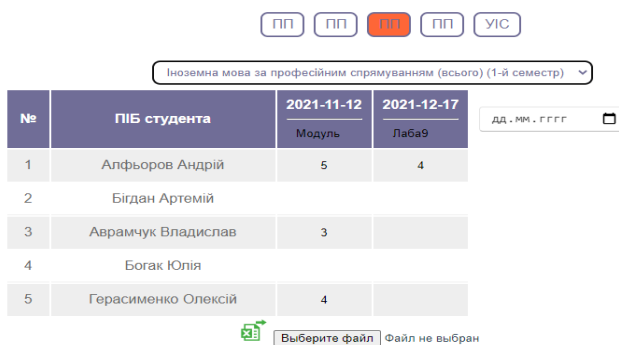
Рисунок 3.3 – Схема таблиці оцінок

Функціональні можливості системи повністю зав'язані на логіці роботи з базою, при кожній взаємодії з будь-яким елементом системи відразу йдуть запити в базу, наприклад зміна групи, додавання нової колонки, додавання чи зміна оцінки – все йде через базу. Для зручної роботи з базою було описано

загальну схему запитів на GraphQL, запити відправляються за допомогою js інструмента axios.

3.3. Тестування та опис роботи

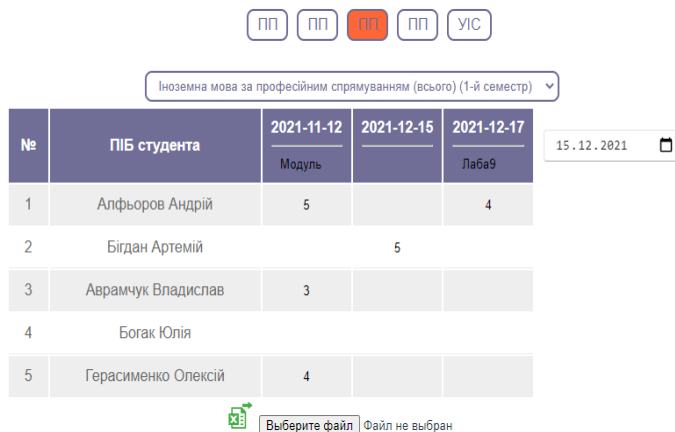
Оглянемо функції в роботі. Нижче на рисунку 3.4 показано вже деякий результат роботи модуля.



№	ПІБ студента	2021-11-12	2021-12-17
		Модуль	Лаба9
1	Алфьоров Андрій	5	4
2	Бігдан Артемій		
3	Аврамчук Владислав	3	
4	Богак Юлія		
5	Герасименко Олексій	4	

Рисунок 3.4 – Початковий вигляд перед тестом

Давайте спробуємо додати сюди нову колонку та додамо туди оцінку одному зі студентів, як показано на рисунку 3.5.



№	ПІБ студента	2021-11-12	2021-12-15	2021-12-17
		Модуль		Лаба9
1	Алфьоров Андрій	5		4
2	Бігдан Артемій		5	
3	Аврамчук Владислав	3		
4	Богак Юлія			
5	Герасименко Олексій	4		

Рисунок 3.5 – Результат тесту функцій додавання

На рисунку 3.5. видно, що колонка успішно було додана та студенту під порядковим номером 2 була поставлена оцінка 5.

Далі приберемо всі типи з оцінок, це показано на рисунку 3.6.

№	ПІБ студента	2021-11-12	2021-12-15	2021-12-17
1	Алфьоров Андрій	5		4
2	Бігдан Артемій		5	
3	Аврамчук Владислав	3		
4	Богак Юлія			
5	Герасименко Олексій	4		

Рисунок 3.6 – Результат тесту функцій оновлення

Тепер відкриємо групу, в якій ще немає студентів та спробуємо додати їх за допомогою функції імпорту Excel, як показано на рисунку 3.7.



Додайте предмети до цієї групи.

В цій групі немає студентів.

Рисунок 3.7 – Таблиця без студентів

Тепер скористаємось функцією імпорту, як показано на рисунку 3.8.



Додайте предмети до цієї групи.

№	ПІБ студента
1	Фамилия1 Имя1
2	Фамилия4 Имя4
3	Фамилия2 Имя2
4	Фамилия3 Имя3

Рисунок 3.8 – Результат функцій імпорту

З вище проведених тестів можна побачити повну працездатність системи та готовність до застосування на кафедрі.

Всі функції на додавання нових оцінок, оновлення існуючих оцінок, додавання нових дат та можливість задавати тип оцінок реалізовано за допомогою клієнта axios, який базується на javascript promises та відправляє запити з клієнтської частини до серверу. GraphQL технологія, а конкретніше розроблена під модуль схема на основі GraphQL, виконала свою функцію передачу даних до бази даних.

Функції імпорту та експорту в форматі Excel було виконано за допомогою мови програмування javascript та її бібліотеки XLSX. Бібліотека призначена для конвертації даних з таблиці в excel формат і навпаки, може легко прочитати Excel файл та конвертувати дані в зручний для використання той самий json.

Вимоги до системи:

- система: Windows, MacOS, Linux та ін.;
- операційна пам'ять: 512мб;
- браузер: будь-який останніх версій (Opera, Chrome, Safari, Edge).

Розробка програми відбувалась у середовищі VSCode. VSCode – це редактор коду, який випустила компанія Microsoft. Цей інструмент для розробки є достатньо простим та легким у використанні та має чималий список підтримуваних мов програмування та технологій.

Модуль ведення обліку та успішності був реалізований за допомогою фронтенд технології Vue.js, стилізовано за допомогою CSS та тримається на сервері, написаному на мові GO, на якому хоститься база даних Postgres та зв'язок між клієнтською частиною та базою, допоміг вирішити GraphQL, як показано на рисунку 3.9.

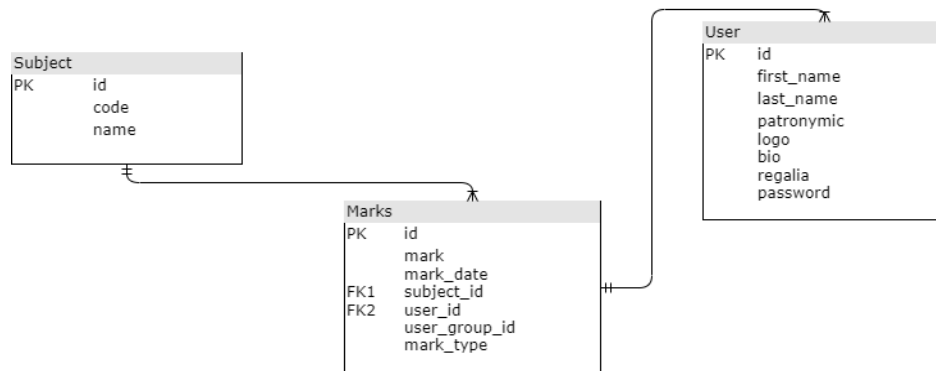


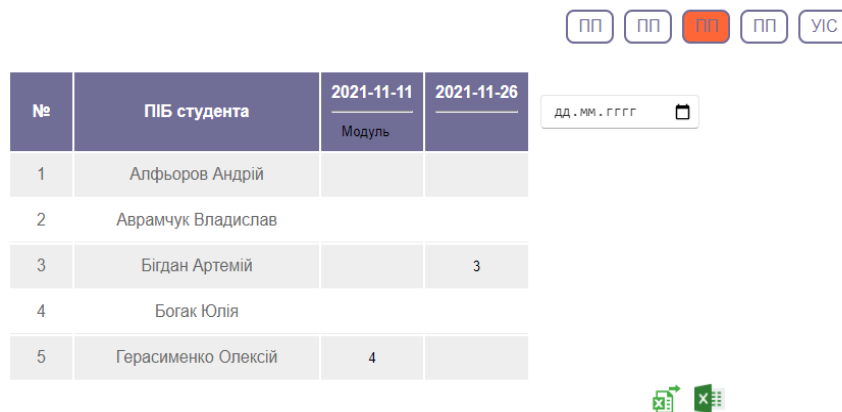
Рисунок 3.9 – UML-діаграма бази даних

Основним компонентом даного модуля являється таблиця, яка буде містити в собі список студентів та їхні досягнення в університеті у вигляді оцінок. Кожна колонка таблиці складається з дати та самій оцінок студентів. Колонка може мати наступні типи: оцінка за модуль, оцінка за лабораторну та звичайна оцінка. Типи можна записувати власно створені. Комірки таблиці напряму взаємодіють з базою, тому з кожним оновленням дані відразу направляються до серверу та не зберігаються на стороні браузера. Окрім таблиці, модуль містить в собі немаркерований горизонтальний список груп з якими можна переключатись. Додаткові та достатньо корисні функції полягають у важливості імпорту та експорту таблиць у форматі Excel, що представлено на рисунку 3.10.

№	ПІБ студента	2021-11-11	2021-11-26
1	Алфьоров Андрій		
2	Аврамчук Владислав		
3	Бігдан Артемій		3
4	Богак Юлія		
5	Герасименко Олексій	4	

Рисунок 3.10 – Основний компонент модуля – таблиця

На рисунку 3.11 представлено вигляд модуля в цілому.



The screenshot shows a web interface with a table of student data. Above the table are five navigation buttons labeled 'пп', 'пп', 'пп', 'пп', and 'уіс'. To the right of the table is a date input field with the placeholder 'дд.мм.гггг' and a calendar icon. Below the table are icons for a folder and a document.

№	ПІБ студента	2021-11-11	2021-11-26
		Модуль	
1	Алфьоров Андрій		
2	Аврамчук Владислав		
3	Бігдан Артемій		3
4	Богак Юлія		
5	Герасименко Олексій	4	

Рисунок 3.11 – Вигляд модуля в цілому

3.4. Інструкція до використання

Не зважаючи на те, що модуль буде прикріплюватись до основної системи, яка вже має свій сервер, цей модуль може працювати незалежно від нього, який має свій так званий тестовий сервер, який був розроблений за допомогою технології Node.js та express.js. Для запуску модуля треба відкрити вікно команд у папці проекту та запустити сервер за допомогою команди `npm run serve` та для самого сервера – `node server.js`. Сервер Vue за замовчуванням відповідає порту 8080, тому щоб перейти до перегляду модуля, потрібно відкрити браузер та ввести в адресній строці наступне: `localhost:8080`.

Основний файл програми має розширення .vue, який складається з 3 основних блоків кода: блок основної розмітки сторінки (HTML код), CSS блок для стилізування та javascript код, який містить в собі основну логіку роботи компонентів та зв'язок даних клієнта з сервером.

Компонент розроблений не для окремого використання, а для впровадження до інформаційної системи та функціонування з іншими модулями. Далі опишемо мінімальні вимоги до системи, вони будуть примітивні, так як браузер є на будь-якій пристрої, а наша система розроблена у вигляді веб-сайту.

ВИСНОВКИ

У рамках кваліфікаційної роботи було розроблено і реалізовано систему автоматизації процесів кафедри «Електронна кафедра». Використано методи порівняння та аналізу автоматизованих інформаційних систем. Проведено дослідження засобів та принципів побудови інформаційних систем автоматизації діяльності кафедри.

Під час виконання роботи були отримані наступні результати:

1. Проведено аналіз існуючих інформаційних систем та їх вплив на освітню сферу. Показано, що інформаційні системи широко застосовуються в освіті для автоматизації діяльності. На сьогоднішній день є необхідність у створенні системи автоматизації діяльності кафедри Прикладних інформаційних систем.

2. Досліджено можливі рішення та підходи для створення ефективної інформаційної системи та підібрано актуальні технології для повного вирішення поставленої задачі. Система має клієнт-серверну архітектуру, для розробки клієнтської частини було використано популярний та гнучкий фреймворк Vue.js, який добре підходить для розробки нашої системи.

3. Спроектовано, реалізовано та впроваджено модуль ведення обліку та успішності студентів для системи автоматизації процесів діяльності кафедри «Електронна кафедра». Була здійснена практична реалізація модуля ведення обліку та успішності студентів. Робота була виконана у середовищі VSCode за допомогою наступних технологій: Vue.js, Node.js, GraphQL.

Результати дослідження апробовано у вигляді тез доповідей під час II Міжнародної наукової конференції «Наукові орієнтири: теорія та практика досліджень» [41].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Павлиш, В. А., Гліненко, Л. К. (2013). Основи інформаційних технологій і систем: навч. посібник. *Київ–Львів: Вид-во Львівської політехніки*. Соколов, В. Ю. (2010). Інформаційні системи і технології: навч. посібник. *Київ: ДУІКТ*, 500 с.
2. Шило, С. Г., Щербак, Г. В., Огурцова, К. В., Огурцова, Е. В. (2013). Інформаційні системи та технології: навч. посібник, 138 с.
3. Іванов В.Г., Іванов С.М. (2014). Сучасні інформаційні системи та технології: навч. посібник. *Харків, НЮУ ім. Ярослава Мудрого*, 129 с.
4. Табунщик, Г. В., Кудерметов, Р. К., Притула, А. В. (2011). Проектування, моделювання та аналіз інформаційних систем: навч. посібник. *Запоріжжя: ЗНТУ*, 115 с.
5. Automation's role in education. URL: <https://www.uipath.com/blog/industry-solutions/teach-me-automations-role-in-education#:~:text=It%20will%20allow%20schools%20to,address%20gaps%20in%20their%20teaching> (дата звернення 13.11.2023).
6. Automation in education system. URL: <https://www.creatrixcampus.com/blog/automation-education-system-more-just-software> (дата звернення 15.11.2023).
7. Robson, E., & Freeman, E. (2005). *Head First Html With CSS & XHTML*. “O'Reilly Media, Inc.”, pp. 119-122.
8. Node.js. URL: <http://nodejs.org/api/> (дата звернення 17.11.2023).
9. Lubbers, P., Albers, B., Salim, F., & Pye, T. (2011). *Pro HTML5 programming*, New York, NY, USA.: Apress, pp. 107-133.
10. Romano Jr, N. C., Nunamaker Jr, J. F., Briggs, R. O., & Vogel, D. R. (1998). Architecture, design, and development of an HTML/JavaScript Web-based Group Support System. *Journal of the American Society for Information Science*, 49(7), pp. 649-667.

11. Visual Studio Code. URL: https://wikipedia.org/wiki/Visual_Studio_Code (дата звернення 19.11.2023).
12. Карпенко, М. Ю., Манакова, Н. О., Гавриленко, І. О. (2017). Технології створення програмних продуктів та інформаційних систем: навч. посібник, 189 с.
13. Лазор Я. Поняття та види інформаційних систем. URL: <http://aphd.ua/publication-146/> (дата звернення 19.11.2023).
14. Інформаційні системи. URL: https://pidru4niki.com/1222090547713/informatika/informatsiyni_sistemi (дата звернення 22.10.2023).
15. Computer Information Systems in Education. URL: https://en.wikibooks.org/wiki/Computer_Information_Systems_in_Education/Chapter_2/Section_1_--_How_do_we_use_IS_in_schools%3F (дата звернення 24.11.2023).
16. GraphQL with server. URL: <https://snipcart.com/blog/graphql-nodejs-express-tutorial> (дата звернення 26.11.2023).
17. Давидова, І. О. (2017). Управління електронними інформаційними ресурсами. Питання методології та організації: збірник наук. *Вісник Харківської державної академії культури*, вип, 10, 55-61 с.
18. Калюжний, Р. А. (2007). Інформаційна культура. *Ірпінь: Національний ун-т ДПС України*, 215-219 с.
19. Дячук, В. І. (2011). Основні підходи до автоматизації обліку на підприємстві, їх проблеми та вплив на працю облікового персоналу. *Економічні науки. Серія: Облік і фінанси*, (8), 137-143 с.
20. Татарчук, М. І. (2005). Корпоративні інформаційні системи: навч. посібник. *К.: КНЕУ*, 111-128 с.
21. Місюрів, А. В. (2007). Інформаційні системи і технології в обліку: навч. посібник, 56-58 с.
22. Береза, А. М. (2001). Основи створення інформаційних систем. *К.: КНЕУ*, 34 с.

23. Гуржій, А. М., Поворознюк, Н. І., Самсонов, В. В. (2007). Інформатика та інформаційні технології, 87 с.
24. Інформаційні технології. URL: http://users.unicyb.kiev.ua/~boiko/it/it_intro1.htm. (дата звернення 25.11.2023).
25. Грицунов О. В. Інформаційні системи та технології. URL: http://eprints.kname.edu.ua/20889/1/Gritsunov_2.pdf. (дата звернення 24.10.2023).
26. Вендров, А. М. (1998). CASE-технології. Сучасні методи і засоби проектування інформаційних систем. М.: *Фінанси та статистика*. Карпова, Т. (2002). Бази даних: моделі, розробка, реалізація, 304 с.
27. Береза А. М. (1998) Основи створення інформаційних систем: навч. посібник. Київ, *КНЕУ*, 140 с.
28. Ситник В.Ф. (2001) Основи інформаційних систем: навч. посібник. Київ, *КНЕУ*, 420 с.
29. Musciano, C., & Kennedy, B. (2002). *HTML & XHTML: The Definitive Guide: The Definitive Guide*. «O'Reilly Media, Inc.», pp. 121-126.
30. Тарасов, О. В., Федько, В. В., Лосєв, М. Ю., Тарасов, А. В., & Лосєв, М. Ю. (2011). Організація баз даних та знань. Проектування баз даних: навчально-практичний посібник для самостійної підготовки студентів, 64 с.
31. Береза, А. М. (2001). Основи створення інформаційних систем. К.: *КНЕУ*, 78 с.
32. Місюров, А. В. (2007). Інформаційні системи і технології в обліку. Навч. посібник, 41-46 с.
33. Любович, А. А., Єсіна, О. Г. (2015). Сучасні інформаційні технології в освіті. In *Інформатика та інформаційні технології: студ. наук. конф* (Vol. 20, 118-120 с.
34. Легомінова, С. В. (2015). Теоретичні засади інформаційної безпеки підприємства. *Економіка. Менеджмент. Бізнес*, (3), 87-92 с.

35. Корченко, О. Г., Терейковський, І. А., & Дзюбаненко, А. В. (2014). Сучасні нейромережеві методи та моделі оцінки параметрів безпеки ресурсів інформаційної системи, 125 с.
36. Собчук, В. В., Мусієнко, А. П., & Ільїн, О. Ю. (2018). Аналіз використання ієрархічної структури для забезпечення функціональної стійкості автоматизованої системи управління підприємством. *Телекомунікаційні та інформаційні технології*, (4), 53-61 с.
37. Орлова, Н. С., & Удовік, А. С. (2012). Інформаційні системи у сучасному корпоративному управлінні. *Актуальні проблеми державного управління*, (1), 35-40 с.
38. Карпенко С.Г., Попов В.В., Тарнавський Ю.А (2004) Інформаційні системи і технології. *Київ, МАУП*, 192 с.
39. Волчкова, Г. В. (2020). Інформатика та інформаційні технології, 59 с.
40. Мельниченко, О. А., & Орлик, О. В. Сучасні інформаційні технології в освіті. *Інформатика та інформаційні технології: студ. наук. конф* (Vol. 20, 28-31 с.
41. Пашук О.Г. (2023) Методи та засоби побудови інформаційної системи процесів автоматизації діяльності кафедри, 122 с.
42. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Tools for fast metric data search in structural methods for image classification, *IEEE Access*, 10, p. 124.
43. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.
44. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), 5-12 с.

45. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), 5-13 с.
46. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.
47. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.
48. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.
49. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium*, September 28, 2023, Kyiv-Uzhorod, Ukraine, pp. 25-27.
50. Yakovleva O., Kovač M., Ardasov V. & Yeremenko I. (2023). Study on adding functionality to the Zoom online conference system for monitoring the participant activities, *Public Administration and Regional Development*, 19(1), pp. 158-184.