

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень


Інформаційна web-система для ІТ школи
Назва теми

КВРІСТ. 200195.01.07.10 ПЗ
Шифр

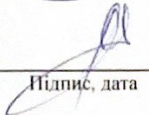
Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»
Шифр, назва

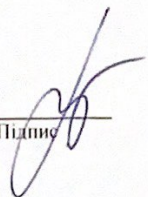
Освітня програма «Інформаційні системи та технології»
Назва

Виконав: студент III курсу, група ICTc-20-1  К.А. Костенко
Підпис Ініціали, прізвище

Керівник  К.Ю. Бобровнікова
Підпис, дата Ініціали, прізвище

Нормоконтролер  С.М. Лисенко
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем

 Т.О. Говорущенко
Підпис Ініціали, прізвище

« 5 » червня 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

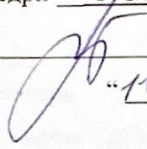
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко


..11.. 01 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Костенко Кирило Андрійович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна web-система для ІТ школи

Керівник проекту (роботи) Бобровнікова К.Ю., доцент кафедри КІС.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2023 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Аналіз відомих засобів та інформаційних систем навчальної діяльності

Вибір засобів реалізації та розробка інформаційної web-системи для ІТ-школи

Реалізація інформаційної web-системи для ІТ-школи


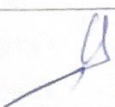


5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Схема бази даних

Схема структурна варіантів використання

Креслення вигляду екранних форм

6. Консультанти розділів дипломного проекту (роботи)

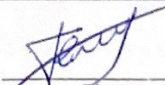
| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|---|---|
| | | завдання видав | завдання прийняв |
| Нормоконтроль | Лисенко С.М., професор кафедри КПС |  |  |
| Антиплагиат | Нічепорук А.О., доцент кафедри КПС |  |  |

7. Дата видачі завдання « 11 » 01 2023 р.

КАЛЕНДАРНИЙ ПЛАН

| №з/п | Назва етапів (розділів) дипломного проекту (роботи) | Термін виконання етапів проекту (роботи) | Примітка |
|------|---|--|----------|
| 1 | Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником | 11.01.2023 | виконано |
| 2 | Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження | 01.02.2023 | виконано |
| 3 | Робота над розділом 1 – аналіз відомих засобів та рішень | 01.03.2023 | виконано |
| 4 | Робота над розділом 2 – Вибір засобів реалізації та розробка інформаційної web-системи для ІТ-школи | 01.04.2023 | виконано |
| 5 | Робота над розділом 3 – Реалізація інформаційної web-системи для ІТ-школи | 30.04.2023 | виконано |
| 6 | Оформлення пояснювальної записки згідно вимог | 10.05.2023 | виконано |
| 7 | Попередній захист ВКР | 26.05.2023 | виконано |
| 8 | Захист ВКР на засіданні ЕК | Червень 2023 року | |

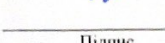
Студент


Підпис

Костенко К.А.

Ініціали, прізвище

Керівник проекту (роботи)


Підпис

Бобровнікова К.Ю.

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна web-система для ІТ школи».

Автор роботи: *Костенко Кирило Андрійович*.

Керівник роботи: *Бобровнікова Кіра Юліївна*.


Пояснювальна записка: 56 с., 35 рис., 6 табл., 3 дод., 60 джерел.

Графічна частина: 7 презентаційних слайдів

ІНФОРМАЦІЙНА WEB-СИСТЕМА, БАЗА ДАНИХ, MVC, ПЛАТФОРМА,
ІТ-КУРСИ

Мета кваліфікаційної роботи: автоматизація обліку студентів, курсів та інструкторів для ІТ школи шляхом розробки інформаційної web-системи.

Сьогодні існує досить багато різних видів освіти, які можна розділити за різними критеріями. Проте основними є два види: формальна та неформальна освіта. Формальна освіта – це освіта, яка надається в інституціях формальної освіти, таких як школи, коледжі та університети. Вона зазвичай передбачає формальну систему оцінювання та видачу сертифікатів або дипломів. В той самий час неформальна освіта відбувається поза інституціями формальної освіти (наприклад, курси, тренінги, воркшопи та інші навчальні заходи) та може бути організована комерційними або некомерційними організаціями, а також державними установами. Однією із популярних форм неформальної освіти є ІТ курси, які можуть бути корисні для людей з різних сфер життя, незалежно від їхнього досвіду та знань у галузі технологій та можуть допомогти покращити кар'єру, оновити знання, покращити бізнес, здобути нові навички та розвинути творчість. Невід'ємною частиною організації ІТ курсів є наявність інформаційної web-системи, на яку покладаються різноманітні функції по обробці та збереженню інформації необхідної для задоволення потреб користувачів. Таким чином розробка інформаційної web-системи для ІТ школи є дуже актуальною темою в сучасному світі інформаційних технологій

Підпис студента 

Дата 05.06.2023

ЗМІСТ

| | |
|---|----|
| СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ | 4 |
| ВСТУП..... | 5 |
| 1 АНАЛІЗ ВІДОМИХ ЗАСОБІВ ТА ІНФОРМАЦІЙНИХ СИСТЕМ НАВЧАЛЬНОЇ ДІЯЛЬНОСТІ | 7 |
| 1.1 Загальні принципи побудови інформаційних систем..... | 7 |
| 1.2 Принципи функціонування інформаційних web-систем для забезпечення навчальної діяльності..... | 9 |
| 1.3 Огляд відомих інформаційних web-систем для забезпечення навчальної діяльності..... | 17 |
| 1.4 Постановка задачі..... | 24 |
| 2 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ WEB- СИСТЕМИ ДЛЯ ІТ-ШКОЛИ | 25 |
| 2.1 Вибір засобів реалізації..... | 25 |
| 2.1.1 Патерн проектування MVC..... | 25 |
| 2.1.2 Entity Framework | 27 |
| 2.1.3 Веб-сервер IIS Express..... | 28 |
| 2.2 Функційні та нефункційні вимоги до інформаційної web-системи для ІТ- школи | 29 |
| 2.3 Розробка концептуальної моделі бази даних..... | 31 |
| 2.4 Висновки до розділу 2..... | 43 |
| 3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ WEB-СИСТЕМИ ДЛЯ ІТ-ШКОЛИ..... | 44 |
| 3.1 Структура інформаційної web-системи для ІТ-школи | 44 |
| 3.2 Реалізація прототипу інформаційної web-системи для ІТ-школи | 46 |
| 3.3 UML діаграм для візуалізація роботи інформаційної системи | 54 |
| 3.4 Висновки до розділу 3..... | 58 |
| ВИСНОВКИ | 59 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ | 61 |

| | | | | | | | | |
|----------|------|-------------------|--------|-------|--|----------------|-------|---------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | | | |
| Зм. | Арк. | №докум. | Підпис | Дата | Інформаційна web-система для ІТ школи | Літера | Аркуш | Аркушів |
| Виконав | | Костенко К.А. | | | | | 2 | 62 |
| Перевір. | | Бобровникова К.Ю. | | | | | | |
| Н.контр. | | Лисенко С.М. | | | | | | |
| Затвер | | Говорушенко Т.О. | | 05.06 | | | | |
| | | | | | | ХНУ, ІСТс-20-1 | | |

| | |
|--|----|
| Додаток А Копія креслення «Екранні форми»..... | 67 |
| Додаток Б Копія креслення «Схема бази даних»..... | 68 |
| Додаток В Копія креслення «Схема структурна варіантів використання»..... | 69 |

| | | | | | | |
|------|------|---------|--------|------|----------------------------|------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. |
| Зм.. | Арк. | №докум. | Підпис | Дата | | 3 |

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – База даних

ІС – Інформаційна система

ІТ – Інформаційні технології

СУБД – Система управління базами даних

EF – Entity Framework

JSON – запис об'єктів JavaScript.

MVC – Model View Controller

| | | | | | | |
|-----|------|---------|--------|------|----------------------------|------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. |
| | | | | | | 4 |
| Зм. | Арк. | №докум. | Підпис | Дата | | |

ВСТУП

На сьогоднішній день існує досить багато різних видів освіти, які можна розділити за різними критеріями. Проте основними є два види: формальна та неформальна освіта. Формальна освіта – це освіта, яка надається в інституціях формальної освіти, таких як школи, коледжі та університети. Вона зазвичай передбачає формальну систему оцінювання та видачу сертифікатів або дипломів. В той самий час неформальна освіта відбувається поза інституціями формальної освіти (наприклад, курси, тренінги, воркшопи та інші навчальні заходи) та може бути організована комерційними або некомерційними організаціями, а також державними установами. Однією із популярних форм неформальної освіти є ІТ курси, які можуть бути корисні для людей з різних сфер життя, незалежно від їхнього досвіду та знань у галузі технологій та можуть допомогти покращити кар'єру, оновити знання, покращити бізнес, здобути нові навички та розвинути творчість. Невід'ємною частиною організації ІТ курсів є наявність інформаційної web-системи, на яку покладаються різноманітні функції по обробці та збереженню інформації необхідної для задоволення потреб користувачів (стейкхолдерів).

Таким чином розробка інформаційної web-системи для ІТ школи є дуже актуальною темою в сучасному світі інформаційних технологій, що продиктовано в першу чергу наступним чинниками:

1. Зростання попиту на ІТ-фахівців: сьогодні ІТ-галузь є однією з найшвидше зростаючих і найбільш перспективних галузей, тому зростає попит на ІТ-спеціалістів. Розробка web-системи для ІТ-школи може бути корисною для збільшення кількості студентів та підготовки їх до кар'єри в ІТ, що дозволить тим самим значно знизити «поріг входження» у сферу ІТ.

2. Ефективне навчання: використання web-систем для навчання є ефективним способом підвищення якості навчання та забезпечення більш гнучкого підходу до навчання.

| | | | | | | |
|-----|------|---------|--------|------|----------------------------|------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. |
| | | | | | | 5 |
| Зм. | Арк. | №докум. | Підпис | Дата | | |

3. Модернізація навчального процесу: розробка web-системи для ІТ-школи може допомогти модернізувати навчальний процес та забезпечити студентам більш сучасний та зручний спосіб навчання.

4. Впровадження нових технологій: розробка web-системи для ІТ-школи може бути корисною для впровадження нових технологій в навчальний процес та підготовки студентів до роботи з новітніми технологіями.

Отже, можна відзначити, що розробка інформаційної web-системи для ІТ школи є дуже актуальною темою, яка може допомогти покращити навчальний процес та підготувати студентів до професійної кар'єри в галузі інформаційних технологій.

Метою роботи є автоматизація обліку студентів, курсів та інструкторів для ІТ школи шляхом розробки інформаційної web-системи.

Об'єкт дослідження є процес обліку студентів, курсів та інструкторів для ІТ школи.

Предмет дослідження є інформаційна web-система для ІТ школи.

| | | | | | | |
|-----|------|---------|--------|------|----------------------------|------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. |
| | | | | | | 6 |
| Зм. | Арк. | №докум. | Підпис | Дата | | |

1 АНАЛІЗ ВІДОМИХ ЗАСОБІВ ТА ІНФОРМАЦІЙНИХ СИСТЕМ НАВЧАЛЬНОЇ ДІЯЛЬНОСТІ

1.1 Загальні принципи побудови інформаційних систем

В загальному існують два шляхи створення інформаційних систем на основі інтегрованих баз даних, кожен з яких має свої переваги та недоліки. Перший шлях, заснований на глибокому аналізі всіх бізнес-процесів та існуючої системи документообігу на підприємстві (наприклад ІТ школі), передбачає створення добре структурованої єдиної бази даних на сервері та розробку інформаційних систем для користувачів усіх підрозділів на загальній платформі на основі товстого або тонкого клієнта. Перевагами такого підходу є те, що підприємство отримує сучасну інформаційну систему адміністрування, в якій враховано всі основні інформаційні потреби, реалізовано необхідні запити та сформовано необхідні вихідні документи. Однак серйозною проблемою при розробці даної системи є її масштаби та висока вартість.

Другий підхід, більш економічний, ґрунтується на аналізі потреб адміністрації в інформаційних ресурсах та можливостях вже існуючих інформаційних систем надавати ці ресурси. За такого підходу на сервері створюється адміністративна мережева інформаційна система з урахуванням сучасних систем управління базами даних, і інструментальних систем розробки клієнтського інтерфейсу (наприклад, Web-інтерфейса). Необхідні додаткові інформаційні системи купуються, а окремі модулі (для конвертування даних із різних інформаційних систем в інтегровану серверну базу даних чи навпаки) розробляються вже самотужки. Це дає можливість, не руйнуючи наявні інформаційні системи, які успішно функціонують, забезпечити адміністративні служби онлайн-доступом до різних даних. При такому підході розробку чи модернізацію окремих інформаційних систем можна виконувати поетапно з урахуванням вже сформованих вимог до баз даних та програмного забезпечення. Головними проблемами такого підходу є супровід та підтримання у робочому

| | | | | | | |
|-----|------|---------|--------|------|----------------------------|-----------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. 7 |
| Зм. | Арк. | №докум. | Підпис | Дата | | |

стані застарілих локальних інформаційних систем, а також конвертування даних та обмін ними між різними платформами. В окремих випадках заміна в адміністративній службі комп'ютера та (або) операційної системи веде до перенастроювання інформаційної системи (перекомпіляції окремих модулів) або її загибелі.

Аналіз розвитку сучасної обчислювальної техніки, програмного забезпечення, методів інженерії знань та штучного інтелекту показує, що на сучасному етапі необхідно орієнтуватися на використання комп'ютерних технологій інженерно-технічними працівниками підприємства, на вбудовування інформаційних систем у реальні технологічні процеси функціонування підприємства. Це дозволить ліквідувати (або послабити) вплив недоліків централізованої та дворівневої систем обробки даних, одночасно забезпечуючи перехід на безпаперову технологію, що за наявності технічних та програмних засобів, локальних обчислювальних мереж суттєво підвищить швидкість та якість комутаційних процесів управління. Тільки застосування автоматизованих робочих місць із урахуванням КС може автоматизувати рутину частину роботи фахівців, на виконання якої працівниками витрачається значний час. Економічний ефект цього процесу очевидний, але потрібна серйозна підготовка інфраструктури підприємства (технічне та програмне забезпечення), серйозна робота з підготовки та перепідготовки кадрів.

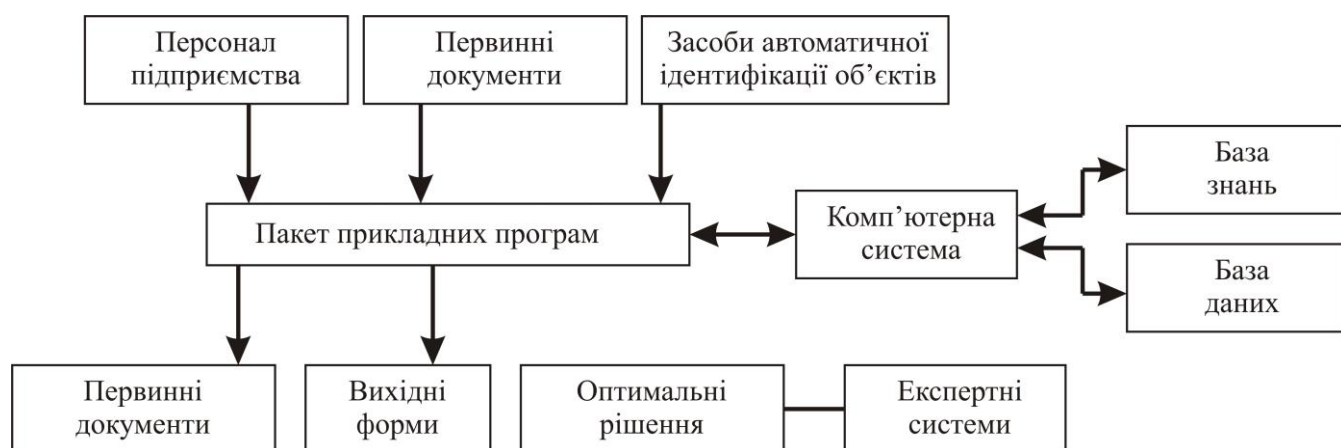


Рисунок 1.1 – Структура та базові принципи функціонування інформаційних систем

| | | | | |
|-----|------|---------|--------|------|
| Зм. | Арк. | №докум. | Підпис | Дата |
|-----|------|---------|--------|------|

Основним елементом у сучасних інформаційних системах є автоматизоване робоче місце (АРМ) – проблемно-орієнтований програмно-технічний комплекс, винесений на робоче місце та який автоматизує в режимі діалогу деякий набір управлінських процедур кінцевого користувача за його безпосередньої участі. Первинна інформація повинна вводиться в КС один раз через АРМ підрозділу, де вона виникає, а потім може бути використана будь-яким підрозділом підприємства. Пошук, обмін, обробка та аналіз інформації повинні виконуватись автоматично за допомогою прикладних програмних засобів. Крім того, при реалізації інформаційних систем необхідно дотримуватись наступних принципів:

- перегляд всієї структури та схеми документообігу підприємства, тобто скорочення до мінімуму первинної документації та (по можливості) формування їх на КС, виключення з обігу всіх вторинних та проміжних носіїв інформації;
- відділення нормативно-довідкової інформації від поточної (що постійно змінюється) та її зберігання на пристроях збереження інформації;
- використання єдиної нормативно-довідкової інформації всіма підрозділами підприємства;
- одноразове введення первинної інформації в КС з використанням усіх можливостей СУБД контролю помилок уведення;
- реалізація обміну інформацією між підрозділами підприємства через локальну комп'ютерну мережу;
- перерозподіл завдань між підрозділами підприємства з метою скорочення обмінних інформаційних потоків;
- робота всіх інформаційних підсистем як реального часу.

1.2 Принципи функціонування інформаційних web-систем для забезпечення навчальної діяльності

Освітня установа, в тому числі платформа ІТ-школи, як складна динамічна система має відповідати сучасним умовам інформаційного світу, в якому основне

| | | | | | | |
|------|------|---------|--------|------|----------------------------|------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. |
| | | | | | | 9 |
| Зм.. | Арк. | №докум. | Підпис | Дата | | |

завдання управління є необхідність впровадження комп'ютерних технологій. Стійке підвищення якості управлінських рішень освітнього установи і забезпечення їх наступності досягається за рахунок ефективного вирішення цього завдання з використанням інформаційних систем (ІС).

Інформаційну систему управління освітньою установою або платформою надання послуг навчальних курсів визначимо як комплекс апаратних і програмних засобів, призначених для управління різними процесами у межах освітньої установи або платформи надання послуг навчальних курсів. Переважна більшість інформаційних систем управління освітньою установою або платформою надання послуг навчальних курсів представляють собою веб-орієнтовані системи.

Веб-орієнтована система є клієнт-серверною програмою, в якій клієнтську частину реалізує браузер, що здійснює діалог з користувачем та відображає інформацію, а серверну частину – веб-сервер і сервер додатків, які реалізують основну логіку системи. Через обмежену функціональність клієнта, подібну реалізацію системи часто ще називають тонким клієнтом. Перевагами веб-орієнтованої системи можна визначити:

Кросплатформенність. В якості клієнта, як вже було зазначено вище, виступає веб-браузер, що означає, що для роботи з системою необхідний лише інтернет-браузер, який входить до складу будь-якої операційної системи. Враховуючи, що в даний час браузери не залежать від конкретної операційної системи користувача, веб-орієнтовані системи можна назвати платформними (або міжплатформними).

Мобільність. Працювати із системою можна перебуваючи у будь-якому місці, де є Інтернет та з будь-якого пристрою, в якому є інтернет-браузер. Таким чином, користувач не обмежений вимогами до апаратної частини. Можна працювати і в офісі, і вдома, і у відрядженні як зі стаціонарного комп'ютера, так і з ноутбука або планшета.

Низька сукупна вартість обслуговування. Вартість обслуговування веб-орієнтованою системою фактично закладається у розробці, підтримці та

| | | | | | | |
|-----|------|---------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. 10 |
| Зм. | Арк. | №докум. | Підпис | Дата | | |

розвитку серверної частини (веб-сервера та сервера додатків) та утримання сервера бази даних системи. Вартість системи управління базами даних залежить від вибору технології розробки веб-додатку: використання відкритої повністю кросплатформової або конкретної платформи для створення веб-додатків, наприклад Microsoft ASP.NET, що в свою чергу може накласти вимоги до СУБД, але в той же час, має й низку своїх переваг.

Проте, разом із тим як і будь-які інші системи, веб-орієнтовані системи також мають власні недоліки:

Для роботи з системою необхідне підключення до мережі Інтернет. Таким чином, реальна мобільність може суттєво відрізнятися від заявленої з незалежних від користувача причин.

Не всі системи можуть бути замінені на веб-орієнтовані через обмеженість функціональних можливостей інтернет-браузерів. Якщо мова йде про просту роботу з документами, то складнощів не виникає, але, наприклад, робота зі специфічною документацією або необхідність відображення тривимірних моделей може накласти обмеження на використання тонкого клієнта.

Вся інформація, включаючи особисту інформацію користувача, зберігається на сервері. Цей факт накладає серйозні вимоги щодо захисту інформації на сервері та захисту каналів передачі даних. Не кожен клієнт готовий передавати корпоративну та особисту інформацію через Інтернет. Якій системі віддати перевагу залежить від потреб і вимог конкретної організації. Але варто зауважити, що ринок веб-орієнтованих систем останнім часом стрімко розвивається через популярність таких аспектів як мобільність та платформна незалежність. А це означає, що йде безперервна робота над удосконаленням веб-орієнтованих систем та поступовим усунення їх недоліків.

Існує значна кількість підходів до створенню інформаційної системи, які дозволяють реалізувати всі існуючі інформаційні процеси, зокрема:

- збір інформації;
- зберігання інформації;

- обробка інформації;
- видача інформації.

Процес зберігання полягає в зберіганні зібраної і обробленої інформації на хостингах в вигляді структурованих масивів даних, визначених як база даних. Обробка інформації здійснюється по певним алгоритмам і є однією з основних інформаційних операцій.

Збір і видача інформації здійснюється через користувальницький інтерфейс розроблений для цих цілей, наприклад, Web-додатки.

Web-орієнтована інформаційна система, як правило, призначена для виходу в світове інформаційне простір мережі Інтернет і для внутрішніх потреб замовника.

Користувачі системи поділяються на наступні групи (ролі):

- адміністративний персонал;
- співробітники підрозділи;
- зареєстровані користувачі;
- відвідувачі веб-сайту.

Ефективність функціонування такої інформаційної системи досягається за рахунок вільного доступу зазначених користувачів в комп'ютерну мережу організації як локальну, так і глобальну.

Web-орієнтована інформаційна система припускає оцінку наступних аспектів майбутнього сайту:

- 1) визначення функціональних можливостей сайту;
- 2) оцінка навантаження на хостинг, кількість запитів до серверу;
- 3) вибір розробника з попереднім моніторингом його попередніх робіт;
- 4) розробка дизайну сайту.

Загальна концептуальна модель Web-орієнтованої інформаційної системи представлена на рис. 1.2.

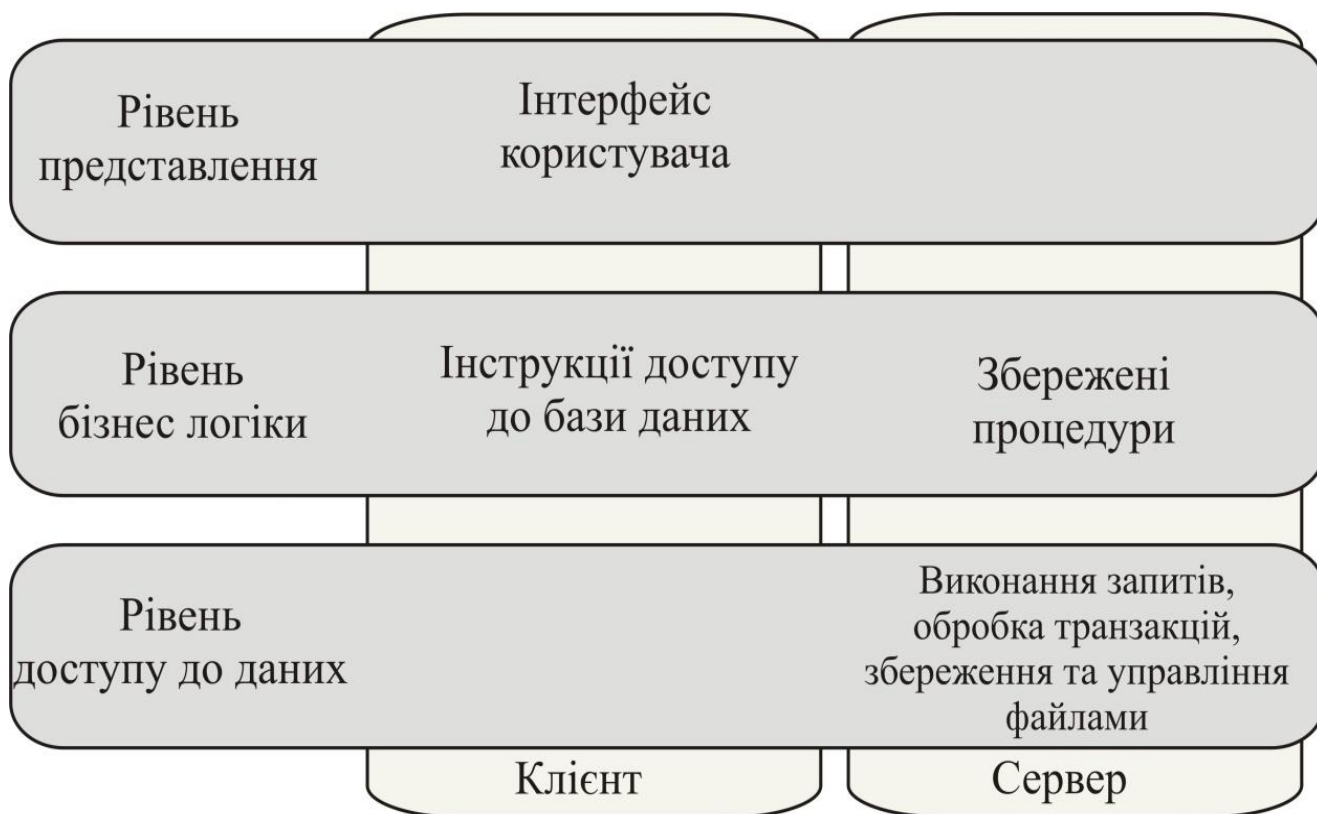


Рисунок 1.2 – Модель web-орієнтованої інформаційної системи

В основі представленої моделі web-орієнтованої інформаційної системи закладено концепцію клієнт-серверної архітектури.

Клієнт-серверна архітектура – це модель розподіленої системи, в якій функції та обов'язки розділяються між двома типами процесів: клієнтами та серверами. Клієнти - це програми або процеси, які надсилають запити серверам для отримання ресурсів або послуг. Сервери - це програми або процеси, які обробляють запити клієнтів та надають їм ресурси або послуги. Модель такої системи полягає в тому, що клієнт відправляє запит на сервер, де він обробляється, і готовий результат відправляється клієнтові. Сервер може обслуговувати кілька клієнтів одночасно. Якщо одночасно приходять більше одного запиту, то вони встановлюються в чергу і виконуються сервером послідовно. Іноді запити можуть мати пріоритети. Запити з більш високими пріоритетами повинні виконуватися раніше.

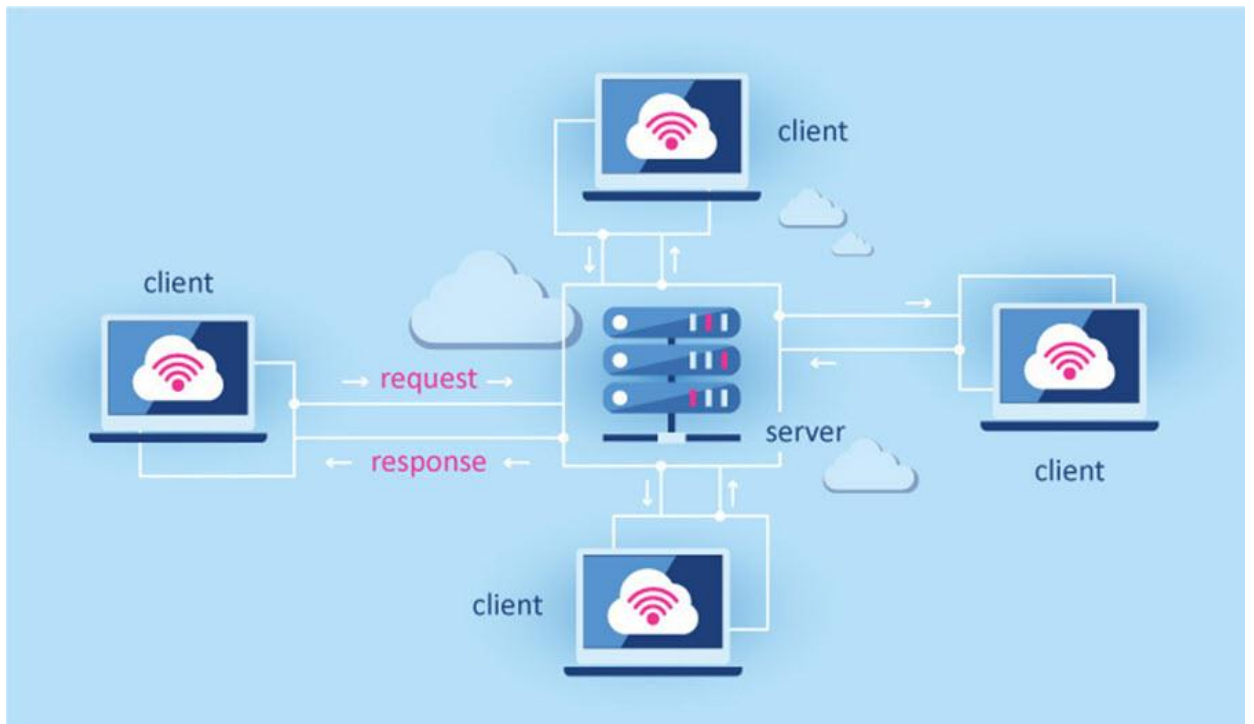


Рисунок 1.3 – Клієнт-серверна архітектура

Функції, які реалізуються на сервері:

- Зберігання, доступ, захист і резервне копіювання даних.
- Обробка клієнтського запиту.
- Відправлення результату (відповіді) клієнту.

Функції, які реалізуються на стороні клієнта:

- Надання користувальницького інтерфейсу.
- Формулювання запиту до сервера і його відправка.
- Отримання результатів запиту і відправка додаткових команд (запитів на додавання, оновлення або видалення даних).

Архітектура клієнт-сервер визначає принципи спілкування між комп'ютерами, а правила і взаємодії визначені в протоколі.

Існують концепції побудови системи клієнт-сервер:

Слабкий клієнт – потужний сервер. У такій моделі вся обробка інформації перенесена на сервер, а у клієнта права доступу суворо обмежені. Сервер відправляє відповідь, яка не вимагає додаткової обробки. Клієнт взаємодіє з

користувачем: складає та відправляє запит, приймає результат і виводить інформацію на екран.

Сильний клієнт – концепція, в якій частина обробки інформації надається клієнтові. У такому випадку сервер виступає сховищем даних, а вся робота по обробці та подання інформації переноситься на комп'ютер клієнта.

Система (додаток), яка заснована на клієнт-серверній взаємодії, включає три основних компоненти: представлення даних, прикладний компонент, компонент управління ресурсами і їх зберігання.

Існують дворівнева і триврівнева клієнт-серверні архітектури.

Дворівнева архітектура складається з двох вузлів:

- Сервер, який відповідає за отримання запитів і відправку відповідей клієнту, використовуючи при цьому лише власні ресурси.
- Клієнт, який представляє користувацький інтерфейс.

Принцип роботи полягає в тому, що сервер отримує запит, обробляє його і відповідає безпосередньо, без використання сторонніх ресурсів.

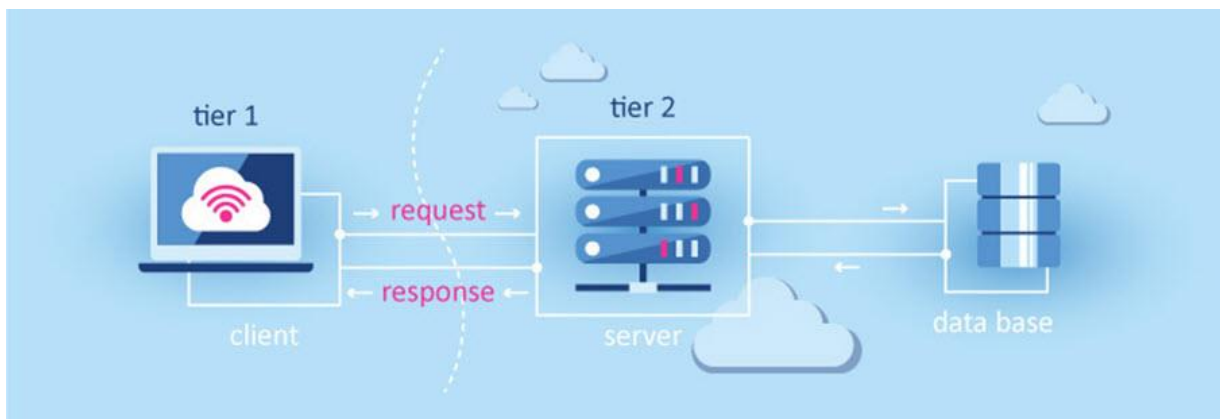


Рисунок 1.4 – Дворівнева клієнт-серверна архітектура

Триврівнева архітектура складається з наступних компонентів:

- представлення даних – призначений для користувача інтерфейс;
- прикладний компонент – сервер додатків;
- керування ресурсами – сервер бази даних, який надає інформацію.

Принцип роботи полягає в тому, що декілька серверів обробляють запит клієнта. Розподіл операцій знижує навантаження на сервер.

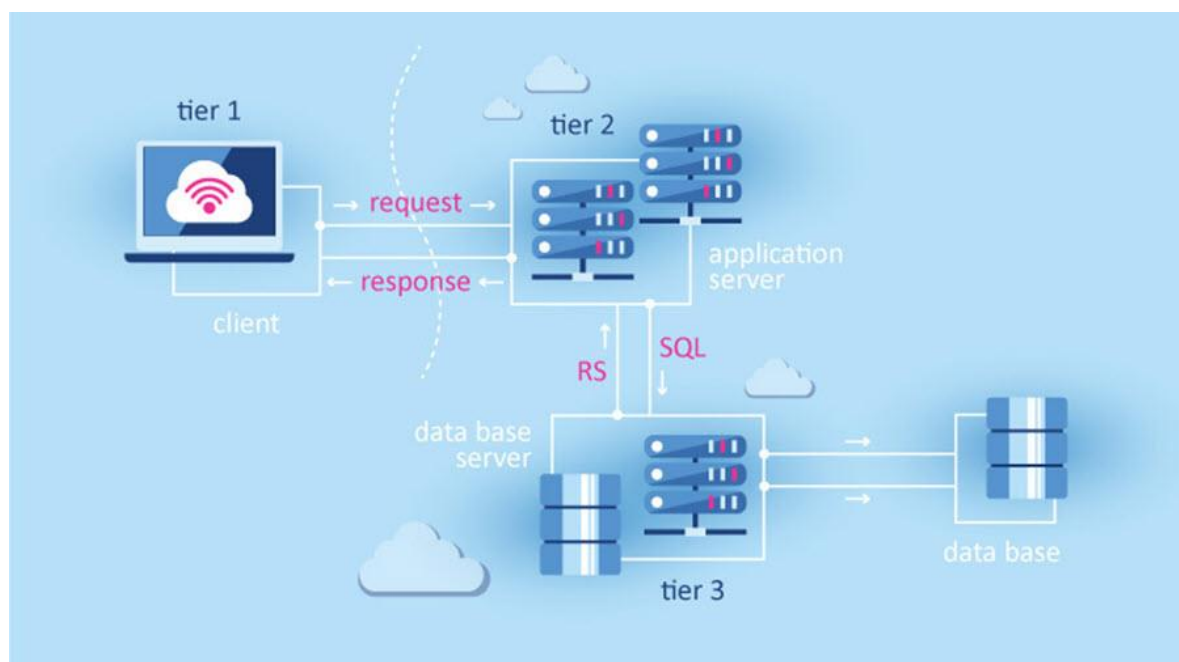


Рисунок 1.5 – Трьохрівнева клієнт-серверна архітектура

Трирівневу архітектуру можна розширити до багаторівневої (N-tier, Multi-tier) способом встановлення додаткових серверів. Багаторівнева архітектура дозволяє підвищити ефективність роботи інформаційної системи, а також оптимізувати розподіл її програмно-апаратних ресурсів.

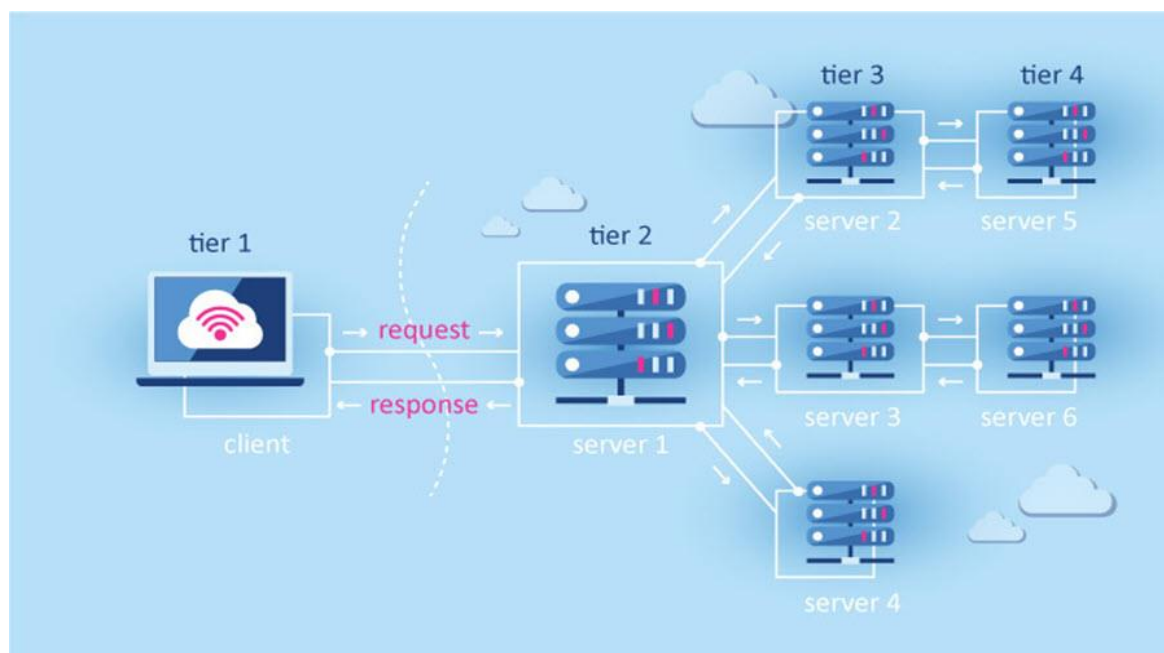


Рисунок 1.6 – Багаторівнева клієнт-серверна архітектура

курси, завдання, завдання та давати оцінки. Учні можуть виконувати різні завдання, спілкуватися з вчителями та одне з одним.

– Електронні портфоліо: Веб-системи, які дозволяють учням та студентам створювати цифрові портфоліо для відстеження та представлення своїх досягнень. Наприклад, Mahara, Seesaw.

– Системи відстеження успішності: Вони дозволяють вчителям та адміністраторам відстежувати академічну успішність учнів, аналізувати дані та надавати звіти. Наприклад, PowerSchool, Infinite Campus.

– Електронні бібліотеки та ресурси: Веб-системи, які надають доступ до електронних книг, статей, відеоматеріалів та інших освітніх ресурсів. Наприклад, EBSCOhost, JSTOR, Google Scholar.

– Системи онлайн-тестування та оцінювання: Веб-системи, які дозволяють проводити онлайн-тести та оцінювати знання учнів. Наприклад, Quizlet, Google Forms, Kahoot.

– Веб-системи для спільної роботи: Наприклад, Google Docs, Microsoft Office 365, Dropbox. Вони дозволяють учням та вчителям спільно працювати над проектами, документами та іншими завданнями віддалено у режимі реального часу.

На сьогоднішній день вищі навчальні заклади, а також різноманітні ІТ-школи мають власні інформаційні web-системи, що дозволяє їм організувати процес освітньої діяльності. Розглянемо більш детально приклади таких систем.

В Хмельницькому національному університеті для забезпечення освітньої діяльності та ведення електронного документообігу, функціонують системи «Електронний університет» та «Модульне середовище для навчання».

Система «Електронний університет» (рис.1.2) забезпечує виконання наступних основних функцій:

- Ведення журналу обліку успішності.
- Відображення розкладу (як по групах, так і викладачах).
- Відображення індивідуального плану студента.

| | | | | |
|------|------|---------|--------|------|
| | | | | |
| Зм.. | Арк. | №докум. | Підпис | Дата |

- Надає функціонал для вибору вибіркових дисциплін.
- Можливість відображення рейтингу студентів.
- Можливість надсилання та отримання повідомлень.

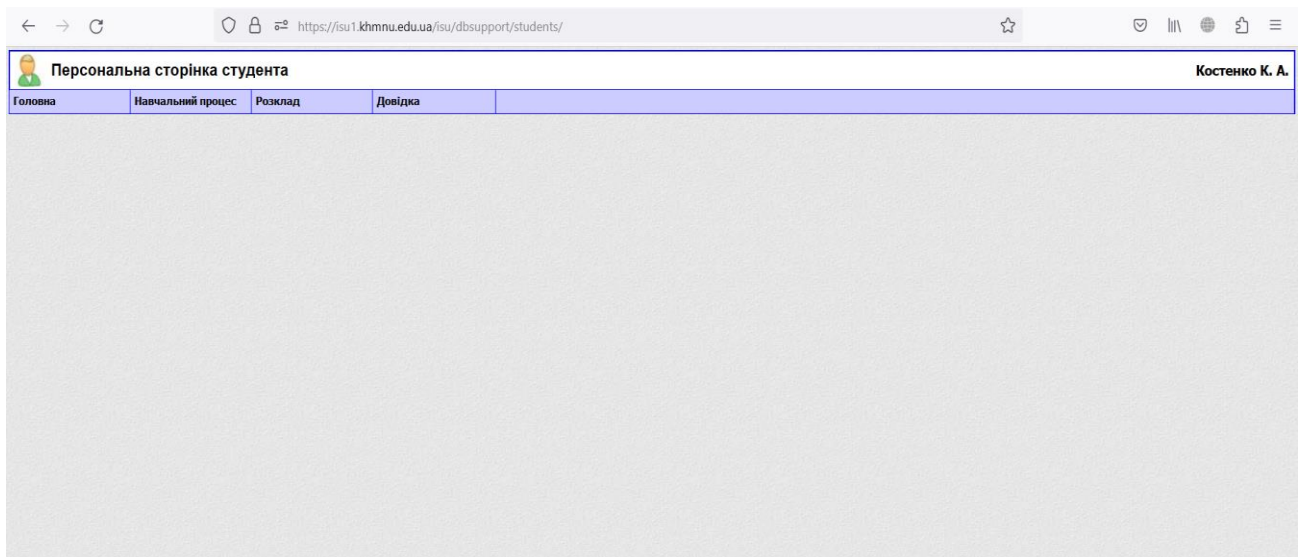


Рисунок 1.7 – Інтерфейс інформаційної web-системи «Електронний університет»

Система «Модульне середовище для навчання» є ще однією платформовою за допомогою якої забезпечується освітня діяльність у Хмельницькому національному університеті (рис. 1.8). Система «Модульне середовище для навчання» реалізована на основі платформи Moodle, що є системою управління курсами або системою управління навчанням або віртуальне середовище та є аббревіатурою від Modular Object-Oriented Dynamic Learning Environment.

Система «Модульне середовище для навчання» дозволяє реалізувати наступні функції:

- Відображати та групувати навчальні курси за кафедрами.
- Розміщувати навчальні матеріали, що можуть бути представлені як файли, тести, URL посилання, завдання, глосарії, база даних, семінари. Є можливість створення форуму.
- Надавати повний функціонал по управлінню користувачами, а саме додавати до курсу, відраховувати із курсу, призначати ролі.

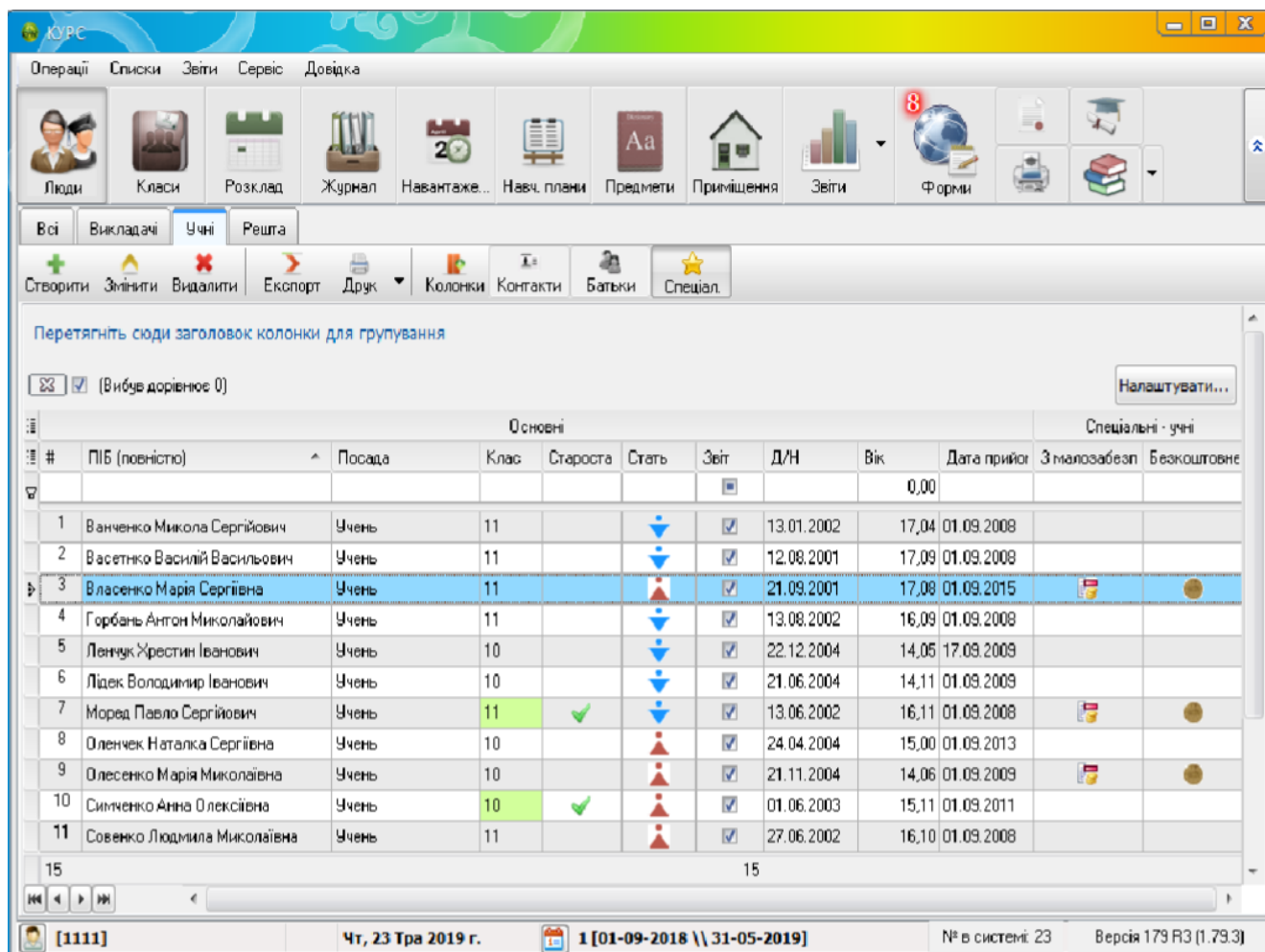


Рисунок 1.9 – Інтерфейс інформаційної web-системи “КУРС: ВНЗ”

Іншим прикладом web-системи для проведення та організації ІТ курсів є платформа від компанії Step IT Academy (рис. 1.10). Представлена web-система дозволяє ознайомити здобувачів із переліком доступних для вивчення курсів, можливих професій, які пов’язані із сферою інформаційних технологій, літніх канікул для дітей, а також містить контакту інформацію та розташування приміщень офісів компанії. Є інформація про структуру компанії та відгуки здобувачів, що пройшли курси.

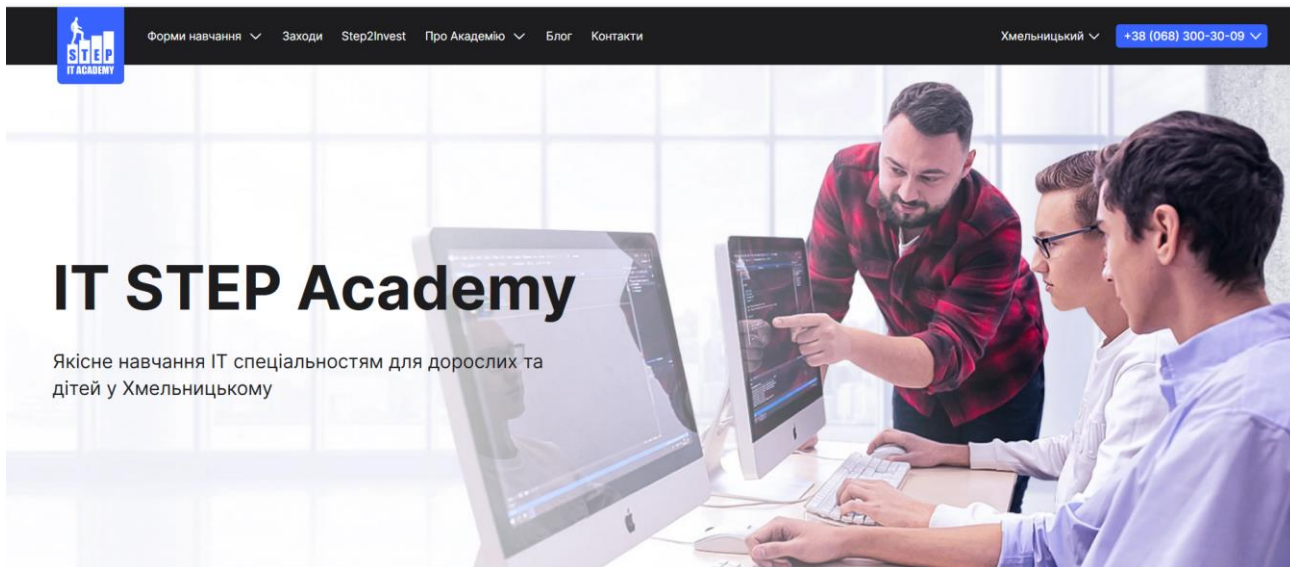


Рисунок 1.10 – Інтерфейсне вікно інформаційної web-системи IT-курсів Step Academy

Ще одним прикладом інформаційної web-системи для IT курсів є Hashtag (рис. 1.11). Дана інформаційна web-система дозволяє ознайомитись із доступними курсами, містить інформацію про академію, а також інформацію про розташування офісів. В якості недоліків даної web-системи можна віднести відсутність механізмів формування звітів та інформації про курси й активні користувачі на цьому курсі.

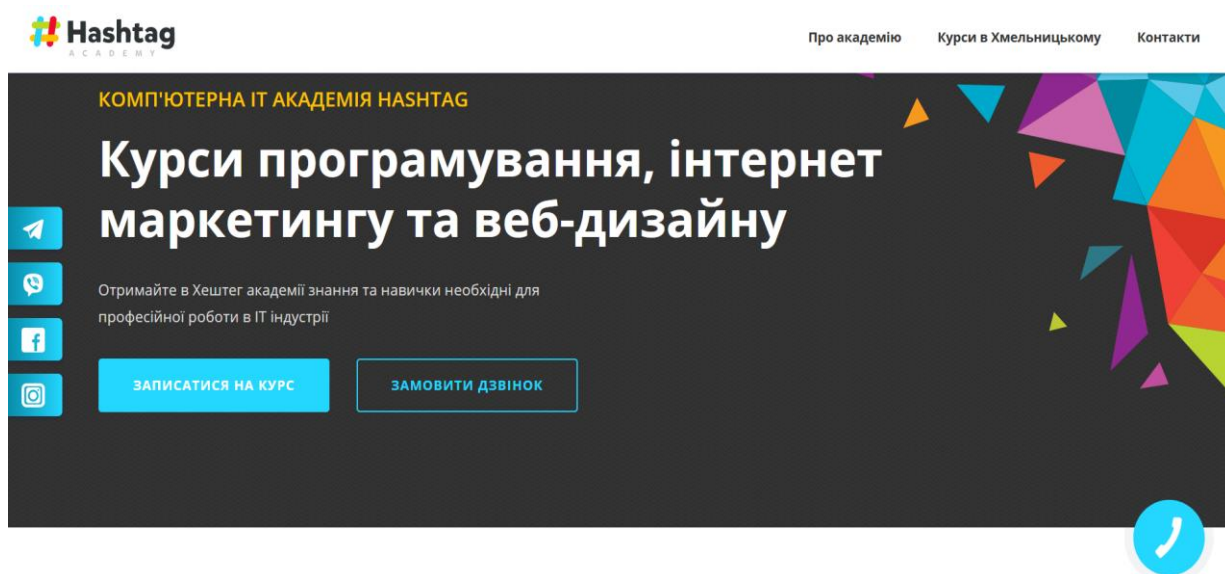


Рисунок 1.11 – Інтерфейс інформаційної web-системи IT-курсів Hashtag

1.4 Постановка задачі

На сьогоднішній день неформальна освіта є однією із важливих форм здобуття освіти для якої, в сучасних реаліях, обов'язково притаманно наявність інформаційної web-системи через яку вирішуються задачі автоматизації обліку студентів, курсів та інструкторів та інших бізнес-процесів. Проведений огляд відомих платформ та інформаційних web-систем показав, що таким система притаманні недоліки, зокрема: складність системи, відсутність простих механізмів формування звітів формування звітів та інформації про курси й активних користувачі. Тому актуальним завданням є розробка інформаційної web-системи, яка б дозволила автоматизувати процеси обліку студентів, курсів та інструкторів для ІТ школи та володіла простотою та легкістю формування звітів, а також мати простий і легкий інтерфейс, без переобтяження надмірним контентом.

Для вирішення поставленого завдання необхідно виконання наступних етапів:

1. проаналізувати відомі засоби та інформаційні системи для освітніх установ або ІТ шкіл, що надають неформальні освітні послуги;
2. визначити функційні та нефункційні вимоги до інформаційної web-системи для ІТ-школи;
3. Провести аналіз предметної області та розробити концептуальну модель бази даних;
4. Спроекувати інформаційну web-систему для ІТ-школи, привести її узагальнену структуру;
5. Реалізувати прототип інформаційної web-системи для ІТ-школи;
6. Провести аналіз отриманих результатів.

2 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ WEB-СИСТЕМИ ДЛЯ ІТ-ШКОЛИ

2.1 Вибір засобів реалізації

Проектування інформаційної web-системи передбачає розробку всіх компонентів системи, що відповідають за створення, додавання, видалення, оновлення інформації доступними програмними засобами та технологіями. При розробці інформаційної web-системи для ІТ-школи було використано наступні засоби та технології:

- Середовище розробки Visual Studio 2019.
- Технологія Entity Framework.
- Технологія MVC.
- IIS Express.

Розглянемо детальніше описані вище засоби реалізації.

2.1.1 Патерн проектування MVC

Для проектування даної інформаційної web-системи було обрано патерн MVC. Model-View-Controller або MVC є шаблоном або патерном програмування, що розділяє архітектуру системи (рис. 2.1) на три складові: модель (Model), представлення (View) та контролер (Controller). Цей патерн дозволяє змінювати кожен компонент незалежно один від одного для простої розробки та підтримки веб-застосунків. В патерні MVC:

– Модель (Model). Це основна логіка програми. Відповідає за дані, методи роботи з ними та структуру програми. Модель реагує на команди із контролера та видає інформацію та/або змінює свій стан. Вона передає дані у подання.

– Подання (View). Завданням компонента є візуалізація інформації, яку він отримує від моделі. View відображає дані на рівні інтерфейсу користувача.

Наприклад, у вигляді таблиці чи списку. Подання визначає зовнішній вигляд програми та способи взаємодії з ним.

– Контролер (Controller). Він забезпечує взаємодію з системою: обробляє дії користувача, перевіряє отриману інформацію та передає її моделі. Контролер визначає, як програма буде реагувати на дії користувача. Також контролер може відповідати за фільтрацію даних та авторизацію.

Компоненти моделі відрізняються ступенем залежності один від одного та обмеженнями:

- Модель не залежить від представлення та контролера, але й не може використовувати класи з їхніх розділів.
- Подання може звертатися до моделі за даними та подіями, але не може її змінювати.
- Контролер не може відображати дані, але здатний змінювати модель в залежності від дій користувача.

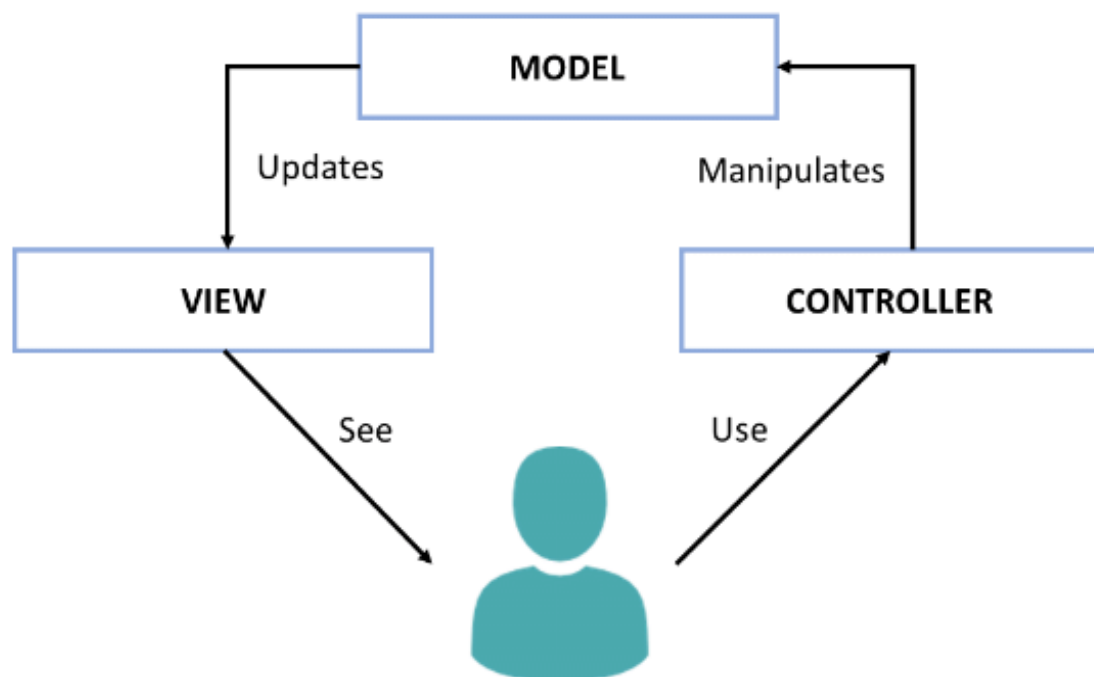


Рисунок 2.1 – Організація патерну проектування MVC

індексами, первинними та зовнішніми ключами, але на концептуальному рівні, який нам пропонує Entity Framework, ми вже працюємо з об'єктами.

Entity Framework Core підтримує багато різних систем баз даних. Таким чином, ми можемо через EF Core працювати з будь-яким СУБД, якщо для неї є потрібний провайдер.

Основними перевагами використання об'єктно-орієнтованої технології доступу до даних Entity Framework Core є:

- Підтримка основних операцій CRUD (create-read-update-delete).
- Підтримка параметризованих запитів.
- Відкритий вихідний код.
- Можливість генерації команд мови SQL із мови LINQ у сутності.
- Безпечна компіляція завдяки перевірці синтаксичних помилок та безпеки типів.
- Підтримка роботи не лише із вбудованими даними, а й також із програмованими класами.
- Не залежить та не має прив'язки від релізів фреймворку .NET.

2.1.3 Веб-сервер IIS Express

IIS Express - це безкоштовний веб-сервер від Microsoft, який призначений для розробки та тестування веб-додатків під платформу .NET. IIS Express надає можливість запускати та тестувати веб-додатки в режимі локального сервера, який може бути доступний лише на локальному комп'ютері. В свою чергу IIS Express є легковажним та безкоштовним веб-сервером, що може бути встановлений та використаний для розробки та тестування веб-додатків на платформі Windows. Деякі з його основних переваг:

- Легкість встановлення та використання: IIS Express може бути швидко встановлений з Visual Studio або завантажений окремо з офіційного сайту Microsoft, і його можна легко настроїти та запустити на комп'ютері.

– Підтримка різних версій IIS: IIS Express може працювати з версіями IIS, що встановлені на комп'ютері, тому можна використовувати його для тестування веб-додатків на різних версіях IIS.

– Підтримка різних мов програмування: IIS Express підтримує різні мови програмування, включаючи ASP.NET, PHP та Node.js, тому його можна використовувати для розробки та тестування веб-додатків на різних мовах.

– Підтримка безпеки: IIS Express має вбудовану підтримку безпеки, яка дозволяє захищати веб-додатки від атак та вразливостей.

– Підтримка віддаленого доступу: IIS Express може бути настроєний для доступу з віддаленого комп'ютера, що дозволяє використовувати його для тестування веб-додатків на віддалених серверах.

– Підтримка віртуалізації: IIS Express підтримує віртуалізацію, що дозволяє використовувати його для тестування веб-додатків в різних віртуальних середовищах.

Беручи до уваги наведені переваги різних програмних засобів та технологій, було вирішено обрати IIS Express в якості веб-сервера, а також використовувати Entity Framework та паттерн MVC для розробки інформаційної веб-системи.

2.2 Функційні та нефункційні вимоги до інформаційної web-системи для ІТ-школи

Одним з ключових етапів розробки інформаційної системи є визначення того, яким вимогам вона повинна задовольняти. Функційні вимоги є вимогами, що ставляться до функціональності системи або програмного забезпечення. Ці вимоги описують те, що система або програмне забезпечення повинні робити, які функції вони повинні виконувати та які результати мають повертати. Функційні вимоги можуть бути виражені в термінах функціональних можливостей, які система або програмне забезпечення повинні забезпечити.

Таким чином проєктована інформаційна web-система для ІТ-школи повинна забезпечувати наступні функційні вимоги:

| | | | | | | |
|-----|------|---------|--------|------|----------------------------|------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. |
| | | | | | | 29 |
| Зм. | Арк. | №докум. | Підпис | Дата | | |

- Можливість збереження даних про студентів (додавання, видалення, редагування, перегляд) та формування звіту.
- Можливість збереження даних про інструкторів (додавання, видалення, редагування, перегляд) та формування звіту.
- Можливість створення нового курсу, видалення, редагування метаданих та формування звіту.
- Можливість створення, видалення, редагування підрозділу та формування звіту.
- Перегляд інформації про систему.

Нефункційні вимоги це вимоги до якості, ефективності, безпеки, надійності, сумісності та інших характеристик інформаційної системи або програмного забезпечення. Ці вимоги описують те, як інформаційна система або програмне забезпечення повинні працювати, які стандарти вони повинні відповідати та які обмеження є на їх функціонування.

- Інформаційна система повинна володіти властивістю доступністю (система повинна функціонувати постійно, що дозволить в будь-який момент здійснити доступ до даних).

- Інформаційна система має бути безпечною та захищеною від несанкціонованого доступу.

- Інформаційна система має бути легко зрозумілою та простою в користуванні.

- Система має бути сумісною з різними операційними системами та браузерами.

Таким чином описані вищі функційні та нефункційні вимоги дозволяють окреслити рамки проектовано інформаційної web-системи.

Таким чином визначення функційних та нефункційних вимог є важливим етапом в розробці будь-якої інформаційної ситеми та дозволяє означити, функції та характеристики які має мати система, щоб задовольнити потреби кінцевих користувачів.

2.3 Розробка концептуальної моделі бази даних

Концептуальна модель бази даних (Conceptual Data Model) є абстрактною моделлю, яка описує структуру даних та їх взаємозв'язки у предметній галузі. Ця модель допомагає розробникам визначити основні сутності (entities), їх атрибути (attributes), а також зв'язки між ними (relationships) у проєктованій базі даних інформаційної системи.

Основна мета проєктування концептуальної моделі бази даних полягає в тому, щоб представити предметну область у формі, доступній для розуміння як для технічних, так і нетехнічних користувачів. Концептуальна модель є високорівневою та не враховує обмеження, пов'язані з конкретною технологією бази даних, що дозволяє легко змінювати її за потреби.

Для створення концептуальної моделі бази даних використовують різні інструменти, такі як діаграми класів UML, діаграми сутність-зв'язок (ERD) та інші. Важливо, що концептуальна модель бази даних є основою подальшої розробки логічної і фізичної моделей бази даних.

Основою запропонованої інформаційної web-системи для ІТ-школи є база даних, що зберігає інформацію про студентів, інструкторів та інші сутності. Для створення бази даних було використано об'єктно-орієнтовану технологію доступу до даних ADO.NET Entity Framework. Скористаємось підходом Code First, який дає змогу визначити модель сутності в коді, створити базу даних із моделі, а потім додати дані до бази даних.

Виконавши аналіз предметної області, а також оцінивши вимоги, що ставляться до інформаційної web-системи виділимо наступні сутності у проєктованій базі даних: студент (Student), інструктор (Instructor), курс (Course), зарахування на курс (Enrollment), підрозділ (Department), розташування підрозділу (OfficeAppartment).

Спроєктуємо необхідні моделі у вигляді класів, а також подамо їх у вигляді таблиць бази даних (таблиці 2.1-2.5).

| | | | | | | |
|------|------|---------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. 31 |
| Зм.. | Арк. | №докум. | Підпис | Дата | | |

Таблиця 2.2 – Поля таблиці Enrollment

| Поле | Тип | Опис |
|--------------|---------------|---|
| EnrollmentID | int | Ідентифікатор зарахування (первинний ключ) |
| CourseID | int | Зовнішній ключ для зв'язку із сутністю Course |
| StudentID | int | Зовнішній ключ для зв'язку Student |
| Grade | nvarchar(MAX) | Оцінка |

```
public enum Grade
```

```
{
    A, B, C, D, F
}
```

```
public class Enrollment
```

```
{
    public int EnrollmentID { get; set; }
    public int CourseID { get; set; }
    public int StudentID { get; set; }
    public Grade? Grade { get; set; }

    public virtual Course Course { get; set; }
    public virtual Student Student { get; set; }
}
```

Властивість EnrollmentID буде первинним ключем; ця сутність використовує шаблон ідентифікатора імені класу замість окремого ідентифікатора. Властивість Grade є перерахуванням. Дане поле може приймати значення NULL. Властивість StudentID – це зовнішній ключ, а відповідна властивість навігації – Student. Сутність Enrollment пов'язана з однією сутністю

Student, тому властивість може містити лише одну сутність Student (на відміну від властивості навігації Student.Enrollments, як у сутності Student, яка може містити кілька сутностей Enrollment). Властивість CourseID є зовнішнім ключем, а відповідна властивість навігації – Course. Сутність Enrollment пов'язана з однією сутністю Course.

Таблиця 2.3 – Поля таблиці Course

| Поле | Тип | Опис |
|--------------|---------------|---|
| CourseID | int | Ідентифікатор курсу |
| Title | nvarchar(MAX) | Назва курсу |
| Credits | int | Кількість кредитів |
| DepartmentID | int | Зовнішній ключ для зв'язку із сутністю Department |

```
public class Course
{
    [DatabaseGenerated(DatabaseGeneratedOption.None)]
    public int CourseID { get; set; }
    public string Title { get; set; }
    public int Credits { get; set; }

    public virtual ICollection<Enrollment> Enrollments { get; set; }
}
```

Властивість Enrollments є властивістю навігації. Сутність Course може бути пов'язана з будь-якою кількістю сутностей Enrollment.

сутності, оскільки його ім'я не відповідає правилам іменування ідентифікатора ID або класу. Тому атрибут Key використовується для ідентифікації його як первинного ключа у сутності:

```
[Key]
[ForeignKey("Instructor")]
public int InstructorID { get; set; }
```

Слід відзначити, коли між двома сутностями (наприклад, як у відношенні між OfficeAssignment і Instructor) існує зв'язок «один до нуля або один» або «один до одного», Entity Framework не може визначити, який кінець зв'язку є основним, а який кінець залежний. Відносини «один-до-одного» мають властивість навігації посилання в кожному класі до іншого класу. Тому для цього використаємо атрибут ForeignKey.

Таблиця 2.6 – Поля таблиці Department

| Поле | Тип | Опис |
|--------------|---------------|---|
| DepartmentID | int | Ідентифікатор підрозділу |
| Name | nvarchar(MAX) | Назва |
| Budget | float | Вартість |
| StartDate | Datetime | Дата початку |
| InstructorID | Int | Зовнішній ключ для зв'язку із сутністю Instructor |

```
public class Department
{
    public int DepartmentID { get; set; }

    [StringLength(50, MinimumLength = 3)]
    public string Name { get; set; }
```

```

[DataType(DataType.Currency)]
[Column(TypeName = "money")]
public decimal Budget { get; set; }

[DataType(DataType.Date)]
[DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}",
ApplyFormatInEditMode = true)]
[Display(Name = "Start Date")]
public DateTime StartDate { get; set; }

public int? InstructorID { get; set; }

[Timestamp]
public byte[] RowVersion { get; set; }

public virtual Instructor Administrator { get; set; }
public virtual ICollection<Course> Courses { get; set; }
}

```

У сутності Department зовнішній ключ і властивості навігації відображають наступні відносини:

- Відділ може мати або не мати адміністратора, а адміністратор завжди є інструктором. Тому властивість InstructorID включено як зовнішній ключ до сутності Instructor, яка може приймати значення NULL;

- У відділі може бути багато курсів, тому додамо властивість навігації Courses.

Спроектувавши всі необхідні сутності, які описують предметну область, побудуємо концептуальну модель бази даних для інформаційної web-система ІТ-школи.

В проєктованій концептуальній схемі існує зв'язок «багато-до-багатьох» між сутностями «Student» і «Course», а сутність «Enrolment» функціонує як таблиця-об'єднання, через яку реалізується зв'язок «багато-до-багатьох». Це означає, що таблиця Enrollment містить додаткові дані, крім зовнішніх ключів для об'єднаних таблиць (у нашому випадку первинний ключ і властивість Grade). На рис.2.2 зображено, ці зв'язки на діаграмі сутності.

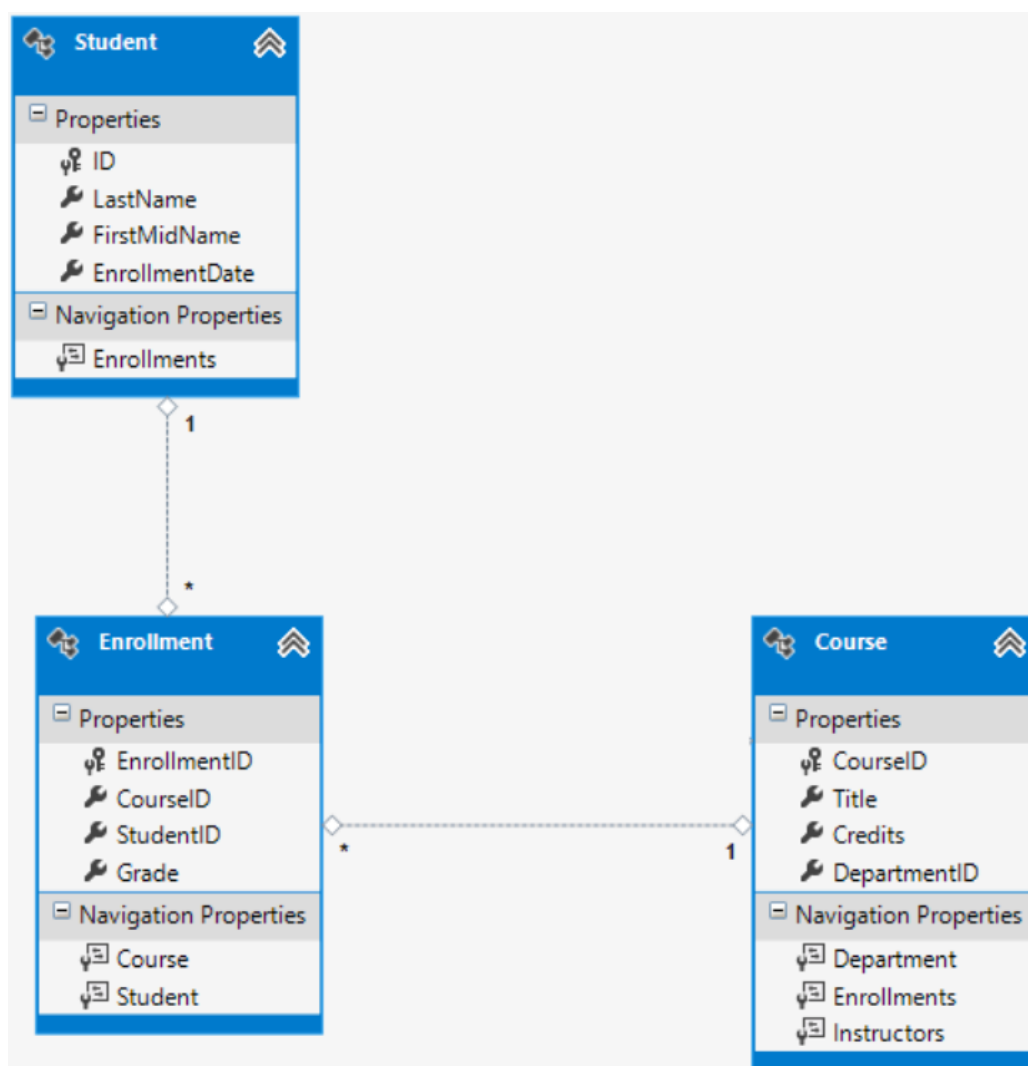


Рисунок 2.2 – Відношення між сутностями Student-Enrolment-Course

Слід відзначити, що для створення концептуальної схеми використано пакет Entity Framework Power Tools. Кожна лінія зв'язку має 1 на одному кінці та зірочку (*) на іншому, що вказує на зв'язок «один до багатьох».

Якби таблиця Enrollment не включала інформацію про оцінку, вона мала б містити лише два зовнішні ключі CourseID і StudentID. У цьому випадку це відповідатиме таблиці з'єднання «багато-до-багатьох» без корисного навантаження (або чистій таблиці з'єднання) у базі даних, і тоді взагалі не потрібно було б створювати для неї клас моделі. Сутності Instructor і Course мають такий зв'язок «багато-до-багатьох», проте між ними не має проміжної сутності (рис 2.3).

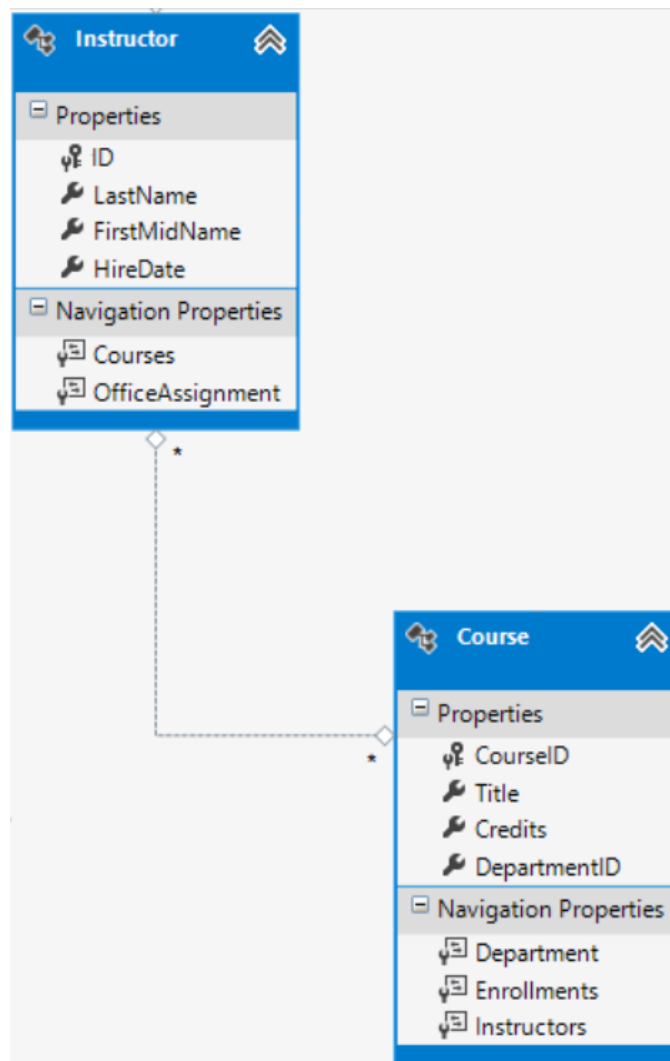


Рисунок 2.3 – Відношення між сутностями Instructor та Course

Однак відношення «багато-до-багатьох» вимагає таблиці об'єднання. Entity Framework автоматично створить таблицю CourseInstructor, і робота з нею здійснюватиметься опосередковано, читаючи та оновлюючи поля навігації Instructor.Courses і Course.Instructors (рис. 2.4).

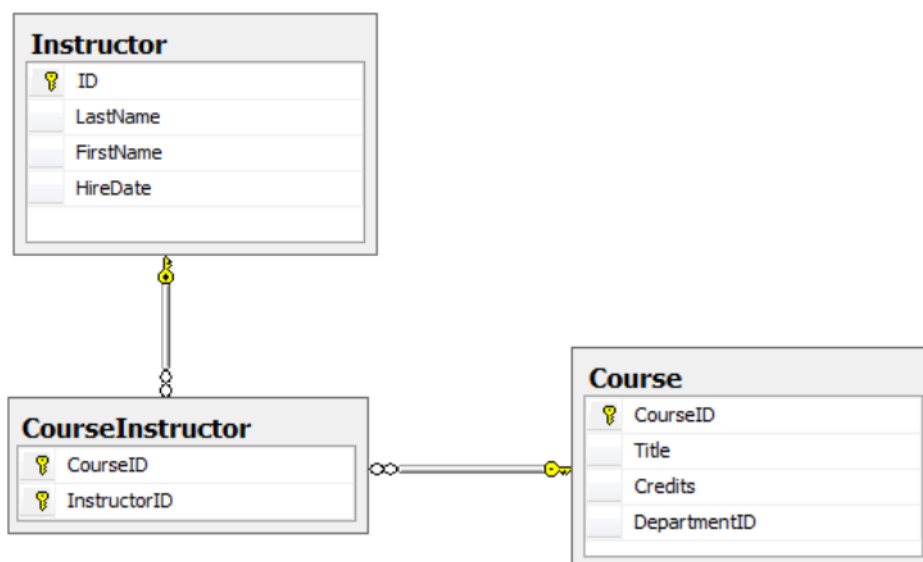


Рисунок 2.4 – Автоматично згенерована таблиця-об'єднання CourseInstructor

Таким чином результуюча концептуальна схема бази даних інформаційної web-системи для ІТ-школи буде виглядати як зображено на рис. 2.5. В результаті спроектована концептуальна модель бази даних дозволить чітко визначити сутності, атрибути та зв'язки між ними в предметній галузі. Це дозволить спростити розуміння того, як дані зберігатимуться і як вони будуть використовуватись в інформаційній системі. Окрім того спроектована модель бази даних ілюструє ключові елементи бази даних та їх зв'язки, що дозволить виявити помилки та протиріччя у предметній області.

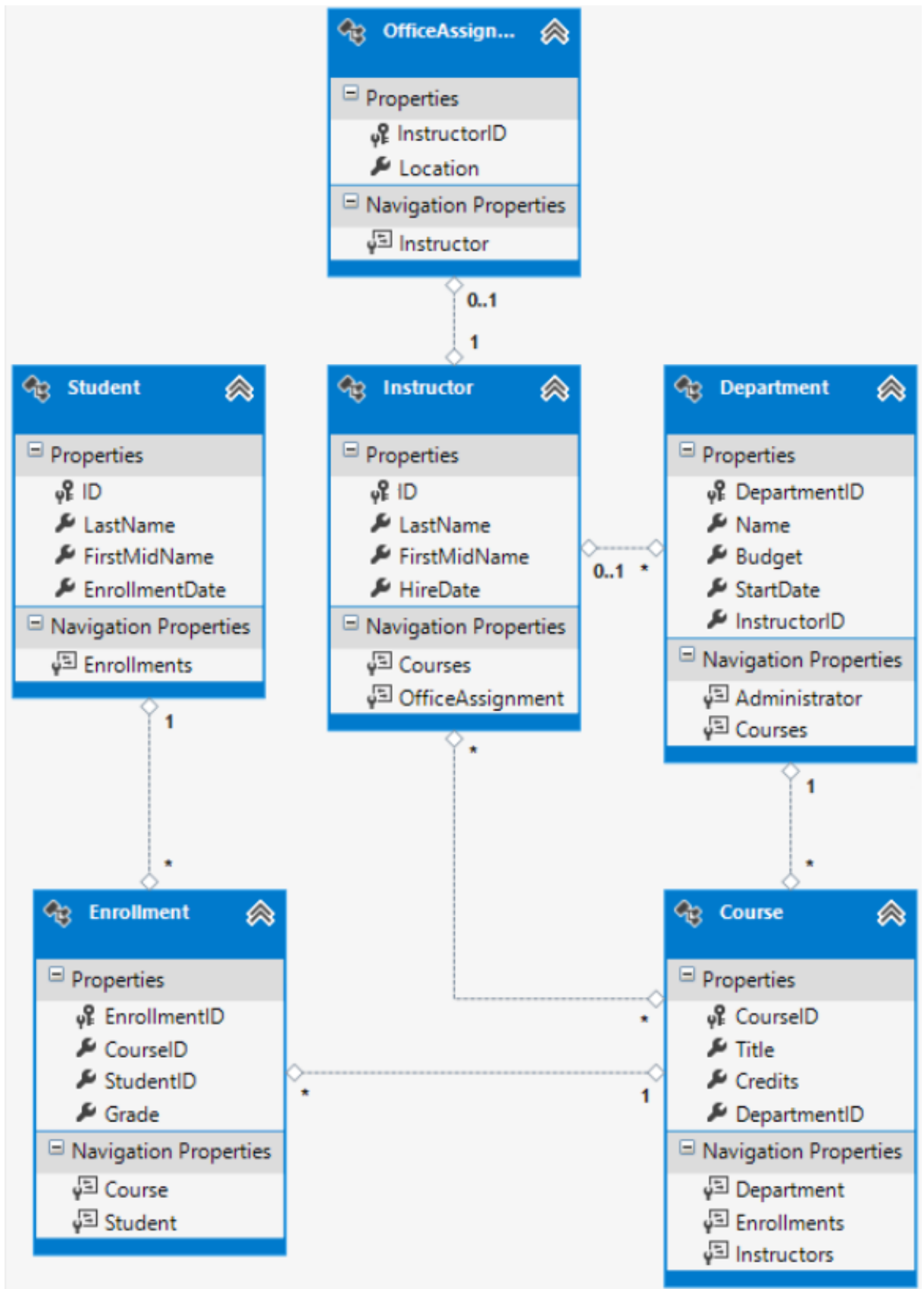


Рисунок 2.5 – Концептуальна схема бази даних для інформаційної web-система ІТ-школи

Висновки до розділу 2

Таким чином проведено огляд технологій та засобів реалізації для проектування інформаційної web-система ІТ-школи. При проектуванні інформаційної web-система для ІТ-школи було використано наступні засоби та технології:

- Технологія MVC.
- Технологія Entity Framework.
- ІIS Express.
- Середовище розробки Visual Studio 2019.

Всі обрані засоби й технології є актуальними на даний час та відкритими, що означає можливість безкоштовного їх використання, навіть для розробки закритих комерційних проектів. Обраний стек технологій дозволить швидко та якісно розробити прототип інформаційної web-система для ІТ-школи. Здійснено проектування сутностей бази даних та запропоновано концептуальну модель бази даних інформаційної web-система для ІТ-школи. Визначено набір функційних та нефункційних вимог до проектованої інформаційної web-система для ІТ-школи, що дозволило означити, функції та характеристики які має мати система, щоб задовольнити потреби кінцевих користувачів.

| | | | | | | |
|-----|------|---------|--------|------|----------------------------|------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. |
| | | | | | | 43 |
| Зм. | Арк. | №докум. | Підпис | Дата | | |

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ WEB-СИСТЕМИ ДЛЯ ІТ-ШКОЛИ

3.1 Структура інформаційної web-системи для ІТ-школи

Інформаційна web-система (або веб-система) є комплексним програмним забезпеченням, яке розробляється для забезпечення функціонування веб-сайту. Подамо структуру інформаційної web-системи для ІТ школи у вигляді наступних компонентів:

– Клієнтська частина (frontend): включає в себе веб-сторінки, які відображаються у браузері користувача. Клієнтська частина відповідає за збір та відображення інформації для користувача.

– Серверна частина (backend): це програмне забезпечення, яке запускається на сервері та відповідає за збереження, обробку та передачу даних між сервером та клієнтом. Серверна частина включає в себе базу даних, веб-сервер.

– База даних: включає в себе набір даних, які зберігаються на сервері та використовуються для забезпечення роботи системи. Основними функціями бази даних в проєктованій інформаційній web-системі для ІТ школи є:

Збереження даних: у проєктованій базі даних зберігається інформація про студентів, викладачів, наявні курси, місце розташування навчальних приміщень.

Організація даних: у проєктованій базі даних всі дані організовані у логічні групи та поєднанні між собою відношеннями, що дозволило спростити їх зберігання та обробку.

Забезпечення доступу до даних: база даних дозволяє робити запити до даних та отримувати необхідну інформацію. Запити до бази даних передбачають створення, вибір, оновлення, видалення та вставку даних про студентів, інструкторів та доступні курси ІТ школи.

Забезпечення цілісності даних: база даних забезпечує цілісність даних шляхом застосування правил та обмежень на дані, які зберігаються в інформаційній системі.

| | | | | | | |
|-----|------|---------|--------|------|----------------------------|------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. |
| | | | | | | 44 |
| Зм. | Арк. | №докум. | Підпис | Дата | | |

– API (Application Programming Interface): інтерфейс програмування додатків, що дозволяє забезпечити взаємодію між клієнтом та сервером. API надає можливість доступу до даних та функцій, що використовуються в інформаційній системі.

Узагальнену структуру інформаційної web-системи для ІТ-школи представлено на рис. 3.1 Вхідними даними для проектованої інформаційної web-системи є набір даних, введених студентами та/або інструкторами. Вихідними даними ІС є інформація для внутрішнього та зовнішнього використання, що представлена у вигляді звітів.

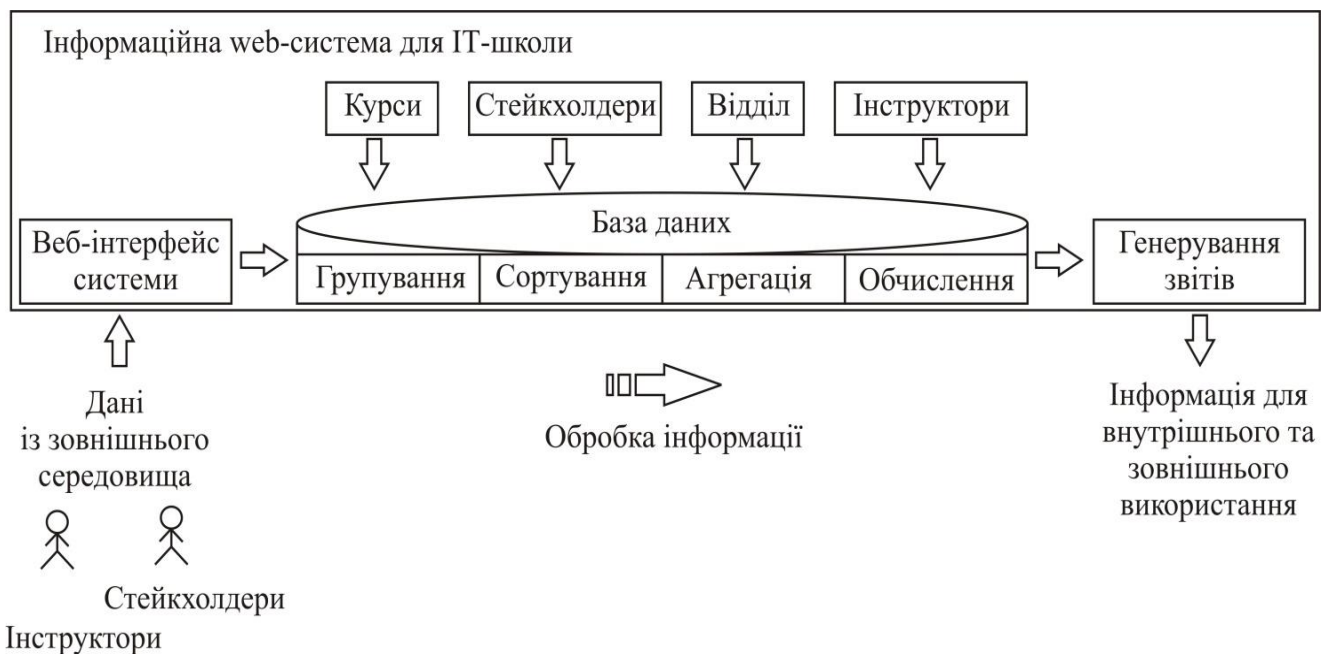


Рисунок 3.1 – Узагальнена структура інформаційної web-системи для ІТ-школи

Після опису узагальненої структури інформаційної web-системи подамо високорівневе представлення ІС у вигляді діаграми системного контексту (context flow diagram), за допомогою якої можна визначити межу між системою або частиною системи та її середовищем, показуючи об'єкти, які з нею взаємодіють.

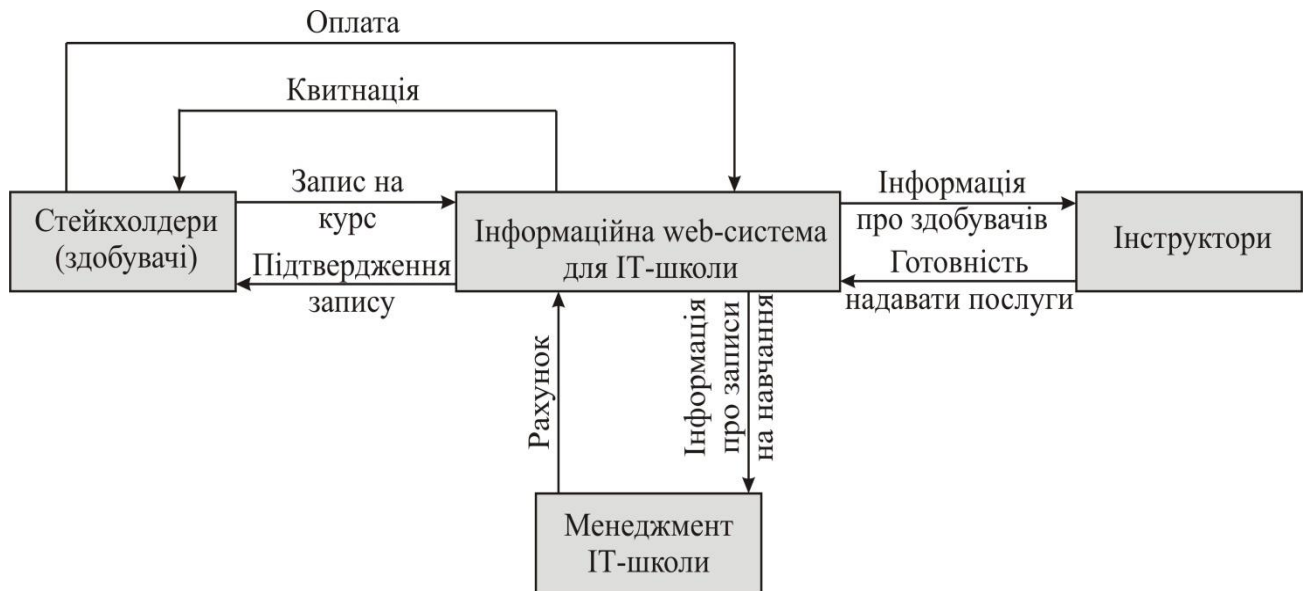


Рисунок 3.2 – Діаграма системного контексту інформаційної web-системи

3.2 Реалізація прототипу інформаційної web-системи для ІТ-школи

Реалізація прототипу інформаційної web-системи для ІТ-школи включала в себе створення програмного забезпечення web-сайту та бази даних для збереження інформації. В результаті реалізації прототипу інформаційної web-системи володіє наступними функціями:

- Збереження даних про студентів (додавання, видалення, редагування, перегляд) та формування звіту.
- Збереження даних про інструкторів (додавання, видалення, редагування, перегляд) та формування звіту.
- Створення нового курсу, видалення, редагування метаданих та формування звіту.
- Створення, видалення, редагування метаданих підрозділу та формування звіту.
- Перегляд інформації про систему.

Інтерфейсне вікно головної сторінки зображено на рис. 3.3. Створена інформаційна web-систем володіє мінімалістичним дизайном, що дозволяє:

– Легкість сприйняття: дизайн спроектованої інформаційної web-системи простий і легко сприймається користувачами, оскільки не завалює їх зайвими деталями. Це дозволяє користувачам швидко зорієнтуватися на сайті і швидше сформулювати необхідний звіт.

– Швидкість завантаження: система не обтяжена графічними елементами, що дозволяє сторінці завантажуватися швидше. Швидкість завантаження є важливою для збереження користувачів на сайті і забезпечення їх задоволеності від користування інформаційною системою.

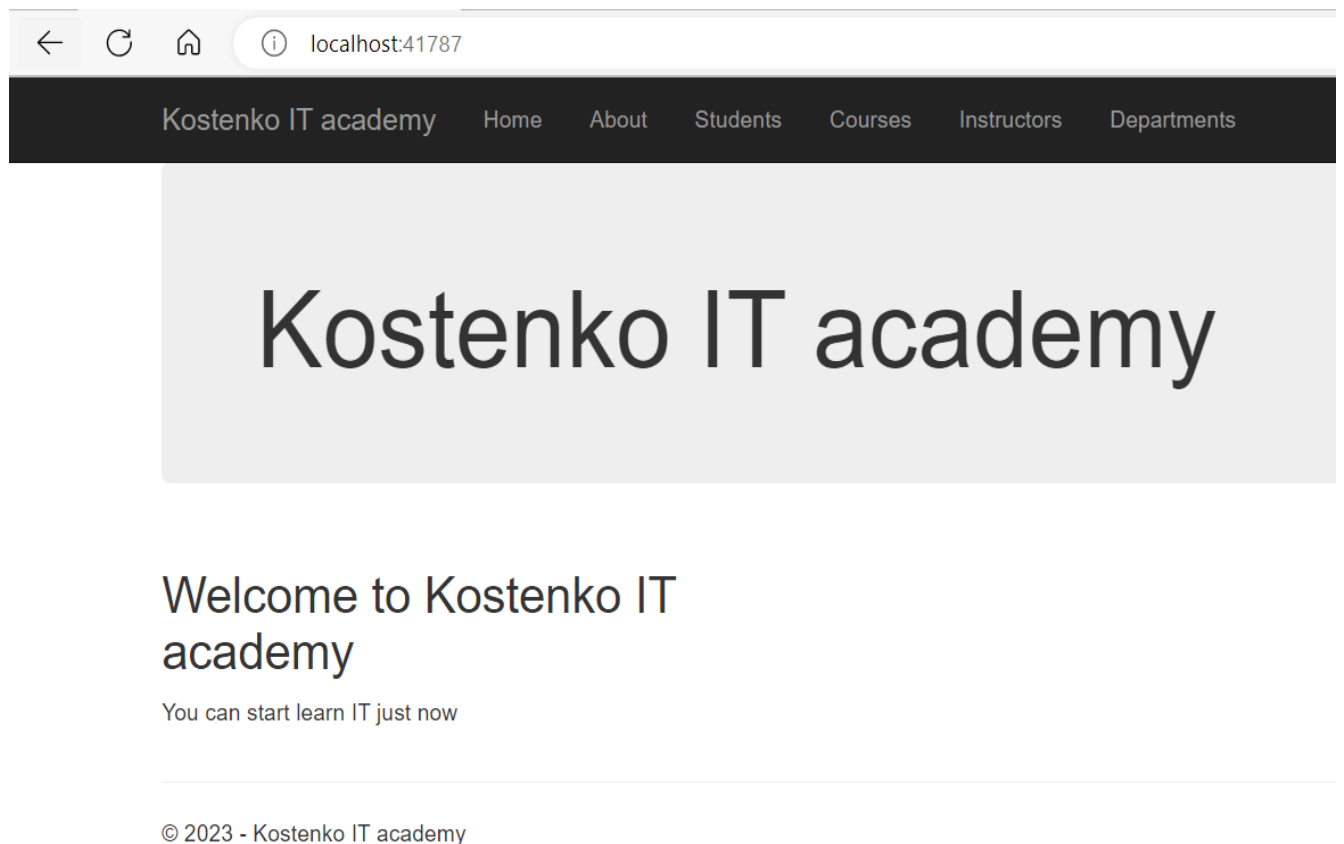


Рисунок 3.3 – Інтерфейсне вікно головної сторінки

– Висока читабельність: спроектована система володіє високою читабельністю тексту, що дозволяє користувачам зосередитися на контенті.

– Сучасний вигляд: інтерфейс спроектованої системи має прості кольори, що дозволяє системі виглядати сучасно і стильно. Таким чином система відповідає сучасним трендам та забезпечує його актуальність.

про інструктора, видалення інструктора, а також формування звіту для вибраного інструктора.

← ↻ 🏠 ⓘ localhost:41787/Instructor/Create

Kostenko IT academy Home About Students Courses Instructors Departments

Create

Instructor

Last Name

First Name

Hire Date

Office Location

1 Programming 2 Web

[Back to List](#)

© 2023 - Kostenko IT academy

Рисунок 3.5 – Інтерфейсне вікно створення нового інструктора

← ↻ 🏠 ⓘ localhost:41787/Instructor

Kostenko IT academy Home About Students Courses Instructors Departments

Instructors

[Create New](#)

| Last Name | First Name | Hire Date | Office | Courses | |
|-----------|------------|------------|-------------|---------------|--|
| Ivanov | Petro | 2023-02-15 | Lviv | 1 Programming | Select Edit Details Delete |
| Sidorenko | Ivan | 2023-03-07 | Khmelnyskyi | 2 Web | Select Edit Details Delete |

© 2023 - Kostenko IT academy

Рисунок 3.6 – Інтерфейсне вікно відображення звіту про доступних інструкторів ІТ школи

При натисканні у вікні Instructor на посилання Select буде сформовано звіт, що відображає курси, що закріплені за цим інструктором (рис. 3.7).

The screenshot shows a web browser window with the URL localhost:41787/Instructor/Index/3. The page title is "Instructors" and it includes a navigation menu with "Kostenko IT academy", "Home", "About", "Students", "Courses", "Instructors", and "Departments". Below the title is a "Create New" link. A table lists two instructors: Ivanov Petro (Hire Date: 2023-02-15, Office: Lviv, Courses: 1 Programming) and Sidorenko Ivan (Hire Date: 2023-03-07, Office: Khmelnytskyi, Courses: 2 Web). Each row has "Select | Edit | Details | Delete" links. Below this is a section "Courses Taught by Selected Instructor" with a table showing one course: Number 1, Title Programming, Department Temp. The footer contains "© 2023 - Kostenko IT academy".

Рисунок 3.7 – Інтерфейсне вікно відображення звіту про курси, які викладає інструктор

Якщо потрібно сформуванати звіт, що відображає записаних студентів на даний курс, слід нижче натиснути посилання Select (рис. 3.8).

The screenshot shows a web browser window with the URL localhost:41787/Instructor/Index/3?courseID=1. The page title is "Instructors" and it includes a navigation menu with "Kostenko IT academy", "Home", "About", "Students", "Courses", "Instructors", and "Departments". Below the title is a "Create New" link. A table lists two instructors: Ivanov Petro (Hire Date: 2023-02-15, Office: Lviv, Courses: 1 Programming) and Sidorenko Ivan (Hire Date: 2023-03-07, Office: Khmelnytskyi, Courses: 2 Web). Each row has "Select | Edit | Details | Delete" links. Below this is a section "Courses Taught by Selected Instructor" with a table showing one course: Number 1, Title Programming, Department Temp. Below that is a section "Students Enrolled in Selected Course" with a table with columns "Name" and "Grade". The footer contains "© 2023 - Kostenko IT academy".

Рисунок 3.8 – Інтерфейсне вікно відображення звіту про курси, які викладає інструктор та студентів, що записані на курс

Для відображення звіту про наявних студентів слід натиснути у меню на Student. Сформований звіт містить інформація про студентів, яка включає прізвище, ім'я та дата зарахування. Для кожного студента є можливість редагування та перегляду метаданих, а також відрахування студентів (видалення). На рис. 3.9 представлено інтерфейсне вікно відображення звіту про студентів. Є можливість відображення даних про студента (рис. 3.10).

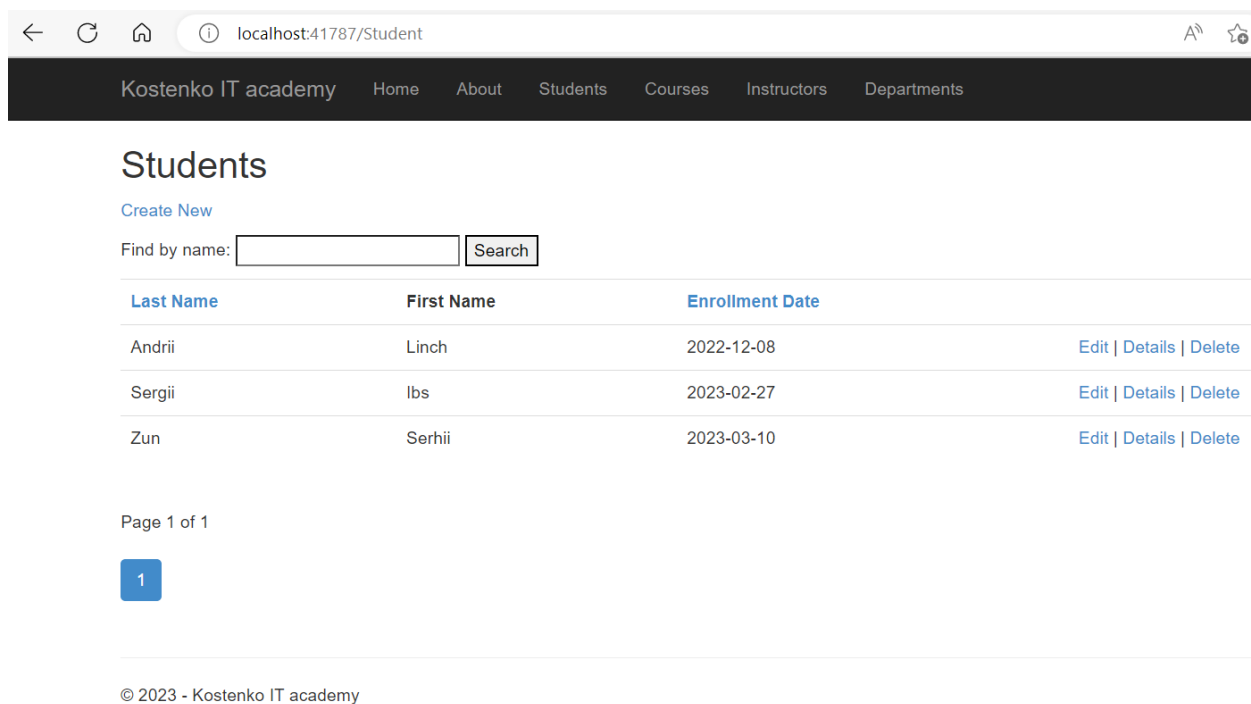


Рисунок 3.9 – Інтерфейсне вікно відображення звіту про студентів

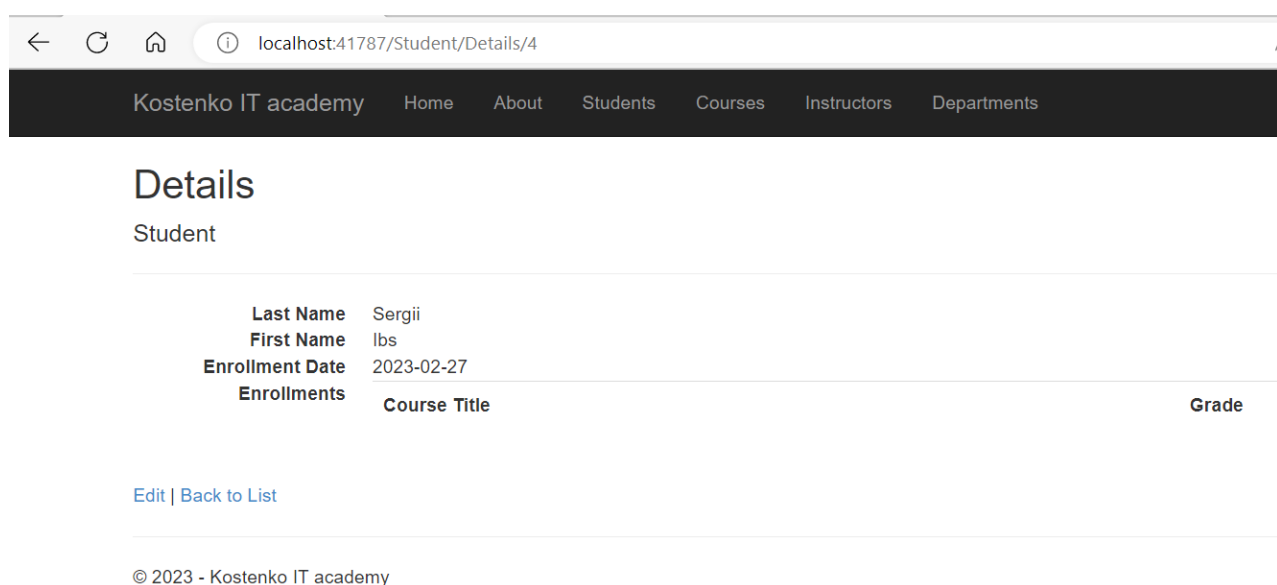


Рисунок 3.10 – Інтерфейсне вікно відображення метаданих про студента

Для перегляду інформації про доступні курси та формування відповідного звіту передбачено вікно Courses. Для кожного курсу передбачено функції перегляду та режагування метаданих про курс, а також видалення курсу. Передбачена опція пошуку студента за введеним ім'я або прізвищем. Є можливість відображення всіх студентів (рис. 3.11, 3.12).

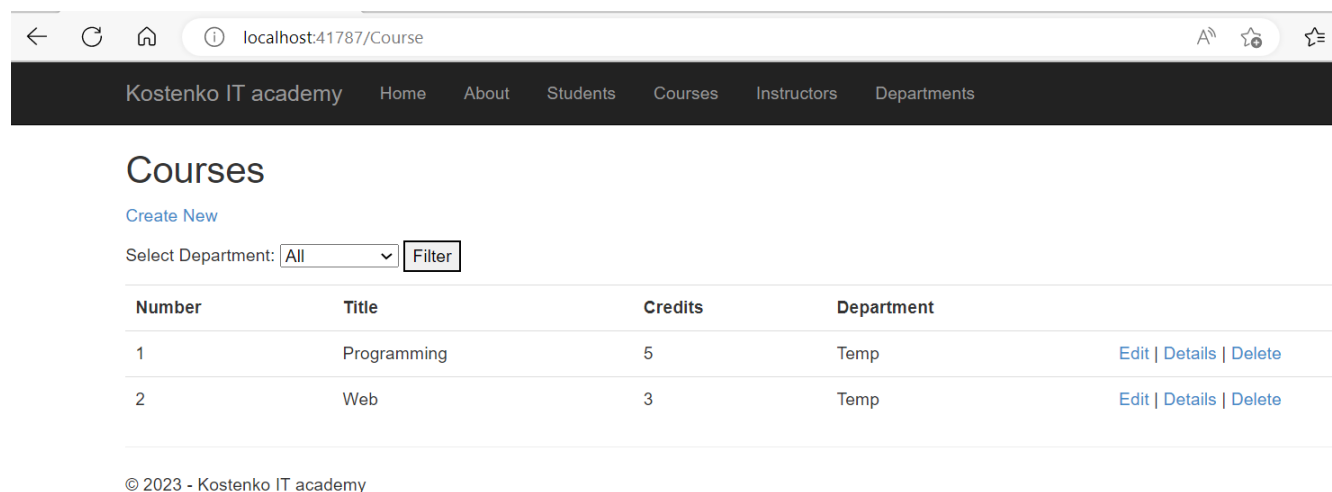


Рисунок 3.11 – Інтерфейсне вікно відображення звіту про доступні курси

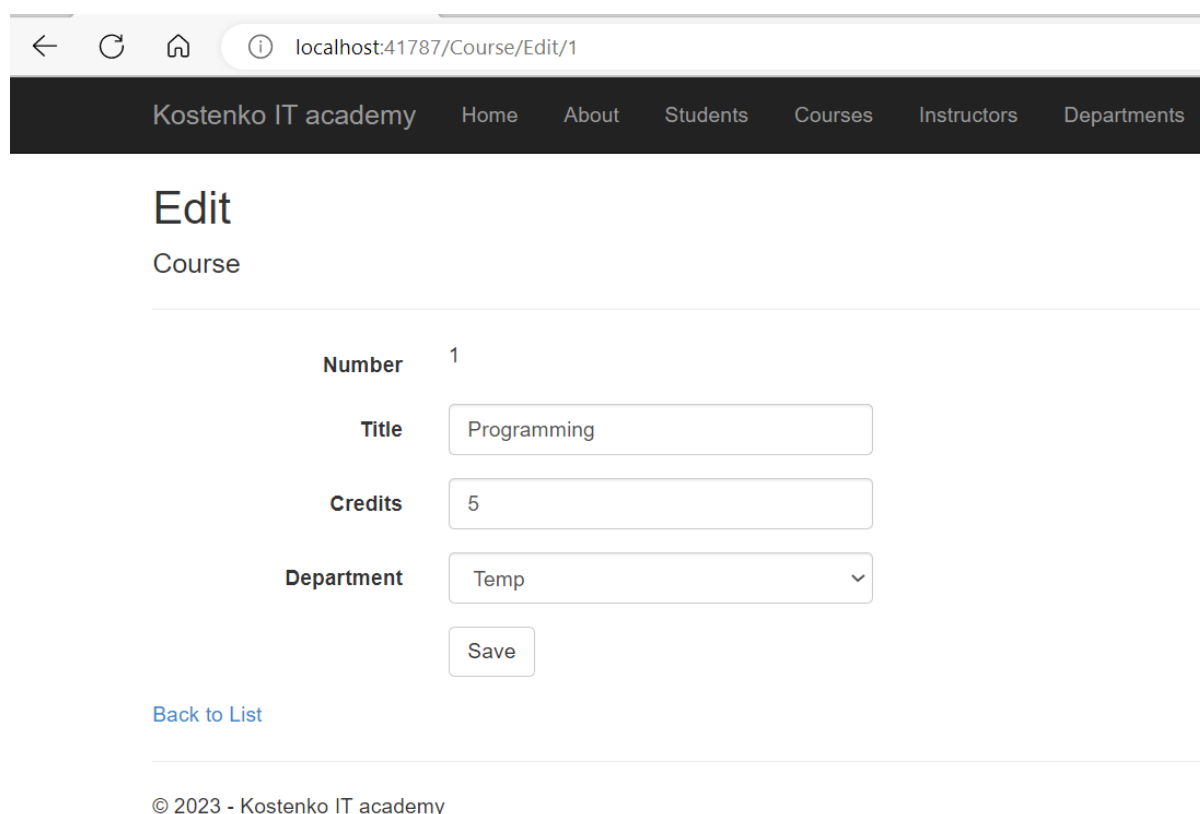


Рисунок 3.12 – Інтерфейсне вікно редагування метаданих курсу

Для відображення інформації та формування звіту про наявні підрозділи передбачено опцію Departments. Для кожного підрозділу є можливість редагування та перегляд метаданих про підрозділ, а також видалення підрозділу. На рис. 3.13 представлено інтерфейсне вікно відображення звіту про доступні офіси.

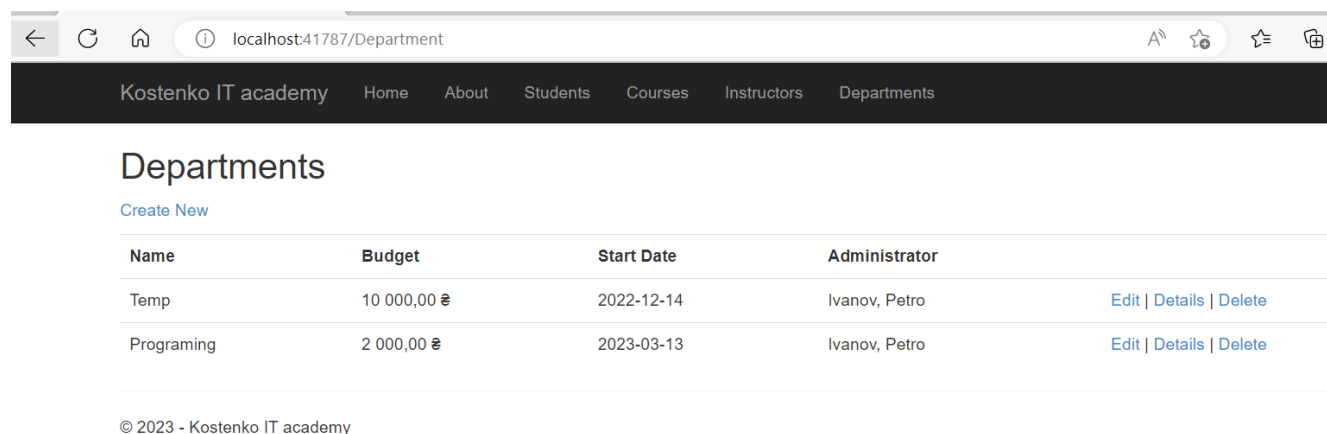


Рисунок 3.13 – Інтерфейсне вікно відображення звіту про доступні офіси

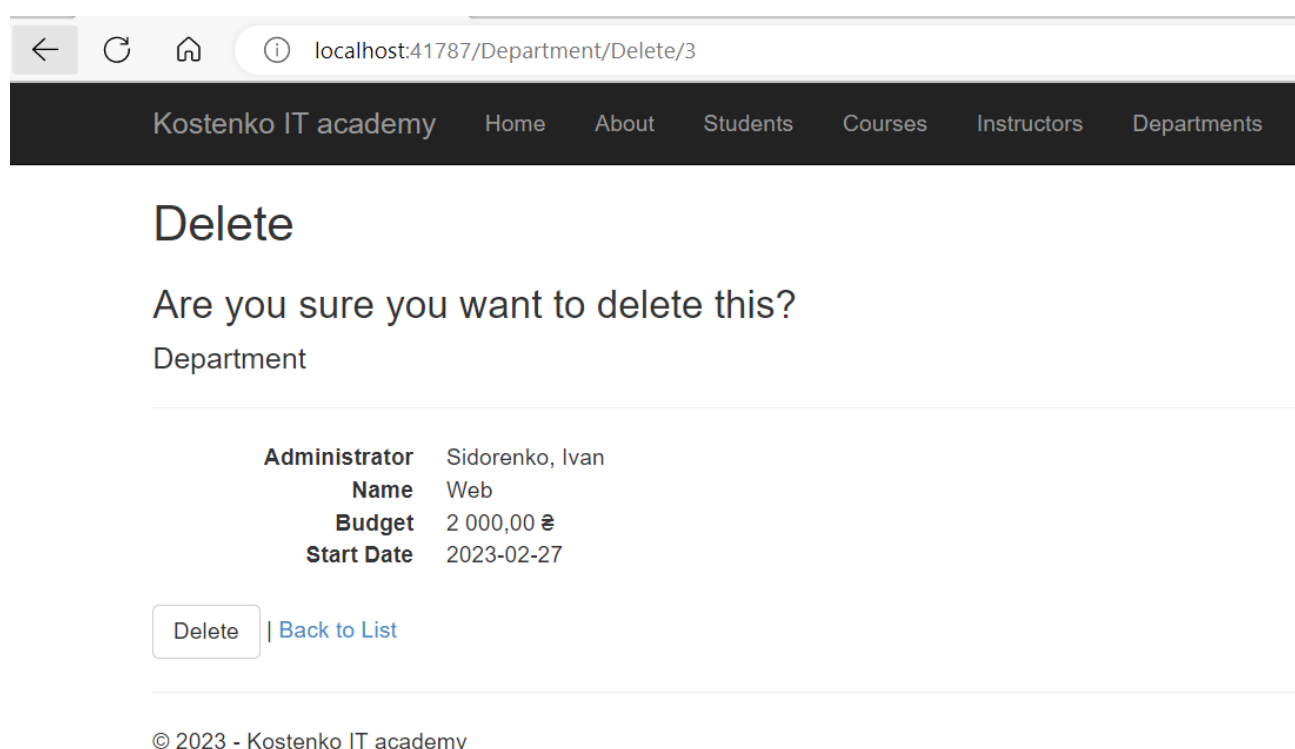


Рисунок 3.14 – Інтерфейсне вікно видалення відділу

Таким чином було реалізовано прототип інформаційної web-системи для ІТ школи у вигляді web застосунку. Розроблена система володіє наступним функціоналом: збереження даних про студентів (додавання, видалення, редагування, перегляд метаданих) та формування звіту; збереження даних про інструкторів (додавання, видалення, редагування, перегляд) та формування звіту; створення нового курсу, видалення, редагування метаданих та формування відповідного звіту; створення, видалення, редагування метаданих підрозділу та формування звіту; перегляд інформації про інформаційну систему. Окрім того можна відзначити досить зручний інтерфейс інформаційної системи, що не переобтяжений зайвим контентом. Проаналізувавши переваги розробленої інформаційної web-системи також можна сформулювати її недоліки, що полягають головним чином у відсутності інструментів для моніторингу прогресу студентів з того чи іншого курсу та відсутність можливості обміну повідомлення між користувачами у системі.

3.3 UML діаграм для візуалізація роботи інформаційної системи

Для візуалізації роботи запропонованої інформаційної системи скористаємось UML (Unified Modeling Language), тобто уніфікованою мовою моделювання, що використовується розробниками програмного забезпечення для візуалізації процесів та роботи систем. Використання UML-діаграм для візуалізації структури розробленої інформаційної системи продиктовано в першу чергу властивістю візуалізації, що проявляється у відображенні структури та поведінки системи графічним способом, що дозволяє представити систему в цілому та легко зрозуміти взаємозв'язки між складовими компонентами системи.

Представимо множину дії адміністратора інформаційної web-системи для ІТ школи у вигляді UML діаграми варіантів використання, що представлена на рис. 3.15.

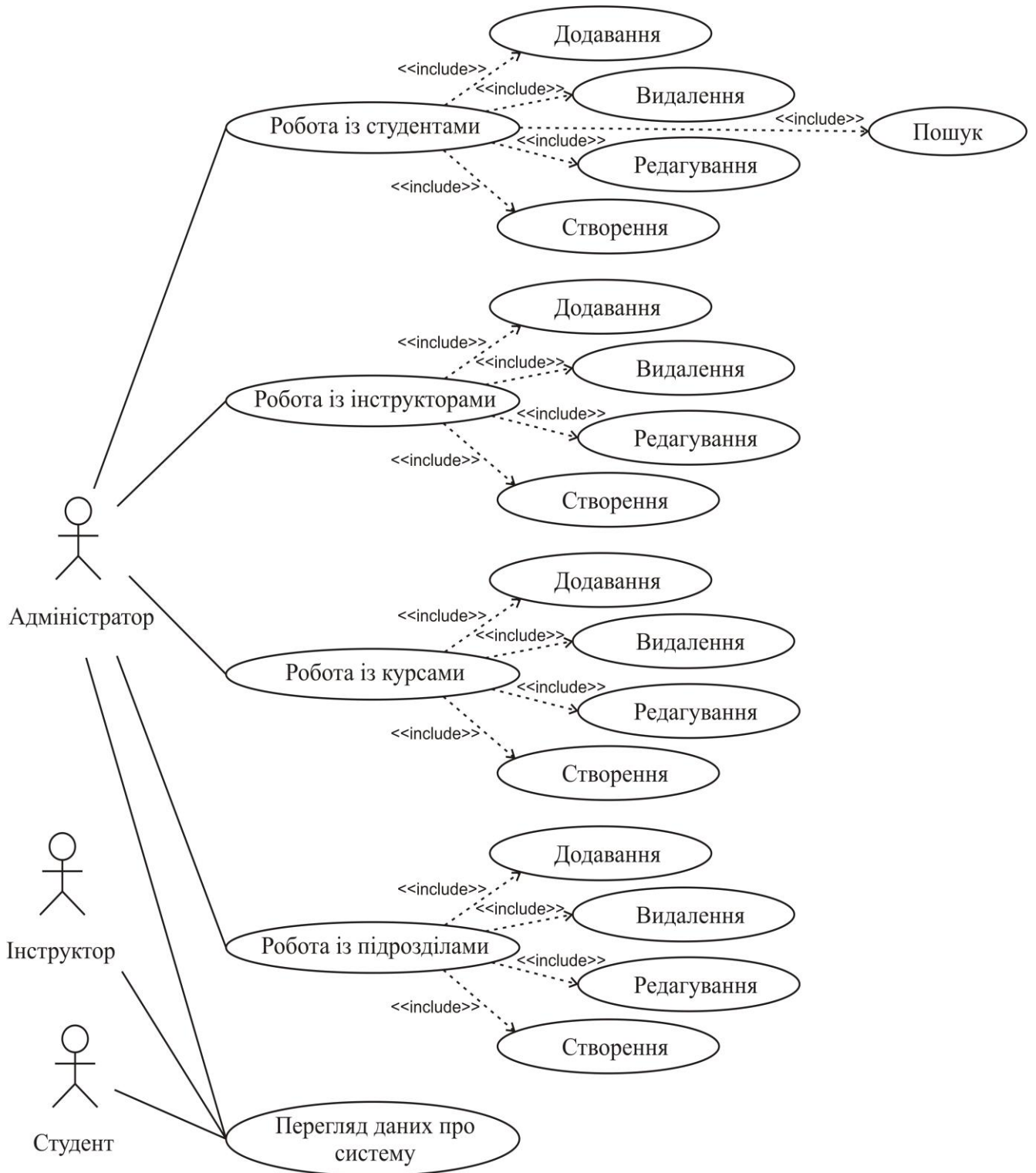


Рисунок 3.15 – UML-діаграма варіантів використання

Представимо порядок роботи із інформаційною системою у вигляді діаграми послідовності (3.16). Діаграма послідовності (Sequence Diagram) –

показує часові особливості передачі і прийому повідомлень об'єктами при роботі із інформаційною системою.

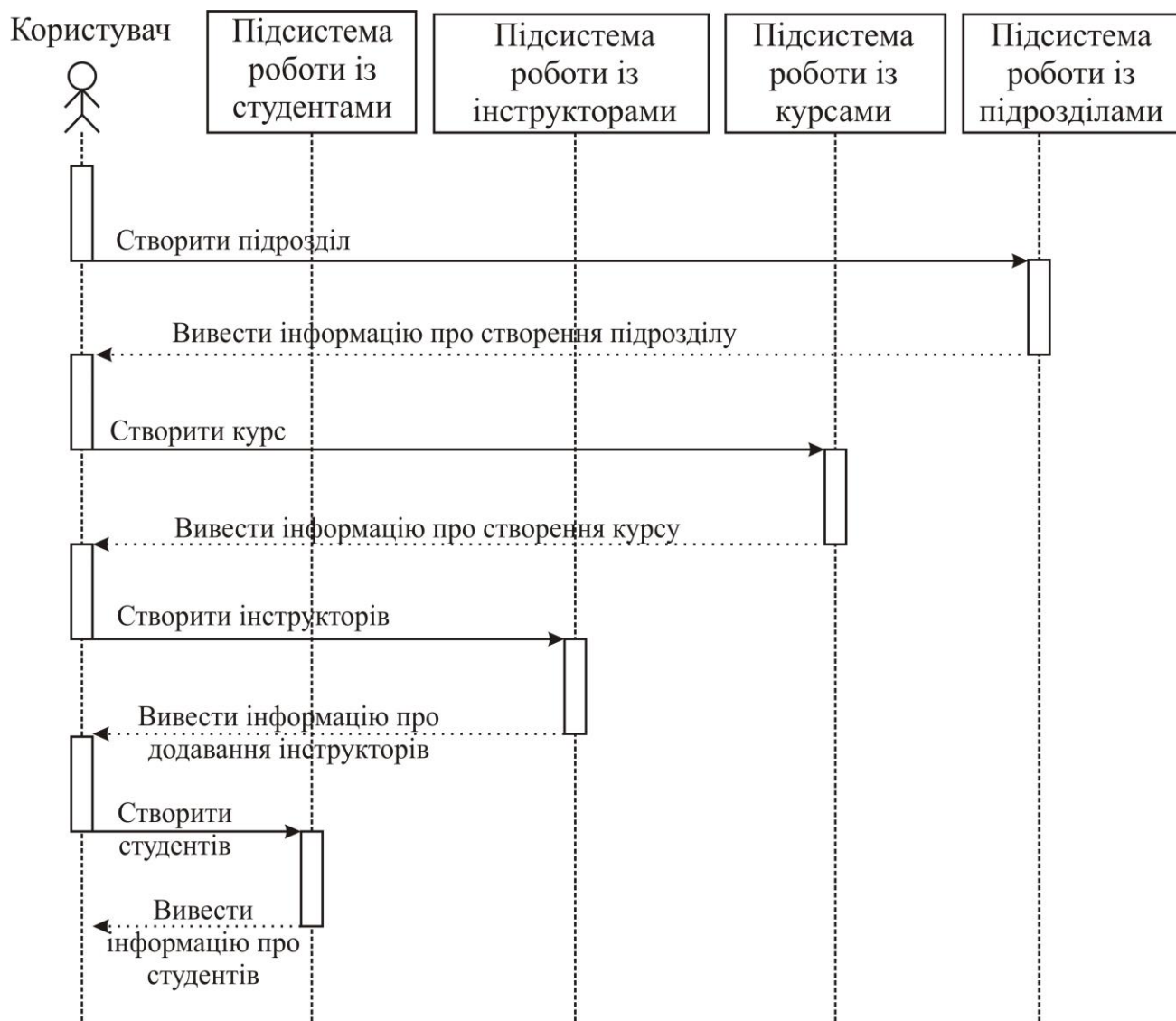


Рисунок 3.16 – UML-діаграма послідовності виконання

Для початку роботи із системою адміністратор здійснює створення підрозділу ІТ школи, у якому будуть створюватись відповідні курси. Прикладами підрозділів можуть бути: розробка програмного забезпечення, методи штучного інтелекту, аналітика, великі дані, devops, тощо. Після створення підрозділу наступним кроком слід створити відповідний курс. Після цих дій здійснюється наповнення системи інформацією про інструкторів та студентів. Всі інструктори та студенти зв'язуються із відповідним курсом. На всіх стадіях роботи із

системою є можливість перегляду інформації про інформаційну систему. UML-діаграма класів зображена на рис. 3.16

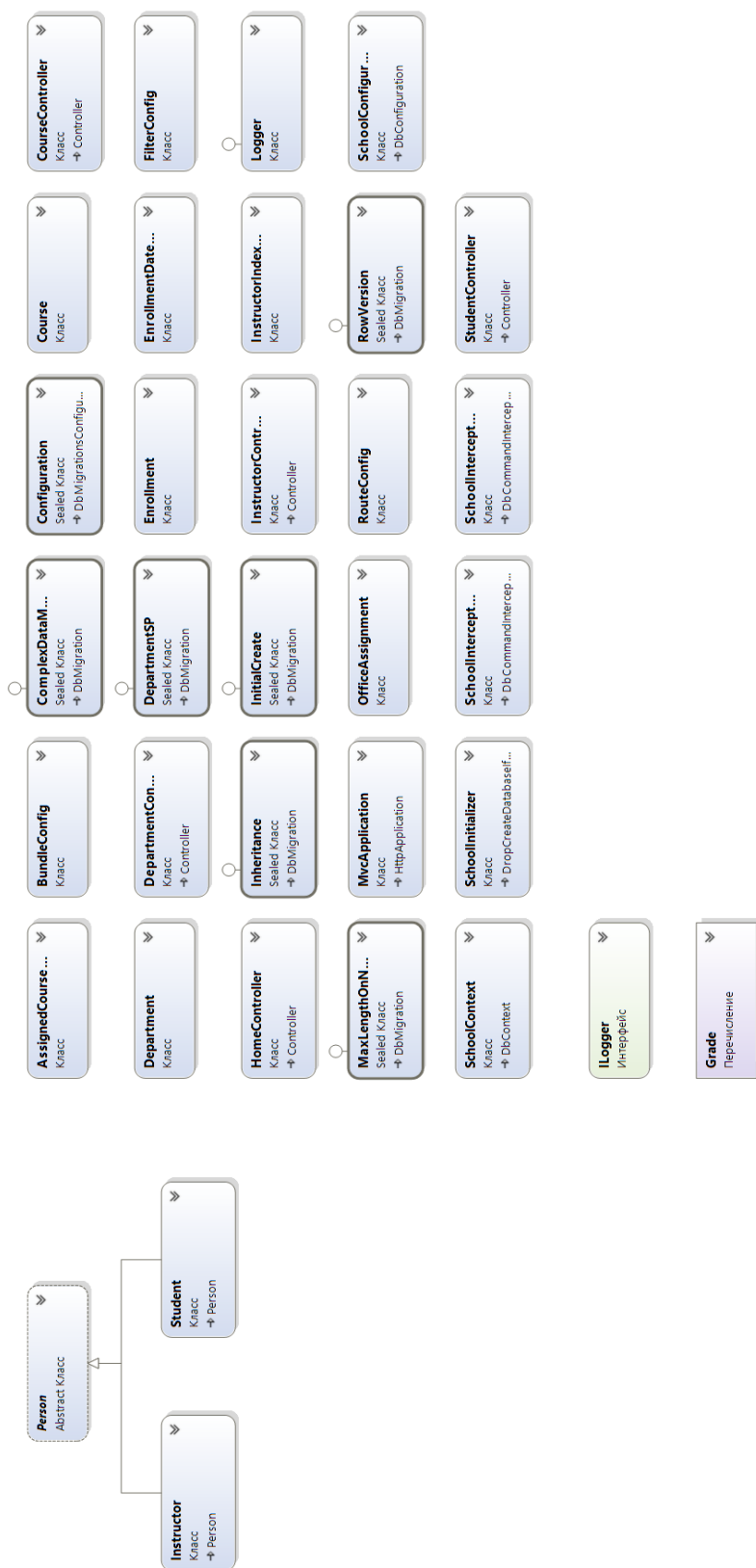


Рисунок 3.17 – UML-діаграма класів

З метою представлення структури запропонованої інформаційної системи подамо її у вигляді UML-діаграма компонентів (рис. 3.17). Відповідно до цієї діаграми запропонована інформаційна система складається із пристрою користувача, що представлений веб браузером, веб-сервера, та сервера бази даних.

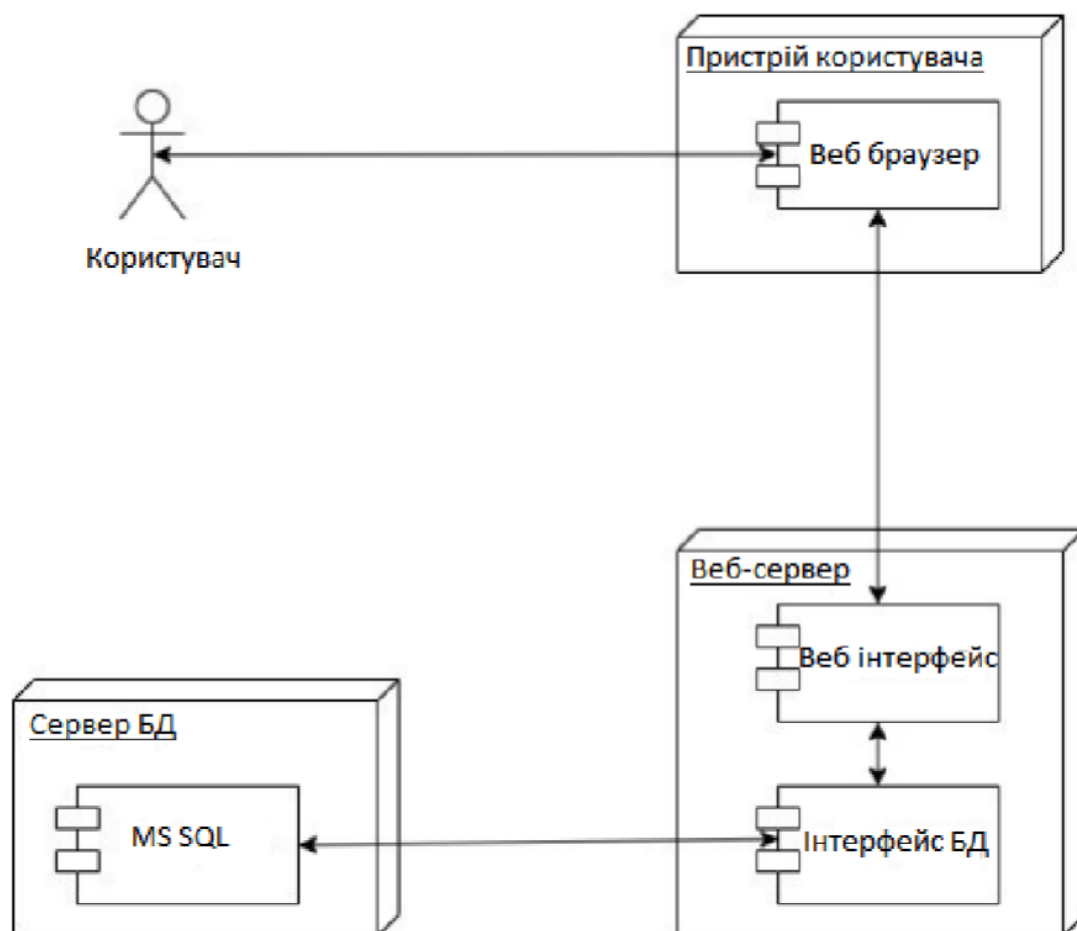


Рисунок 3.18 – UML-діаграма компонентів

Висновки за розділом 3

Таким чином розглянуто структуру запропонованої інформаційної web-системи для ІТ-школи. Реалізовано прототип інформаційної web-системи для ІТ школи, функціонування якого дозволило виконувати такі операції як збереження даних про студентів, інструкторів, створення нового курсу, перегляд інформації про інформаційну систему.

| | | | | |
|-----|------|---------|--------|------|
| Зм. | Арк. | №докум. | Підпис | Дата |
| | | | | |

ВИСНОВКИ

На сьогодні існує багато різних типів освіти, які можна класифікувати за різними критеріями, але основні із них це формальна та неформальна освіта. Формальна освіта реалізується у формальних інституціях, таких як школи, коледжі та університети, та зазвичай включає систему оцінювання та видачу сертифікатів або дипломів. Неформальна освіта, у свою чергу, відбувається поза інституціями формальної освіти та може бути організована різними організаціями. Один з популярних видів неформальної освіти – це ІТ курси, які можуть бути корисні для людей із різних сфер життя. Наявність інформаційної web-системи є важливою складовою організації ІТ курсів, де виконуються функції по обробці та збереженню інформації, необхідної для задоволення потреб користувачів (стейкхолдерів).

В результаті виконання кваліфікаційної роботи було спроектовано та реалізовано інформаційну web-систему для ІТ школи.

В першому розділі розглянуто поняття формальної та неформальної освіти. Проведено огляд принципів функціонування інформаційних web-систем для забезпечення навчальної діяльності, зокрема були розглянуті інформаційні системи Хмельницького національного університету та Київського політехнічного інституту імені Ігоря Сікорського. Також було розглянуті відомі платформи ІТ шкіл. Здійснено огляд їх функціонування та визначено їх недоліки.

В другому розділі проведено огляд технологій та засобів реалізації для проектування інформаційної web-система ІТ-школи. Всі обрані засоби й технології є актуальними на даний час та відкритими, що означає можливість безкоштовного їх використання, навіть для розробки закритих комерційних проектів. Обраний стек технологій дозволив швидко та якісно розробити прототип інформаційної web-система для ІТ-школи. Здійснено проектування сутностей бази даних та запропоновано концептуальну модель бази даних інформаційної web-система для ІТ-школи. Визначено набір функційних та нефункційних вимог до проекрованої інформаційної web-система ІТ-школи, що

| | | | | | | |
|-----|------|---------|--------|------|----------------------------|------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. |
| | | | | | | 59 |
| Зм. | Арк. | №докум. | Підпис | Дата | | |

дозволило означити, функції та характеристики які має мати система, щоб задовольнити потреби кінцевих користувачів.

В третьому розділі розглянуто структуру запропонованої інформаційної web-системи для ІТ-школи. Вхідними даними для зпроектованої інформаційної web-системи є набір даних, введених студентами та/або інструкторами. Вихідними даними інформаційної системи є інформація для внутрішнього та зовнішнього використання, що представлена у вигляді звітів. Реалізовано прототип інформаційної web-системи для ІТ школи, функціонування якого дозволило виконувати такі операції як збереження даних про студентів (додавання, видалення, редагування, перегляд метаданих) та формування звіту; збереження даних про інструкторів (додавання, видалення, редагування, перегляд) та формування звіту; створення нового курсу, видалення, редагування метаданих та формування відповідного звіту; створення, видалення, редагування метаданих підрозділу та формування звіту; перегляд інформації про інформаційну систему. Окрім функціональних особливостей зпроектована інформаційна система характеризується мінімалістичним дизайном, що дозволяє користувачам швидко зорієнтуватися у системі й швидше сформувавши необхідний звіт.

| | | | | | | |
|-----|------|---------|--------|------|----------------------------|------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. |
| | | | | | | 60 |
| Зм. | Арк. | №докум. | Підпис | Дата | | |

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Якість та тестування інформаційних систем. Навчальний посібник для самостійної роботи студентів вищих навчальних закладів. Київ: ННІТ ДУТ, 2020. 128 с.
2. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська. – Київ : КПІ ім. Ігоря Сікорського, 2020. 192с.
3. Hassanien A. E., Elhoseny M. Cybersecurity and Secure Information Systems. Challenges and Solutions in Smart Environments, *Springer Cham*, 2019. p. 314.
4. Єсін В.І. Безпека інформаційних систем і технологій : навчальний посібник, Х.: ХНУ імені В. Н. Каразіна, 2013. 632 с.
5. Chapple M. CISSP Certified Information Systems Security Professional, *Sybex*, 2018. 1616 p.
6. Kim D. Fundamentals of Information Systems Security 3rd Edition, *Jones & Bartlett Learning*, 2016. 571 p.
7. Zetter K. Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon. *Broadway Books*, 2015. – 448 p.
8. Solomon M.G. Fundamentals of Communications and Networking. *Jones & Bartlett Learning*, 2014. 512 p.
9. Azad S. Practical Cryptography: Algorithms and Implementations Using C++ 1st Edition, *Auerbach Publications*, 2014. 365 p.
10. Mayer-Schönberger V. Big Data: A Revolution That Will Transform How We Live, Work, and Think, *Eamon Dolan. Mariner Books*, 2014. 272 p.
11. Antoniou J. Quality of Experience and Learning in Information Systems, *Springer Cham*, 2019. 110 p.
12. Jena A. K. Automated Software Testing: Foundations, Applications and Challenges (Services and Business Process Reengineering), *Springer* 1st ed., 2020. 179 p.

| | | | | | | |
|-----|------|---------|--------|------|----------------------------|------|
| | | | | | КВРІСТ. 200195.01.07.10 ПЗ | Арк. |
| Зм. | Арк. | №докум. | Підпис | Дата | | 61 |

13. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська. Київ : КПІ ім. Ігоря Сікорського, 2020. 192с

14. Мінухін С. В. Методи і моделі проектування на основі сучасних CASE– засобів. Навчальний посібник, Харків: Вид. ХНЕУ, 2008. 272 с.

15. В.С. Авраменко, А.С. Авраменко Проектування інформаційних систем: навчальний посібник, Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. 434 с.

16. Методичні вказівки для студентів спеціальності 8.05010102 «Інформаційні технології проектування» ОКР Магістр – Таврійський державний агротехнологічний університет, 2013. – 21 с.

17. Ушакова І. О. Основи системного аналізу об'єктів та процесів комп'ютеризації : навчальний посібник. Ч. 2, Вид. ХНЕУ, 2008. 324 с.

18. Ушакова І. О. Практикум з навчальної дисципліни "Основи системного аналізу об'єктів і процесів комп'ютеризації": навчальний посіб-ник, Х. : Вид. ХНЕУ, 2010. 344 с.

19. Muhammad K., Maryati M.. Lean IT Transformation Plan for Information Systems Development. *International Journal of Advanced Computer Science and Applications*. 2020. Vol 11 (8).

20. Mikalsen, M., Moe, Nils, B., Wong, Sut I., Stray, Viktoria. Agile Information System Development Organizations Transforming to Large-Scale Collaboration. *In proceedings of the Forty-Second International Conference on Information Systems (ICIS 2021)*, Austin, USA. 2021.

21. Isyaku. I. Iterative and Incremental Development Analysis Study of Vocational Career Information Systems. *International Journal of Software Engineering & Applications*. 11. 2020. pp. 13-24.

22. Chan O. SQL: Learn SQL (using MySQL) in One Day and Learn It Well. SQL for Beginners with Hands-on Project, 2018. 166p.

48. Костріков С. В. Інформаційні технології в БД. Навчально-методичний посібник, Харків : ПБВ ХНУ, 2015. 56 с.
49. Chan Ою. SQL: Learn SQL (using MySQL) in One Day and Learn It Well. *SQL for Beginners with Hands-on Project*, 2018. 166р.
50. Мулеса О.Ю. Інформаційні системи та реляційні бази даних. Навч.посібник. – Електронне видання, 2018. 118 с
51. Entity Framework URL: [Corehttps://learn.microsoft.com/en-us/ef/core/](https://learn.microsoft.com/en-us/ef/core/)
52. Muhammad K., Maryati M.. Lean IT Transformation Plan for Information Systems Development. *International Journal of Advanced Computer Science and Applications*. 2020. Vol 11 (8).
53. Mikalsen, M., Moe, Nils, B., Wong, Sut I., Stray, Viktoria. Agile Information System Development Organizations Transforming to Large-Scale Collaboration. *In proceedings of the Forty-Second International Conference on Information Systems (ICIS 2021)*, Austin, USA. 2021.
54. Nachenberg, C. Understanding heuristics: Symantec's bloodhound technology, *Symantec White Paper Series*. 1998. Vol. 34. P. 17.
55. Perriot F. Principles and practise of xraying, *The 14-th Virus Bulletin International Conference*, September 12, 2004: Proceedings. – 2004. – Pp. 51-56.
56. Kim D. Fundamentals of Information Systems Security 3rd Edition, *Jones & Bartlett Learning*, 2016. 571 p.
57. Zetter K. Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon. *Broadway Books*, 2015. 448 p.
58. Solomon M.G. Fundamentals of Communications and Networking. *Jones & Bartlett Learning*, 2014. – 512 p.
59. TeachTarget, Information systems, URL: <https://www.techtarget.com/whatis/definition/IS-information-system-or-information-services>
60. PressBook, Information systems for business and beyond, URL: <https://pressbooks.pub/bus206/chapter/chapter-1/>

ДОДАТОК А

Копія креслення «Екранні форми»

| Last Name | First Name | Hire Date | Office | Courses |
|-----------|------------|------------|--------------|---------------|
| Ivanov | Petro | 2023-02-15 | Lviv | 1 Programming |
| Sidorenko | Ivan | 2023-03-07 | Khmelnytskyi | 2 Web |

| Number | Title | Department |
|--------|-------------|------------|
| 1 | Programming | Temp |

| Number | Title | Department |
|--------|-------------|------------|
| 1 | Programming | Temp |

КаПІКТ. 200195.01.07.10 Е8

| | | | | | | | |
|-----------|--------------|----------|-------|-----|--------|-------|---------|
| № | Дат. | № докум. | Датум | Дія | Підпис | Місце | Масштаб |
| Розроб. | Володимир К. | | | | | | |
| Перевір. | | | | | | | |
| Із змісту | | | | | | | |
| І зміст | | | | | | | |
| Дата | | | | | | | |

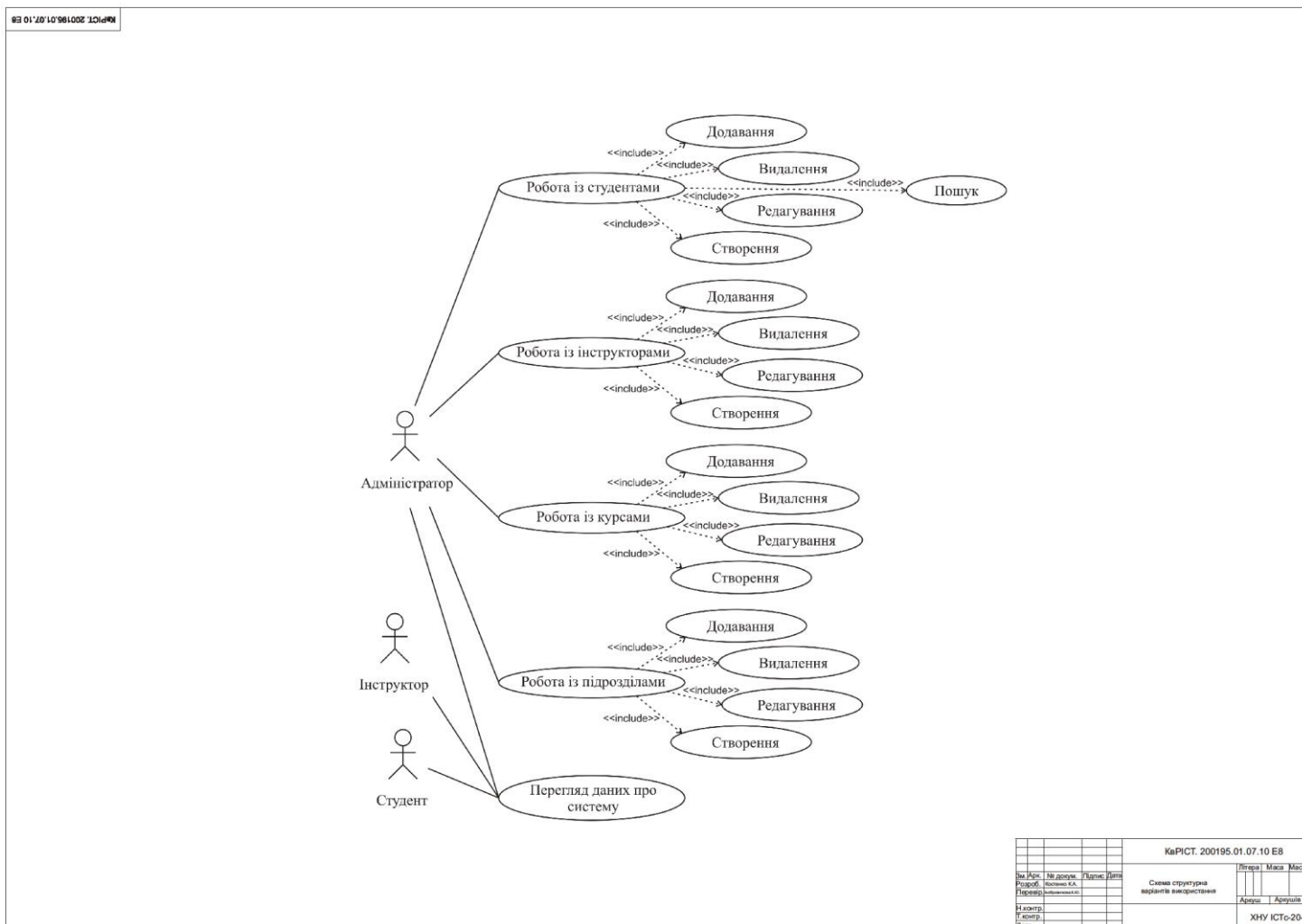
Креслення виготовлено за допомогою програмних форм

Архив: Архивує

XHV ICTo-20-1

ДОДАТОК В

Копія креслення «Схема структурна варіантів використання»



Ім'я користувача:
Кафедра КІ

ID перевірки:
1014814396

Дата перевірки:
26.04.2023 15:48:48 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
26.04.2023 15:50:17 EEST

ID користувача:
100005591

Назва документа: Костенко_Інформаційна web-система для ІТ школи

Кількість сторінок: 66 Кількість слів: 9011 Кількість символів: 69404 Розмір файлу: 7.57 MB ID файлу: 1014517564

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

11.3% Схожість

Найбільша схожість: 2.94% з Інтернет-джерелом (https://dut.edu.ua/repozitorii/komp_nauk/2021/%d0%a3%d1%81%d1%).

10.6% Джерела з Інтернету

204

Сторінка 68

1.88% Джерела з Бібліотеки

53

Сторінка 69

1.62% Цитат

Цитати

10

Сторінка 70

Посилання

1

Сторінка 70

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

5

Підозріле форматування

20
сторінок

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилоч в документах: 10%

| | | | | |
|--|----------|---------|-----------------------------|---------|
| ID: 112607 Назва: Інформаційна web-система для ІТ школи Додано в БД: 2023-04-26 Автора: К.А. Костенко Керівники: К.Ю. Бобровнікова Консультанти: Опоненти: | Документ | | Сумарний збіг по Базі Даних | |
| | Символи | Лексеми | Символи | Лексеми |
| | 59939 | 506 | 2841 (5%) | 38 (8%) |

Джерело плагіату

| ID | Опис | Наявність плагіату в документі | |
|----|------|--------------------------------|---------|
| | | Символи | Лексеми |
| | | | |

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Костенко Кирило Андрійович

Тема: Інформаційна web-система для ІТ школи

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 55

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано інформаційну web-систему для ІТ школи

2. Висновок про відповідність роботи дипломному завданню _____

Дипломний проект відповідає виданому завданню _____

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз відомих інформаційних систем і засобів освітньої діяльності. У другому розділі проведено вибір засобів реалізації та спроектовано концептуальну модель бази даних. У третьому розділі реалізовано прототипу інформаційної web-системи для ІТ школи

4. Позитивні сторони роботи: Запропоновано концептуальну модель бази даних ІТ-школи та спроектовано інформаційну web-систему ІТ-коли.

5. Негативні сторони роботи: Запропонований прототип інформаційної системи автосалону не має функції журналу оцінок.

6. Оцінка графічного оформлення та пояснювальної записки роботи:
пояснювальна записка та листи креслення виконані згідно діючих вимог

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому рівні.

8. Інші зауваження: -

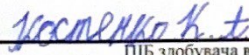
9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «задовільно» 3,75 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)
д-р ф-м. н. професор Берратюк Л.П.

“ 2 ” 06 2023р.

Завідувачу кафедри КІС
д-р.техн.наук, проф. Говорущенко Т. О.



ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи ІСТс-20-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

20.05.2023

дата



підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна web-система для ІТ школи

Автор: Костенко Кирило Андрійович

Спеціальність: 126 – Інформаційні системи та технології

Освітня програма: освітньо-професійна

Науковий керівник: Бобровнікова Кіра Юліївна, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|---|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

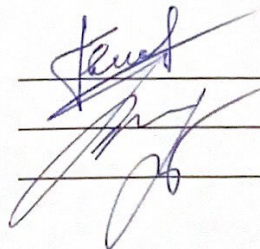
- 1) запозичення розміщені в розділі аналізу існуючих аналогів та відомих рішень, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 11,3 % і адресується до 257 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КПС



К.Ю. Бобровнікова

Є.Г. Гнатчук

Т. О. Говорущенко