

Державний вищий навчальний заклад  
“Прикарпатський національний університет імені Василя Стефаника”  
Кафедра інформаційних технологій

УДК 004

**ДИПЛОМНИЙ ПРОЕКТ**

Тема: Створення навчальної платформи на основі технології .NET

Спеціальність: 121 “Інженерія програмного забезпечення”

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

ДП.ПЗ-09.ПЗ

(позначення)

Рецензент

Доцент Ткачук В.М.  
(посада) (підпис) (дата) (розшифровка підпису)

Студент

ПЗ-41 Луців Є.М.  
(шифр групи) (підпис) (дата) (розшифровка підпису)

Нормоконтролер

Доцент Ткачук В.М.  
(посада) (підпис) (дата) (розшифровка підпису)

Керівник дипломного проекту

доцент Лазарович І.М.  
(посада) (підпис) (дата) (розшифровка підпису)

Допускається до захисту

Завідувач кафедри

Козленко М.І.  
(посада) (підпис) (дата) (розшифровка підпису)

Державний вищий навчальний заклад  
«Прикарпатський національний університет імені Василя Стефаника»  
Факультет математики та інформатики Кафедра інформаційних технологій  
Спеціальність 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Завідувач кафедри Козленко М. І.

студенту Луців Євгеній Миколайович

» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
НА ВИКОНАННЯ ДИПЛОМНОГО ПРОЕКТУ**

Студенту \_\_\_\_\_ Луціву Євгенію Миколайовичу

(прізвище, ім'я, по батькові студента)

1. Тема проекту Створення навчальної платформи на основі технології .NET

затверджена розпорядженням по факультету математики та інформатики від  
„ 25 ” Жовтня 2019р. №7

2. Термін здачі студентом закінченого проекту 22 травня 2020 р.

3. Вихідні дані до дипломного проекту категорія дистанційне навчання, дані  
використання дистанційного навчання технології розробки .Net Core MVC,  
HTML, TypeScript, SQL, оцінювання ефективності дистанційного навчання.

4. Зміст пояснювальної записки (перелік питань, що їх належить опрацювати)  
Аналіз літератури та дослідження особливостей аналогічних проектів до  
Навчальної платформи, аналіз основних характеристик Навчальної платформи  
та розробка моделей, розробка та опис Навчальної платформи на основі  
технологій .Net, бізнес план до Навчальної платформи

5. Перелік графічного матеріалу (з точним забезпеченням обов'язкових  
креслень) Мета та завдання дипломного проекту, актуальність Навчальної  
платформи, аналоги Навчальної платформи, технології розробки .Net для  
Навчальної платформи, розробка UML діаграм та діаграм бази даних,  
проекткування Навчальної платформи, розробка Front end Навчальної  
платформи, розробка Back end Навчальної платформи, аналіз ринку Навчальної  
платформи, план розвитку на ринку Навчальної платформи.

6. Дата видачі завдання \_\_\_\_\_

Керівник \_\_\_\_\_

(підпис)

Лазарович І.М.

(розшифровка підпису)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

Луців Є.М.

(розшифровка підпису)

## КАЛЕНДАРНИЙ ПЛАН

Номер і назва етапів дипломного проекту	Термін виконання етапів проекту	Примітка
1. Аналіз літератури та дослідження особливостей аналогічних проектів до Навчальної платформи	10.12.2019	Виконав
2. Аналіз основних характеристик Навчальної платформи та розробка моделей	07.01.2020	Виконав
3. Розробка та опис Навчальної платформи на основі технологій .Net	05.03.2020	Виконав
4. Бізнес план до Навчальної платформи	02.05.2020	Виконав
5. Оформлення пояснювальної записки	18.05.2020	Виконав

Студент

Луців Є.М.

(підпис) (розшифровка підпису)

Керівник проекту

Лазарович І.М.

(підпис) (розшифровка підпису)

## РЕФЕРАТ

Пояснювальна записка: 75 сторінок (без додатків), 37 рисунків, 13 таблиць, 24 джерел, 4 додатки на 76-84 сторінках.

Ключові слова: Навчальна платформа , .Net Core, база даних , Інтернет, навчання студентів, розробка, аналіз.

Об`єктом дослідження є : можливості впровадження сучасних технологій Інтернет в процес навчання студентів.

Мета роботи: створити навчальну платформу на основі технологій .Net

Стислий опис тексту пояснювальної записки: розробка Навчальної платформи, дослідження і аналіз аналогів проекту.

## **ABSTRACT**

Explanatory note: 75 pages (without appendix), 37 figures, 13 tables, 24 references, 4 appendix on 76-84 pages.

Key words: Learning platform, .Net Core, database, Internet, student learning, development, analysis.

Object of study: opportunities for the introduction of modern Internet technologies in the learning process of students.

Brief description of the text of the explanatory note: newsletter Training platform, research and analysis of project analysis.

## Зміст

<b>ВСТУП</b> .....	9
<b>1 АНАЛІЗ ЛІТЕРАТУРИ ТА ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ АНАЛОГІЧНИХ ПРОЕКТІВ ДО НАВЧАЛЬНОЇ ПЛАТФОРМИ</b> .....	11
1.1 Платформа для навчання.....	11
1.2 Дослідження актуальності навчальної платформи.....	12
1.3 Дослідження та аналіз аналогів навчальних платформ .....	18
1.4 Постановка задачі.....	24
<b>2 АНАЛІЗ ОСНОВНИХ ХАРАКТЕРИСТИК НАВЧАЛЬНОЇ ПЛАТФОРМИ ТА РОЗРОБКА МОДЕЛЕЙ</b> .....	26
2.1 Аналіз основних характеристик .....	26
2.2 Технологія для розробки ASP.NET Core MVC для створення навчальної платформи.....	27
2.3 Системний аналіз предметної області для Навчальної платформи.....	30
2.4 Функції користувачів навчальної платформи .....	31
2.5 Проектування архітектури програмної системи.....	32
2.6 Проектування бази даних для Навчальної платформи .....	34
2.7 Проектування шаблонів підсистеми створення.....	39
2.8 Декомпозиція моделі для Навчальної платформи.....	41
2.9 Розробка авторизація на основі ролі в ASP.NET Core .....	44

					ДП.ІПЗ-09.ПЗ							
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Створення навчальної платформи на основі технології .NET							
<i>Розроб.</i>		Луців Є.М.								<i>Лім.</i>	<i>Аркуш</i>	<i>Аркуші</i>
<i>Перев.</i>		Лазарович І.М.								Н	6	84
<i>Н. контр.</i>		Ткачук В.М.								ПНУ ІПЗ-41		
<i>Затверд.</i>		Козленко М.І.										

<b>3 РОЗРОБКА ТА ОПИС НАВЧАЛЬНОЇ ПЛАТФОРМИ НА ОСНОВІ ТЕХНОЛОГІЙ .NET .....</b>	<b>45</b>
3.1 Розробка інтерфейсу в Навчальній платформі .....	45
3.2 Розробка бази даних для Навчальної платформи .....	46
3.3 Розробка головного модуля Навчальної платформи.....	48
3.4 Графічний опис Навчальної платформи .....	49
3.5 Розробка головного програмного модуля в навчальній платформі.....	58
3.6 Організація компонентів в Навчальній платформі.....	64
<b>4 БІЗНЕС ПЛАН ДО НАВЧАЛЬНОЇ ПЛАТФОРМИ .....</b>	<b>66</b>
4.1 Резюме.....	66
4.2 Маркетинг Навчальної платформи на .Net.....	66
4.3 Тип розроблюваних Навчальної платформи організаційний план .....	67
4.4 Зведений календарний план реалізації Навчальної платформи .....	68
4.5 Склад фінансового бюджету .....	69
4. 6 Оцінка можливих ризиків підприємства .....	70
<b>ВИСНОВКИ .....</b>	<b>72</b>
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....</b>	<b>73</b>
<b>Додаток А .....</b>	<b>766</b>
<b>Додаток Б .....</b>	<b>799</b>
<b>Додаток В .....</b>	<b>80</b>
<b>Додаток Г .....</b>	<b>84</b>

## Перелік основних позначень і скорочень

ILC - Independent Learning Centre

ДН - дистанційне навчання

ДО - дистанційна освіта

ДСН – дистанційна система навчання

ЗВО - закладах вищої освіти

ІКТ – інформаційно-комп'ютерні технології

НП – навчальна платформа (назва проекту).

СДН - система дистанційного навчання

					ДП.ІПЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8



## ВСТУП

На сьогоднішній день система освіти забезпечує денну, заочну, екстернатну та дистанційну форми навчання.

Стратегічна мета освіти в теперішньому світі – це підвищити доступність якісної освіти, що буде відповідати сучасним потребам суспільства і кожного громадянина.

Дистанційне навчання - це якісно нова, прогресивна форма навчання, що виникла в останній третині ХХ століття, завдяки новим технологічним можливостям, появі яких сприяли інформаційна революція і виникнення ідеї відкритої освіти.

В порівнянні з традиційними формами дистанційне навчання має ряд переваг, що гарантують його перспективу розвитку. Найсуттєвіші - гнучкість і доступність. Людина може навчатися в будь-який час за обраною ним програмою, не покидаючи місце основної діяльності. Все, що необхідно для дистанційного навчання - це комп'ютер з доступом до Інтернету. Навчальні матеріали, лекції та завдання для тих, хто навчається, відправляються в електронному вигляді. Крім того, є можливість індивідуально спілкуватися з вашим лектором за допомогою електронної пошти.

Онлайн-навчання є дешевшим, оскільки скорочуються витрати на транспорт, підручники, канцтовари та ін.

Дуже важливо, що дистанційна освіта є індивідуальною, на відміну від класичних форм навчання. Навчальна програма складається для кожного по-різному, спираючись на індивідуальні потреби студента.

Варто зазначити, що при великій кількості переваг система ДН не є досконалою. У ній присутні проблемні моменти такі як: некоректно використовуються інформаційні технології, відсутня соціальна взаємодія. Хоча вони не мають істотного впливу на те, наскільки якісно і ефективно навчаються студенти.

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

Інколи відсутність у користувачів належних навичок роботи з комп'ютерними технологіями може зменшити потенційні можливості як лекторів, так і студентів. Та цю проблему можна вирішити, підвищивши кваліфікацію в галузі комп'ютерних технологій.

Великий недолік онлайн-навчання – відсутність соціальної взаємодії, що знижує ефективність навчального процесу. Ця проблема вирішується за допомогою вебінару, семінару, конференції, які можуть проводитися онлайн.

Таким чином, дистанційне навчання в даний час є найперспективнішою і досить ефективною системою освіти. Тому вибрана мною тема дослідження є актуальною в сучасних умовах життя.

Предметом дослідження є система дистанційного навчання.

Об'єктом дослідження є можливості впровадження сучасних технологій Інтернет в процес навчання студентів.

Метою створення Навчальної платформи є те щоб дати можливість користувачам забезпечити зручне та якісне навчання з хорошим адміністрування і правильною роботою системи . Також система розрахована на україномовних користувачів. В даній системі буде наданий такий функціонал: студентам та учням: легкий доступ до матеріалу, правильну систему оцінювання; лекторів, викладачів та вчителів: можливість створювати якісні курси та домашню роботу, додавання студентів та учнів до курсів, а також видавати їм домашні завдання; адміністратору: правильне керування системою навчання, а також створення користувачів таких як лектор та студент.

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

# 1 АНАЛІЗ ЛІТЕРАТУРИ ТА ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ АНАЛОГІЧНИХ ПРОЕКТІВ ДО НАВЧАЛЬНОЇ ПЛАТФОРМИ

## 1.1 Платформа для навчання

Як можна бачити, у сучасному житті технологічний прогрес не зупинився – незважаючи на різні перешкоди, він все ще швидко рухається вперед. Тепер, коли у нас є Інтернет, він змінив усі існуючі концепції останніх кількох років. Справді, завдяки розвитку Інтернету нам стала помітна реалізація великої кількості проблем. Однією з таких проблем є творче новаторське використання новітніх технологій в освітній сфері. За допомогою такої системи навчання учні відкривають для себе різноманітні інформаційні дані, які скорочують час вивчення потрібної їм інформації.

Використовуючи існуючі засоби телекомунікацій, система відкритої освіти надає змогу школярам використовувати інформацію, якою володіє весь світ. Більше того, це дає можливість молодим людям повністю реалізувати їх потенціал. Тому, відповідно до нинішньої ситуації, ми можемо підсумувати, що без цього методу неможливо розвинути власну індивідуальність, та загалом неможливо розвиватись і розвивати суспільство.

Після цього очевидно стає, що дистанційна освіта (ДО) - одна з галузей доступної освіти, яка швидко розвиваються та є доступними. Вона дає перспективи реалізації таких освітніх принципів:

- можливість будь-кому використовувати існуючу систему освіти, здолавши різноманітні фізичні обмеження людей, та збільшення кількості учнів;
- індивідуальний підхід до навчання для кожного учня, створивши комфортні умови для кожного особисто – як для викладача, так і для того, хто навчається, врахувавши індивідуальні психологічні особливості другого;
- постійно розвивати інформаційну компетентність, навички в роботі, які передбачають використання сучасних засобів телекомунікацій;

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

- соціалізувати початковий процес та врахувати особисті та комунікативні особливості всіх учнів.

Дистанційне навчання (ДН) – це шлях, яким реалізують систему дистанційної освіти. Це перша світова рішення заміни існуючою класно-урочною системи.

Головними компонентами системи ДН є навчально-методичні матеріали, що утворюють комплекти, які мають спільну єдину методологію побудови та структуру зв'язків, з різноманітних предметів. Вони доступні у таких формах:

- комп'ютерні демонстрації;
- електронні бібліотеки;
- імітаційне моделювання із застосуванням комп'ютерних засобів;
- дистанційні курси;
- періодичні е-видання;
- інтерактивні навчальні ресурси;
- онлайн-уроки та навчальні фрагменти;
- віртуальне середовище навчальної та практичної діяльності;
- е-тренажери;
- системи, щоб контролювати та вимірювати результативність навчання в електронному форматі [19].

Вищезазначені блоки повинні бути розроблені так, щоб вони відповідали державним освітнім стандартам для кожної дисципліни, зважаючи на обсяг та склад, змістове наповнення роблять достатнім або надмірним, щоб організувати процес зі студентами, які отримують різні базові знання та навички навчання, при використанні різних методів навчання.

## 1.2 Дослідження актуальності навчальної платформи

Досвід показує, що дистанційна освіта є відносно дешевою формою навчання для країни. Оцінюючи світові освітні програми, можна побачити, що

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

ДН в середньому на 50% дешевша, ніж будь-яка з форм навчання, що належить до традиційних. Аналізуючи діяльність ДН-центрів, зроблено висновок, що вартість підготовки експерта при використанні ДН становить близько 60% від вартості навчання фахівця на денній формі. Однак вартість студентських курсів коливається від безкоштовних до досить дорогих, які коштують тисячі доларів.

На кінець 1997 року існувало приблизно тисячу навчальних установ ста семи країн, в яких використовували дистанційне навчання. Близько п'ятдесятьма мільйонами людей в тому ж році була отримана вища освіта при застосуванні такої форми навчання, роком пізніше чисельність таких випускників збільшилася до дев'яносто мільйонів. Експерти прогнозують, що через три роки році здобувачів вищої освіти за допомогою дистанційного навчання становитиме 120 млн чоловік.

Вивчивши досвід введення дистанційного навчання в систему освіти кожної з держав, можна визначити десять основних причин, які мотивують навчальні заклади використовувати систему ДН. Перелічимо їх від найбільш значущих до менш:

- підвищення якості освіти;
- прагнення учнів вдосконалити свої знання та вміння;
- пріоритет нових методів навчання;
- зростання затребуваності нової форми навчання;
- дохід;
- постійна взаємодія викладача та студента;
- потреба вижити в інформаційному суспільстві;
- необхідність зменшити витрати на перебудову освіти;
- міністерський наказ;
- скорочення адміністративного персоналу.

З одного боку, лише 2% респондентів, згідно з аналізом, мають думку, що розробка ДН дешевша в порівнянні з традиційною освітою. А з іншого – 35% переконані, що поставка ДН дешевше традиційного навчання. Покладаючись на власний досвід, автори монографії схиляються до того, що розробка ДН – це

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

дуже трудомісткий процес, в якому беруть участь викладач, дизайнер, методист, адміністратор. Врахування основних переваг дистанційної форми навчання при розробці ДН впливає, насамперед, на те, наскільки буде успішним проведення ДН. Постановка ДН залежить від розвитку інформаційно-комунікаційної структури навчальних закладів та відповідної підготовки педагогічних кадрів.

У процесі ДН важливі надані для підтримки ресурсу можливості і сам його зміст. В результаті аналізу відомо, що 89% ресурсів, наданих для ДН, потребують постійної підтримки (мається на увазі, що особистий контакт студентів та викладачів не являється обов'язковим), 8% потребують, щоб був особистий контакт, і тільки 3% не мають ніякої підтримки. Розглянувши інший бік, маємо, що 14% ресурсів мають цілодобову підтримку, а 86% - обмежені в часі (тобто підтримка не відбувається 24/7).

На противагу стаціонарному навчанню з'явилося дистанційне, яке поступово закріплюється в методичній системі освіти передових країн. У зв'язку з диференційованим розвитком технології ДН завдяки глобальним комп'ютерним комунікаціям в державах СНД і Заходу, ми розглянемо окремо розвиток цих технологій в СНД, Америці, Європі та Азії.

Світовими лідерами реалізації дистанційного навчання можна сміливо вважати Сполучені Штати Америки та Канаду. У багатьох університетах, коледжах та навчальних центра в США та Канаді відбувається успішне впровадження дистанційного навчання за допомогою Інтернету. На реалізацію впливає доступність Всесвітньої мережі у вузах та національних школах цих країн. Комп'ютеризація жителів країн, розвинений зв'язок і телекомунікація є міцним фундаментом ДО. Дані фактори сприяли розвитку технології дистанційної освіти США та Канади. В сьогоденні ці країни володіють багатьма дистанційними програмами, які вільно забезпечуються міністерствами освіти, окремими університетами, коледжами та компаніями (щоб підвищити кваліфікацію працівників). Одним лише ІЛС, що знаходиться в Онтаріо, організовуються близько 40 000 різноманітних курсів кожного року. В даних, який надає Світовий банк, вказуються заклади, яких налічується більше 3000 зі

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

спеціалізацією є професійного навчання в діалоговому режимі. У більшій частині штатів США створена велика кількість віртуальних університетів, місцеві коледжі можуть запропонувати проходження курси в дистанційній формі.

Новим типом закладу, світовим лідером в галузі ДО, європейським новатором використання дистанційного навчання вважається Відкритий університет Великобританії, рік його заснування – тисячу дев'ятсот шістдесят дев'ятий. Через двадцять сім років його факультети вже налічували біля 200000 студентів. Програма, розроблена Відкритим університетом Великобританії, використовується в навчанні двадцяти одної країни. В цій же країні Інститутом освіти Лондонського університету забезпечуються розповсюдження ДО та її високий рівень. Практикується використання додипломних та післядипломних курсів. Зараз багато європейських країн практикують ДН для забезпечення перспективної освіти. Для прикладу, Національний університет дистанційної освіти, який знаходиться в Іспанії впроваджує ДН за системою вищої освіти для тих, хто не може навчатися в звичайних університетах.

Німецька дистанційна освіта також є своєрідною. Вона брала початок з безпосередньої дистанційної підготовки спеціалістів вищої освіти, що була запланована напередодні. За рахунок долучення земель зі Сходу, сфера дистанційних послуг розширилася. Заочний університет Хагену є зразковим закладом вищої освіти, який орієнтується на дистанційне надання освіти. Цей університет навчає близько півсотні тисяч студентів за один рік. Але зауважимо, що в результаті невідповідності профорієнтаційним вимогам, студентів виключають з вищого навчального закладу і в результаті, дипломованими стають менше двадцяти відсотків тих, хто навчається.

Уряди країн Європи розглянули документи, в яких вектор був спрямований на підтримку впровадження інформаційно-комп'ютерних технологій (ІКТ) у школи, університети та інститути. Головною метою цих документів стало забезпечення індивідів можливістю проживати у соціумі без неправильного трактування інформації, а в особливості і до здатності аналізу великої кількості джерел, щоб сформувавши правильну інформаційно-орієнтаційну сферу. Це дало

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

змогу впровадити інноваційні методи і принципи у сферу користуванням ІКТ на різноманітних сходах освіти, яка спрямована на усіх громадян . Виокремлюють три ключові напрямки запровадження ІКТ в Європейських країнах: підвищення кваліфікації викладачів з можливістю використовувати ІКТ, розвиток навчальних інформаційних послуг і запровадження технологічної системи інфраструктури.

Винайдення великої кількості центрів ДН та організації, для її поширення (European Distance Education Net) є найкращим свідченням про ефективність запровадження ІКТ.

Зокрема, особливої уваги також варто приділити і розвитку засобів ІКТ в країнах Азії. Завдяки ньому було ініційовано відкриття соціальних інститутів ДН у таких країнах як: Пакистан, Таїланд, Тайвань, Шрі-Ланка, Індія та Китай, який став осередком розвитку інноваційної форми навчального процесу. Для дистанційного навчання не потрібно додаткових фінансових витрат та бути безпосереднім учасником стаціонарного навчання тому вона вкорінилася в університети настільки сильно, що при її відсутності освіта б функціонувала в рази менше.

Пройшовши певний історичний розвиток, азіатські країни, а зокрема їхні університети вирішили застосувати поняття "кооперація". Даний феномен змістовно відображає проблеми дистанційної освіти. Засобами у співвідношенні з ДН виступають: телекомунікації, радіопередачі, розповсюдження CD технологій, а також і друкованих матеріалів. Особливими якостями кооперація наділяється тоді, коли постає питання інтеграції очної та дистанційної форми навчання у закладах вищої освіти (ЗВО). Додаткової уваги до себе притягує й розвиток ДН на теренах колишнього СРСР, де глобалізація стала дотичною до розвитку навчальної діяльності. Концепції створення та розвитку науки відобразилася на подальшому функціонуванні дистанційного навчання.

Перспективні комп'ютерні технології навчання стали полотном для відображення новітніх стратегій розвитку системи освіти, яскравим прикладом є Молдова, яка змогла за допомогою цих нових можливостей підняти рівень

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16



ефективності та покращити умови надання навчальних послуг. Передові ЗВО Республіки Молдови стали ініціаторами створення сучасної програми, яка в себе включала підвищення кваліфікації та отримання можливості первинного навчання.

Литовська Республіка відзначилася цікавою і новітньою методикою розробки системи навчання, яка була запроваджена в двохтисячному році. Якщо детальніше розглянути цю стратегію, то можна виокремити концепції освіти, які мають докорінно змінені:

- соціалізація освіти, відношення до соціуму;
- перебування та навчання в закладах середньої освіти;
- концепцію і принципи навчання;
- комплекс частин технології комунікації потрібних для достатньо високого рівня функціонування людства в системі відносин;
- роль наукового керівника і спеціалізація в галузі освіти;
- взаємозв'язок освіти та науки для максимально високого розвитку життєдіяльності суспільства;
- забезпечення правильного стратегічного підходу та фінансових ресурсів управління і фінансування ІКТ.

Фахівці у міністерстві освіти і науки Литви вирішили підтримувати ДН, оскільки воно постає перед нами як майбутнє, як можливості розвиватися у сфері наукових технологій максимально швидко та дієво. Осередки інфраструктури цієї країни апелюють до обов'язкового використання ДН у науковому пізнанні, характеризуючи це як важіль розвитку, на який залишилося натиснути. Головним з-поміж них виокремлюють Технологічний Університет, що у Канаусі. Щодо країн Середньої Азії, то в загальному можливість користуватися ІКТ, на сьогоднішній день, знаходиться на початковому рівні розвитку.

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

### 1.3 Дослідження та аналіз аналогів навчальних платформ

У зв'язку з тим, що інформаційні технології розвиваються, суспільство стикнулося з проблемою: як створити перспективну нову систему освіти, що підготує людей до того, як жити в цивілізації з новими умовами. Звідси очевидним, є те, чому появилася така нова форма навчання, як дистанційна, яка функціонує разом із такими формами, як стаціонарна, заочна та іншими традиційними. Для того, щоб забезпечити успішне проектування, розвиток та ініціювання ДСН необхідним є створення інфраструктури відповідного закладу освіти, проведення підготовки спеціалістів, розробка навчальної програми і тому подібне. Важлива складова частина ДН – його впровадження, використовуючи інформаційні технології, особливо систему управління навчальним процесом, яка призначена для того, щоб розробляти, управляти та поширювати навчальні матеріали, що обов'язково включає спільний доступ великої кількості користувачів.

Традиційні види освіти в нашій країні включають денну, заочну, суботню форми навчання, а також екстернат. Очна освіта компонується з денної та вечірньої форм. До їх особливостей належать: навчальний процес, ключовим елементом якого є двостороння взаємодія між викладачем та студентом; використання таких форм навчання, які називаються груповими: консультація, лекція, семінари, практична та лабораторна роботи, колоквиуми тощо; чітке задання переліку предметів для вивчення; час навчання, який чітко кредитується.

Очевидно, що ДО базується на принципах та особливостях навчальних форм, які є традиційними, крім того, доданих нових особливостях, таких як використання навчальних матеріалів за допомогою комп'ютерної мережі Інтернет, інтерактивна взаємодія між тими, хто навчається та інше.

Принцип, який полягає в тому, що навчання спрямовується на те, щоб вирішити завдання освіти і відбувався загальний розвиток студентів в традиційних формах освіти, вказує на те, що увага викладача повинна бути спрямована не тільки на вміння та як вирішити завдання, а й на те, наскільки

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

ефективною є проведена система заходів виховання у розглянутих темах. У ДСН він називається принципом креативності характеру пізнавальної діяльності. Використовуючи інтерактивні технології, характер креативності ДН здійснюється завдяки суперництву та змагань багатьох студентів, тим самим сприяючи підвищенню їхнього творчого потенціалу.

Один із головних принципів, які є фундаментом традиційної форми освіти, вважається принцип науковості. Він опирається на зв'язок між наукою та дисципліною, яку вивчають, і потребує, щоб матеріали навчання містили інформацію про наукові концепції, моделі, нові відкриття та сучасні здобутки. Даний принцип набуває більш базової форми в дистанційному навчанні: він покладений вирівнювати основний характер навчання і пізнавальні потреби студентів. Суть цього принципу полягає в тому, що якщо навчання орієнтоване на те, щоб визначити основи та залежності між різними процесами, які відбуваються в навколишньому середовищі, то його можна вважати фундаментальним. При цьому принцип пред'являє до студента певні вимоги, які включають високу мотиваційну потребу, корекцію та самокритику, бажання розвиватися, спрямування особистості до того, щоб досягати поставлену перед собою мету [21].

Зміст принципу науковості має логічне продовження змісту принципу у принципі вільного вибору інформації, отриманої в результаті певної діяльності: виключено те, що наявне єдине досконале джерело інформації, тому навчання спрямоване не на саму інформацію, а на те, яким чином вона перетворюється та опрацьовується, коли студенти беруть участь в дискусії, телеконференціях, використовують у роботі пошукові машини тощо.

У традиційній формі навчання виділяють ще один досить важливий принцип навчання – послідовності та систематизації, що вимагає впорядкованості, класифікації та логічного зв'язку поданих знань для того, щоб покращити результати навчання. Зате дистанційне навчання дає змогу самому вибрати власну навчальну ціль, а також в якій формі та темпі навчатися. Даний принцип ще називають принципом освітньої траєкторії студента.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

Вважається, що засвоєння знань відбувається краще, коли у високій концентрації сприйняття задіяна вся людська сенсорна система. Це відображається в принципі наочності. Через те, що у ДН немає безпосереднього контакту між слухачами та викладачами, то перейдено до формулювання принципу віртуалізації навчального процесу. При навчанні в дистанційній формі часто застосовуються електронні видання, відео- конференції та ролики, ілюстраційний матеріал тощо.

Поряд з тими принципами, що ми вже описали, можна розглянути деякі принципи ДН, які створені завдяки тому, що інформаційні технології активно розвивалися та використовувалися.

Першим буде принцип ідентифікації. Його важливість полягає у тому, що під час дистанційного навчання можна більше сфальсифікувати в порівнянні з навчанням традиційної очної форми. Контроль за самостійним виконанням можна забезпечити, використовуючи технічні засоби такі як відеозв'язок.

За допомогою інформаційних технологій викладач має мати змогу контролювати навчальний процес, можливість внесення змін в курс, який викладає. Також повинен бути доступний контакт як між викладачем та студентами, так і останніх між собою. Цей принцип є принципом інтерактивності.

Ефективність ДН визначається не менш значимим принципом – початкових знань. Його зміст: користувачі дистанційних курсів повинен мати навички роботи з комп'ютером, можливість онлайн-доступу, необхідно вміти працювати в Інтернеті та мати потрібні технічні засоби, щоб повноцінно навчатися.

Важливим принципом навчання являється педагогічна доцільність застосування ІТ. На всіх етапах проектування і організації, через які проходить дистанційний навчальний процес, потрібно педагогічно та доцільно оцінювати те, як використовуються новітні інформаційні технології, що впливають на кожен компонент навчання: зміст, завдання, методи тощо.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

Звернемо увагу, що побудова розглянутих педагогічних принципів буде змінюватися протягом всього часу, також можливі новації, у зв'язку з тим, що дистанційне навчання все більше розвивається та глибше впроваджується.

Втім саме дані принципи відграють важливу роль, коли відбувається побудова системи дистанційного навчання. Ці системи складаються з великої кількості взаємопов'язаних модулів. У той же час потрібно зосереджуватися на основних функціональних можливостях для забезпечення сучасних систем ДН. Різні системи в більшості випадків мають подібний список можливостей.

Доступний навчальний контент. Необхідним є забезпечення можливості авторизації користувачів, управляти правами їх групи, контролювати доступність матеріалів навчання.

Забезпечити зручні засоби управління. Типовий набір функцій включає в себе реєстрацію користувача, керування групою зареєстрованих осіб, регулювання дистанційних курсів та заходів управління та інше.

Надати засоби комунікації між тими, хто використовує курс. На сьогоднішній день ми маємо достатні можливості, щоб забезпечити зв'язок, основними є аудіо- та відеоконференція, чат, блог, форум і тому подібне.

Формувати навчальні плани, використовуючи моделі, які управляють компетенціями. Для того, щоб розподілити навчальні плани, потрібно побудувати модель, яка вказує виконувану користувачем роль та його компетенції відповідно до ролі, яка йому задана. Надалі кожній можливості відповідає блок курсів, які необхідні для вивчення та блок домашньої роботи, яка обов'язкова до виконання. Аналізуючи побудовану модель системи створюється відповідний навчальний план для кожного з користувачів.

Формувати звіти. Важливо забезпечити можливість формувати звіти, якщо користувач забажає.

Встановити зв'язки між системою і різними видами інформаційних систем. Перша не може бути відокремлена від будь-якої інформаційної системи, місцем перебування яких є спільне середовище.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

Формувати складні розподілені системи. Завдяки побудові багатовузлової системи ДН варто спростити доступ користувача до будь-якого навчального курсу. Якщо відсутній постійний доступ до навчальних систем, то потрібним є автономний клієнт для забезпечення доступу до потрібного матеріалу.

На сьогоднішній день застосовується великий спектр систем дистанційного навчання та керування навчанням у дистанційній формі з використанням відкритого коду (безоплатних), а також використовуються платні, широковживані та вузькоорієнтовані.

ATutor є модульною системою дистанційного керування навчальним процесом, у якій код відкритий. Її поширення відбувається на основі GNU GPL. Щоб установити дану систему потрібен комп'ютер, в якому є веб-сервер Apache 1.3.x, PHP версії вищої за 4.2.0 та MySQL версій, які вищі 3.23.x і 4.0.12 (4.1.x і 5.x не можуть підтримуватися). При розробці системи враховувалася доступність та можливість адаптації при бажанні користувачів. Якщо розглядати серверну операційну систему, то обмеження відсутні – вона є багатоплатформною [1].

Claroline є платформою ДН та електронної діяльності, код якої відкритий. Основа поширення аналогічна до основи ATutor – GNU GPL. Працює в таких операційних системах: Linux, Mac і Windows. Ця платформа забезпечена інстинктивно зрозумілим інтерфейсом адміністрування. Основа організації Claroline LMS – поняття простору, який пов'язаний з курсами чи професійною діяльністю педагога. Будь-який простір має інструментарій для того, щоб створити, організувати та управляти навчальним матеріалом; можливості, щоб забезпечити взаємодію користувачів і таке інше [2].

Live@EDU є системою ДН, виконаною за допомогою технології Active Server Pages на платформі Microsoft. Щоб відбувалася установка та коректна робота, серверну частину необхідно забезпечити ОС Microsoft Windows NT Server 4.0, БД Microsoft SQL Server 7.0 та Microsoft Internet Information Server 4.0. Клієнтську частину потрібно забезпечити установленою операційною системою, яка гарантує доступ до Інтернету та браузера, що обслуговується протоколом

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

HTTP 3.0, і програмним забезпеченням, щоб переглядати та створювати матеріал для лекцій [3].

eFront – це нове покоління системи навчання в електронному форматі, в якій поєднані призначення систем, перша з яких – управління навчальним процесом, а інша полягає в створенні та управлінні навчальними матеріалами. Її використовують, щоб організувати навчальний процес у навчальному закладі і підвищити кваліфікацію, атестацію та відбір спеціалістів у різних компаніях. На Адміністраторі, Викладачі та Студенті побудовано дану систему [4].

Moodle є пакетом модульного ПЗ, код якого відкритий (ліцензія GNU GPL), його призначення – створити курси дистанційного навчання та web-сайти. Moodle орієнтується на співпрацю викладача та студента, її ще використовують, щоб підтримувати очні курси. Moodle можна встановити на комп'ютери, на яких підтримується PHP та робота із СУБД MySQL, PostgreSQL, Microsoft SQL Server. ПЗ – багатоплатформне. Проект має фундамент, який складається з п'яти принципів, що разом називаються «соціальним конструктивізмом»: сьогодишнє навчальне середовище визначає нас всіх одночасно потенційним викладачем і студентом; вчимося успішно, коли створюємо дещо або пояснюємо дещо іншим особам; вагомому внеску в навчальний процес сприяє те, що ми спостерігаємо, як працюють наші колеги; сприйняття других людей допомагає більше їх вивчати; гнучкість навчального середовища для забезпечення учасників навчання простим інструментом, щоб реалізувати їхні навчальні потреби [5].

SharePoint LMS є системою ДН, основа якої – потужна багатофункціональна платформа MOSS 2007. Це комплексне рішення, що створює цілісний інформаційний простір навчання завдяки об'єднанню викладачів, студентів, адміністраторів та здатний забезпечити їх спільну роботу. SharePoint LMS – платна. Її використовують не тільки навчальні заклади та центри навчання, а й підприємства, організації, державні структури [6].

Learningapps сприяє створенню різноманітних вправ різної тематики. Так само можна використовувати вправи, які вже підготовлені. Але варто зазначити, що велика кількість не українською мовою. Є різні типи вправ. Наведемо деякі

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

прикладі: встановлення відповідностей, сортування в стовпчики, правдивість і неправдивість порад. Зразу видно результат, чи правильну відповідь ми обрали [7].

Метою нашого університетського ЦДНКЗ є впровадити сучасні ІКТ навчання в освітній процес та наукові дослідження, насамперед дистанційне навчання. Основні задачі і напрями діяльності d-learn: впровадити та управляти системою ДН університету; підготувати та керувати роботою, яку виконують локальні адміністратори ДН; досліджувати освітні і науково-методичні послуги ДН в університеті; підготувати розробника і тьютора курсу ДН; проводити дослідження, як відбувається практика та виклад теорії ДН; моніторити та впроваджувати інновації в програмному та апаратному забезпеченні ДН; забезпечити кооперацію національних й міжнародних освітніх закладів та установ, інших учасників е-навчання з нашим університетом [8].

Ми розглянули основні дидактичні принципи дистанційної та традиційної форм навчання, визначили переваги та недоліки дистанційного навчання. Враховуючи основні дидактичні принципи дистанційного навчання, виведено головні модулі функціоналу сучасної системи управління дистанційним навчанням, в яких містяться забезпечення адміністрування системи, доступ до навчального матеріалу, засоби, що забезпечують комунікацію учасників ДН тощо. На основі модулів, які сформульовані, розглянули та порівняли наявність та реалізацію модулів у таких систем дистанційного навчання (СДН) – Moodle, Claronline, ATutor, SharePointLMS, Live@EDU, eFront, Центр дистанційного навчання та контролю знань [20].

#### **1.4 Постановка задачі**

У комплексному дипломному проекті моя задача – організувати роботу з навчальними курсами. Необхідно реалізувати: можливість додавання, видалення і редагування навчального курсу лектором і адміністратором;

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24



розмежувати права на доступ: врахувавши, що видалення, додавання і редагування всіх навчальних курсів може здійснюватися користувачем-адміністратором, а користувачем-лектором може здійснюватися додавання, редагування і видалення тільки свого навчального курсу проходження і перегляд користувачем-студент курсів; перегляд студентом оцінок за свої предмети; ідеологія і методи дистанційного навчання не повинні відрізнятися від традиційних; система дистанційного навчання повинна надавати доступ до навчальних матеріалів; інтерфейс системи повинен бути доступним будь-кому з користувачів; пріоритетними є стійкість та безпечність системи при роботі; робота в системі має бути зручна для використання; потрібно передбачити, що система розширюватися і масштабуватися. Отже, основне, що вимагають від системи ДН – щоб процес дистанційної форми навчання щонайменше відрізнявся від традиційних форм.

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

## 2 АНАЛІЗ ОСНОВНИХ ХАРАКТЕРИСТИК НАВЧАЛЬНОЇ ПЛАТФОРМИ ТА РОЗРОБКА МОДЕЛЕЙ

### 2.1 Аналіз основних характеристик

Щоб додатково проілюструвати, що процес розвитку стає більш зрозумілим, необхідно визначити певні явища процесів та об'єктів, що використовуються в моделювальному процесі.

СДН програмне та апаратне забезпечення, що дозволяє студентам використовувати інтернет для дистанційного навчання. Метою цього проекту є розробка СДН.

Адміністратор є користувачем системи та виконує функції, що керують навчальним процесом.

Лектори є користувачам системи та виконують функцію створення матеріалу для курсу та контролюють студентами в даному курсі.

Студенти користувачі що проходять навчальні курси .

Матеріал – інформація, створена лектором системи для подання студентові та збереження їх у бібліотеці системи. Навчальні матеріали включають лекції та домашні завдання.

Курс - набір навчальних матеріалів на одну або декілька тем. Курс та його наповнення створює лектор.

Лекція - набір інформаційних матеріалів подається у текстовій, графічній та анімаційній формі. Лектори в основній частині створюють лекції при створенні курсів.

Творцем курсу є лектор або адміністратор, який розробив курс, а керівник курсу - лектор, який має на меті контролювати ефективність студентів, які вивчають курс та консультуються на даному курсі.

					ДП.ІПЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

## 2.2 Технологія для розробки ASP.NET Core MVC для створення навчальної платформи

Структура контролера ASP.NET Core Model view controller (MVC). View - це структура з відкритим кодом та легкою базою. Це рамка, яка дуже перевірена і оптимізована для використання з ASP.NET Core. ASP.NET Core MVC - це першокласний вибір для розробників, які прагнуть створювати сучасні та динамічні веб-додатки за допомогою шаблонів на основі шаблонів. Він всебічно підтримує тестову розробку, використовуючи останні веб-стандарти [9].

Microsoft випустила останню версію попереднього перегляду ASP.NET Core MVC 6.0 з деякими вдосконаленнями. Останній стабільний реліз - 5.2.7, який був запущений 29 листопада 2018 року. Отже, важливо мати уявлення про те, які важливі вдосконалення [23].

Microsoft представила набір нових інструментів, які спростять процес розробки:

- MVC розроблений на .NET Core, щоб запропонувати широкую підтримку версії додатків;
- однією з найбільш помітних змін є те, що ASP.NET Core MVC версії 6.0 може безперебійно працювати на всіх основних операційних системах, таких як Windows, Linux та Mac OS. Це спрощує розробників для створення та запуску кросплатформних додатків на всіх цих платформах;
- конфігурація на базі навколишнього середовища готова до хмар;
- ви отримуєте підтримку щодо розробки та використання пакетів NuGet [10].

Маршрутизація рамка ASP.NET Core MVC розробляється у верхній частині маршрутизації ASP.NET Core. Якщо нам доведеться визначити маршрутизацію простими термінами, це надійний компонент відображення на основі URL, який допомагає вам розробити URL-адреси, які можна шукати і розуміти. Це дозволить визначити шаблон назви вашої URL-адреси, який працює належним чином для оптимізації пошукових систем та створення посилань [24].

					ДП.ІПЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

Прив'язка моделі роль прив'язки моделі в рамках ASP.NET Core MVC полягає в перетворенні даних запиту клієнта в об'єкти, контрольовані контролером [15].

Валідація моделі крім моделі зв'язування, ASP.NET Core MVC також допомагає в процесі перевірки. Він прикрашає об'єкт моделі за допомогою атрибутів перевірки анотацій даних. `Validator` перевіряє атрибути перевірки перед публікацією значень на сервері.

Ін'єкційна залежність ще однією життєво важливою особливістю ASP.NET Core MVC є введення залежності. Контролери фреймворку вимагають необхідних служб за допомогою конструкторів, що дозволяє їм слідувати принципу явних залежностей.

Фільтри роль полягає у стисканні наскрізних питань, таких як авторизація та виняткове поведження. Це також допомагає розробникам запускати як перед, так і після публікації індивідуальної логіки обробки методів дій. Він також може виконувати роль атрибутів, коли реалізується для контролерів або дій. До фреймворку додаються різні фільтри як атрибути для побудови фільтрів авторизації MVC [16].

Області допомагають розділити велику веб-програму ASP.NET MVC на менші функціональні групи. Ми можемо визначити область як структуру MVC у програмі. Під час роботи над проектом MVC вам потрібно зберігати логічні компоненти, такі як Model View та Controller, в окремих папках. MVC використовує конвенції іменування для встановлення зв'язку між цими компонентами. Вигідно розлучатися з будь-якими масштабними програмами, такими як електронна комерція, на невеликі функціональні області.

Розробники можуть отримати безліч переваг, все завдяки видатним можливостям ASP.NET Core MVC. Деякі з цих переваг включають: розробка кросплатформних додатків найбільша перевага, яку розробник може отримати з ASP.NET Core MVC, полягає в тому, що це дозволяє йому розробляти кросплатформні програми. Рамка підтримує не тільки Windows, але й інші

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

платформи, такі як Mac та Linux. Розробники отримують підтримку від Visual Code Studio Code для розробки програм у Linux та Mac OS [11].

Підтримуйте модульність та спрощуйте кодування ASP.NET Core MVC є відкритим кодом та гнучким, що означає, що ви можете внести модульність у ландшафт розробки проекту для спрощення процесу кодування. Ви також можете додати бібліотеку та компоненти фреймворку, необхідні для побудови програм. Це дозволяє мати ваші інструменти для управління пакунками NuGet для управління часом виконання. Ви можете додати обидві бібліотеки .NET Core, а також ASP.NET Core Framework відповідно до вимоги проекту.

Підтримка хмарного розгортання модульна архітектура рамки ASP.NET Core MVC дозволяє розгорнути в хмару. Це означає, що ви можете легко розробляти програми за допомогою підтримки ASP.NET Core MVC, яка готова до розгортання у хмарі.

Переваги незалежного хостингу однією з головних відмінностей між ASP.NET Core та ASP.NET є те, що колишня розробка додатків може безперебійно працювати на всіх веб-серверах, за винятком IIS (Internet Information Services). Причина в тому, що ASP.NET Core підтримує крос-платформу і не може зберегти залежність програми лише від IIS. IIS виступав сервером за замовчуванням для розробки додатків ASP.NET [15].

На обгортковій записці завдяки системі ASP.NET Core MVC, що пропонує низку переваг, вона справді стала одним з найпопулярніших варіантів сьогодні для розробників. Він має набагато вдосконалені функції, ніж його попередні версії. Це дозволяє розробникам створювати інтуїтивно зрозумілі додатки з оптимальним користувацьким досвідом. Тож, якщо ви хочете розробити додаток за допомогою ASP.NET Core, зв'яжіться з досвідченими розробниками вже сьогодні [11].

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

### 2.3 Системний аналіз предметної області для Навчальної платформи

Щоб використовувати навчальні курси, ви повинні додавати та редагувати навчальні курси. Простота використання студентами курсу полягає в тому, щоб легко використовувати матеріали курсу та домашні завдання, щоб отримати оцінки та додати практичну роботу.

Лектори мають доступ до освітніх ресурсів, створених лекторами через адміністраторів. Студенти, які беруть участь у курсі, повинні бути об'єднані. Адміністратору необхідно порівняти створений курс із необхідною студентською сукупністю та призначити відповідального лектора. Вивчивши матеріали роздачі, учні повинні виконати домашні завдання. Лектор, відповідальний за дослідницький курс, може отримати доступ до результатів групи. Протягом курсу студенти повинні мати можливість спілкуватися з лектором.

Розглянемо функціональні вимоги: система повинна бути простою у використанні для всіх користувачів; можливість додавання і редагування курсів лектором своєї дисципліни; розмежування можливостей – лектор іншої дисципліни не повинен вміти редагувати курс чужого предмета і виставляти оцінки чужого предмета; можливість переглядати курси та домашні роботи лектором, студентом і адміністратором системи. Проектування архітектури підсистеми в MVC. Вид включає в себе: список курсів; додавання курсу; редагування курсу; інформація про курс.

Контролер включає: список вихідних курсів; додавання курсу; редагування курсу; скасування курсу; висновок про інформацію курсу.

Модель включає в себе: Список курсів: шаблон для відображення списку курсів, відображення назви курсу та посилання на більш детальний опис; взаємодія з контролером "Висновок списку курсів". Додати курс: шаблон для створення курсу; взаємодійте з контролером, доданим курсом, використовуйте його для зберігання вхідних даних користувача; редагування курсу: шаблон для редагування та видалення курсів, взаємодія з контролером для редагування

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

курсів - використовує його для збереження вхідних даних користувачів та контролер для видалення курсів; інформацію про курси: шаблон для відображення детальної інформації про курси та контролер.

Модуль управління: відкрити список курсів: отримати список курсів і перенести його на модуль відображення списку алгоритмів; додати курс: виклик користувача, отримати початкові дані модуля створення курсу (перегляд) та надіслати його модулю створення курсу (вид); редагування курсу: викликається лектором для отримання початкових даних для редагування курсу (перегляду) та даних відредагованого курсу та надсилає їх до модуля для редагування курсу (перегляду); видалення курсу: лектор має право на видалення вибраного курсу із системи та викликає модуль для відображення списку курсів; вихід інформації про курси: викликається лектором, отримує дані про відкритий курс та передає його в модуль для виведення інформації про курс (перегляд). Модуль моделі курсу: верхня модель, що описує структуру курсу; включаючи назву та опис курсу; вхідні дані: верхня модель включає вхідні дані курсу; початкові дані: верхня модель включає початкові дані курсу.

## 2.4 Функції користувачів навчальної платформи

Для успішного здійснення процесів навчання, перш за все, потрібно розібратися з функціями користувачів системи. Технічне завдання визначає три види користувачів: адміністраторів, лекторів і студентів. При організації загальнодоступного інформаційного ресурсу слід також врахувати, що системою може користуватися будь-який користувач мережі Інтернет. У таблиці 2.1 представлені необхідні функції для кожного типу користувачів.

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

Таблиця 2.1 - Функції користувачів Навчальної платформи

Тип користувача	Необхідність авторизації користувача	Функції користувача в системі
Лектор	Так	1. Створення і управління курсами 2. Створення лекцій і методичних матеріалів 3. Розробка контрольних завдань 4. Контроль успішності 5. Створення домашніх робіт до курсів.
Студент	Так	1. Вивчення матеріалів курсу 2. Проходження домашніх робіт завдань 3. Контроль власних результатів
Адміністратор	Так	1. Створення і управління лекторами 2. Адміністрування навчального процесу 3. Управління курсами і інформаційними ресурсами системи 4. Створення та управління ролями 5. Створення користувачів сайту та управління ними 6. Додавання до курсів студентів та лекторів
Користувач мережі Інтернет	Немає	1. Отримання доступу до вільних інформаційних ресурсів системи

## 2.5 Проектування архітектури програмної системи

Використання UML для розробки включає виконання декількох етапів, і властива їм невизначеність у цій роботі призводить до того, що їх потрібно повторити кілька разів, від більш загального опису завдання до детальної обробки на рисунку 2.1.



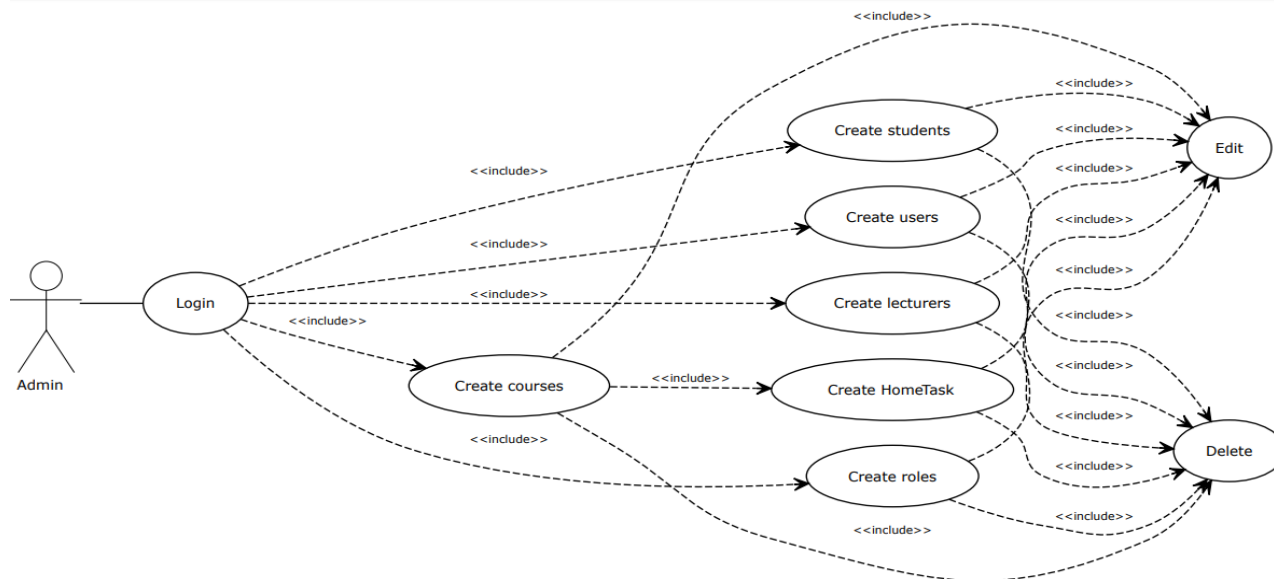


Рисунок 2.1 - Розробка UML-діаграми створення курсу Навчальної платформи

В результаті дослідження предметної області була розроблена модель організації навчання . Архітектура системи представлена на рисунку 2.2.

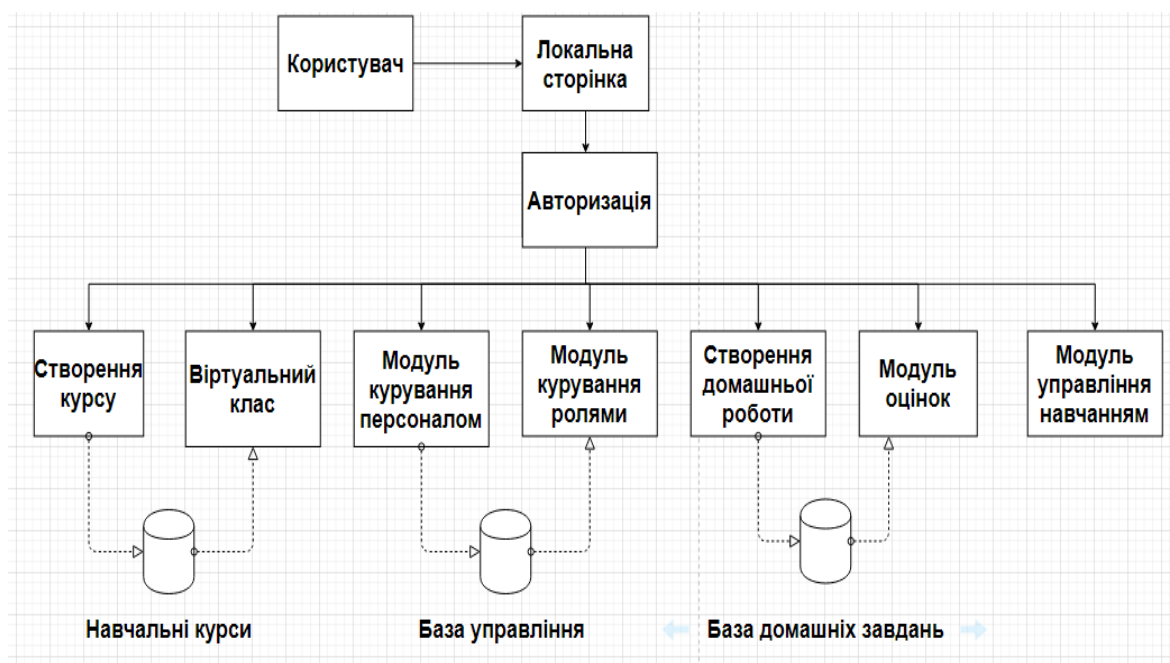


Рисунок 2.2 - Архітектура НП

Зм.	Арк.	№ докум.	Підпис	Дата

Аналогічно, діаграми класів використовуються для представлення статистичної структури системної моделі з точки зору об'єктно-орієнтованого програмування на рисунку 2.3.

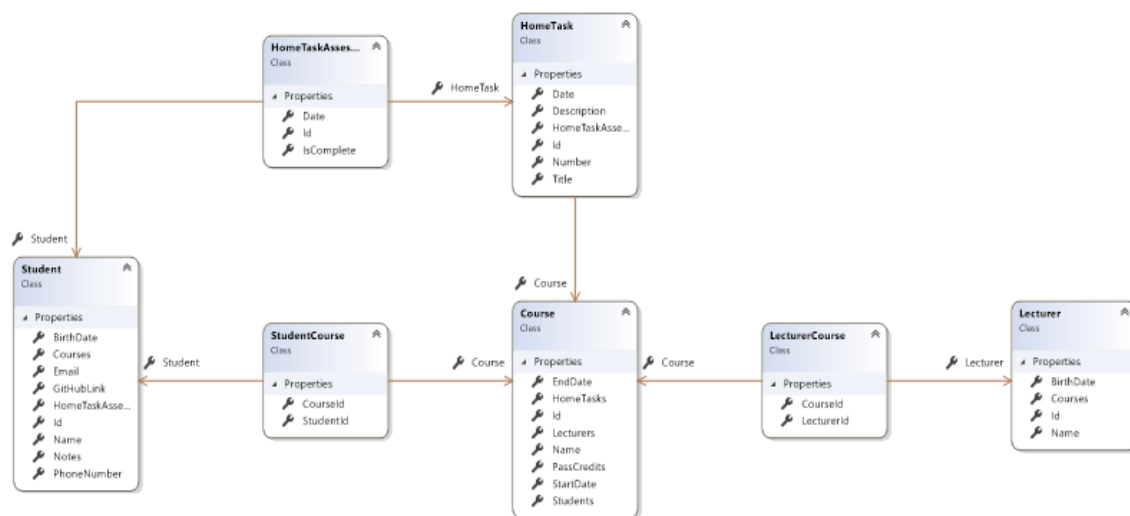


Рисунок 2.3 – Розробка Uml діаграми класів моделей для ПН

## 2.6 Проектування бази даних для Навчальної платформи

Для подання логічної структури інформаційної системи було обрано модель представлення даних "сутність-зв'язок". Базується модель "сутність-зв'язок" на важливій смисловій інформації про реальний світ і має на меті представити дані логічно. Він визначає значення даних на основі співвідношення між даними та іншими даними.

Визначившись з функціональними вимогами веб-системи, ви можете реалізувати базу даних, яка реалізує модель "Entity". При розробці системи була створена база даних за допомогою бази даних MySQL. Визначте основну суть системи проектування. Суб'єкт - це об'єкт, який певним чином можна відрізнити від інших об'єктів. Суб'єкт - це фактично набір атрибутів, що описують атрибути всіх членів даної сукупності сутностей. Набір (діапазон) значень атрибутів називається доменом. Зв'язок - це об'єднання, встановлене між декількома сутностями [12].

					ДП.ПЗ-09.ПЗ		Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			34

Перший етап процесу проектування бази даних називається концептуальним дизайном бази даних. Він включає створення концептуальних моделей даних предметних областей. Модель даних створюється на основі функціональних вимог користувача. Концептуальна конструкція бази даних не залежить від деталей її реалізації, таких як тип обраної бази даних, набір програм, використовувана мова програмування, тип обраної обчислювальної платформи та будь-які інші функції, фізично реалізовані. Концептуальна конструкція - створення концептуального подання бази даних, що включає визначення типів основних суб'єктів, а також існуючих зв'язків та атрибутів між ними. Послідовність етапів проектування концептуальної моделі даних: визначення сутності; визначити взаємозв'язок між сутностями; визначити атрибути сутності; розподіл первинних та резервних ключів.

Для системи бази даних ми вибираємо такі об'єкти: клієнт та відповідь. Кожна сутність повинна містити один або набір атрибутів, щоб однозначно ідентифікувати кожен екземпляр сутності. Цей атрибут називається первинним ключем. Тому необхідно вибрати атрибути, які будуть включені в первинний ключ. Первинний ключ повинен бути обраний таким чином, щоб значення атрибута, що міститься в ньому, міг точно ідентифікувати примірник сутності, а крім того, всі атрибути первинного ключа не повинні мати нульового значення. Значення атрибута первинного ключа не слід змінювати. Якщо значення змінилося, це ще один екземпляр суті. Вибираючи первинний ключ, ви можете додати додатковий атрибут до об'єкта та зробити його ключем. Тому для визначення первинного ключа зазвичай використовується унікальний номер, коли екземпляр сутності додається до бази даних, число може автоматично генеруватися системою. Використання унікальних чисел допомагає індексувати та шукати в базі даних. Первинний ключ, обраний під час створення логічної моделі, може не мати можливості ефективно отримувати доступ до бази даних, тому його необхідно змінити під час проектування фізичної моделі.

Система повинна організовувати механізм зберігання, опису, обміну та пошуку навчальних матеріалів. Всі дані, що вводяться, повинні бути чітко

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

організовані та відображатись як курс. Курс повинен включати лекції та закінчувати оцінюванням студентів. Модуль збору даних повинен бути уніфікованим і підтримувати введення даних з інших баз даних, що поширюються в інтернеті. Тепер ми визначимо атрибути для кожної сутності, а також розмістимо первинні ключі для всіх сутностей. Далі розміщуємо зв'язки між сутностями.

Об'єкти, отримані в результаті інформаційного проектування, зводяться до третьої нормальної форми, оскільки некритичні атрибути не залежать від функції передачі критичних атрибутів. Тому об'єкт складається у формі першого та другого форматів. Перший загальний формат забороняє дублювати стовпці, забороняє декілька стовпців і вимагає вказати первинний ключ для таблиці. Друга стандартна форма вимагає, щоб стовпці ключів таблиці не були вивішені повністю на первинному ключі.

Основним компонентом реляційної бази даних є таблиця. Ця таблиця використовується для побудови та зберігання інформації. У реляційній базі даних кожна комірка таблиці містить значення. Крім того, в одній базі даних є зв'язки між таблицями, і кожне відношення визначає обмін даними таблиці. Діаграма «Основне з'єднання» графічно представляє структуру даних розробленої інформаційної системи на рисунку 2. 4.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

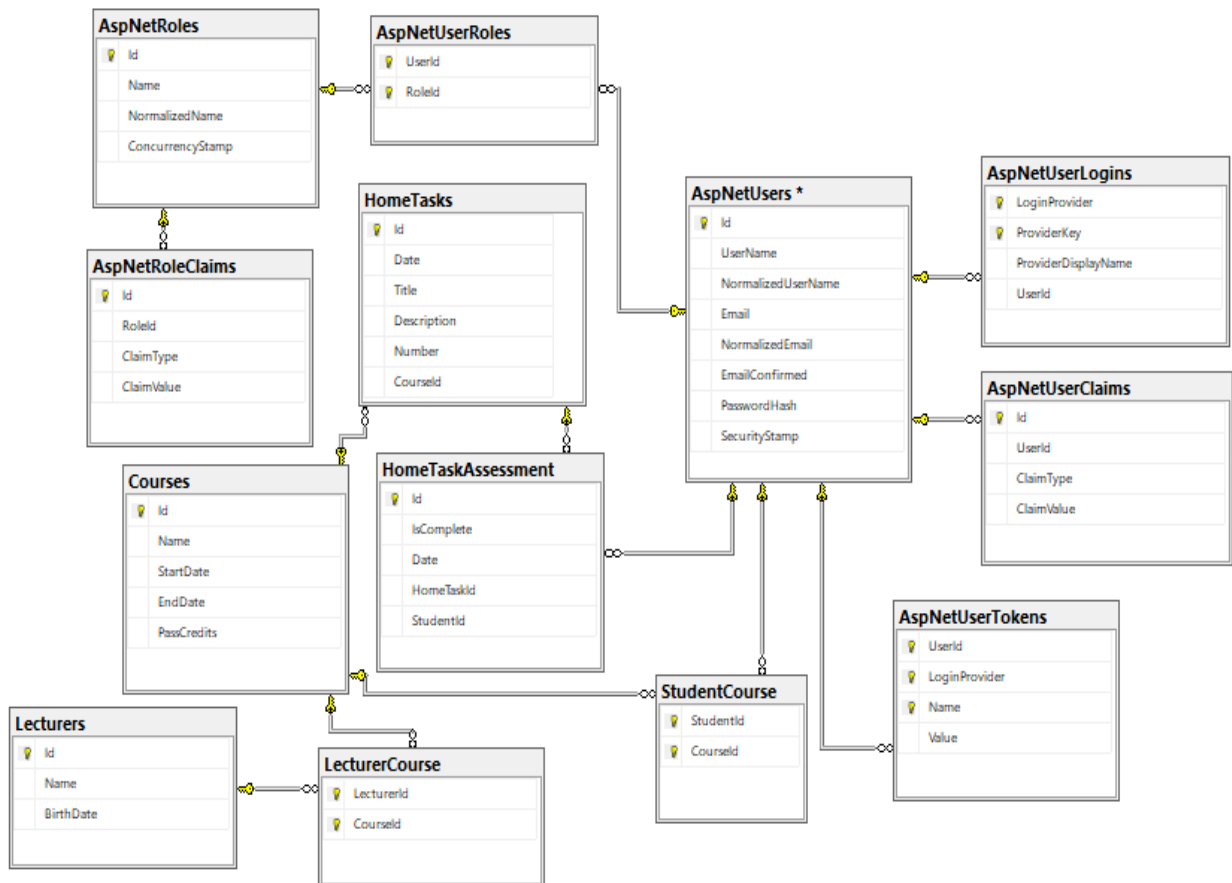


Рисунок 2. 4 – розробка діаграма БД для ПН

При проектуванні системи повинні враховуватися сутності та їх атрибути. Характер предмета включає такі ознаки: номер предмета (первинний ключ), назва предмета, кількість лекторів, ім'я лектора, ім'я студента. Суть лектора містить такі атрибути: Id лектора (первинний ключ), ім'я лектора, дата народження. Суб'єкт студента містить такі атрибути, номер студента (первинний ключ), ім'я студента, номер, групу та додаткову інформацію про студента. Між предметом списку та списком студентів існує взаємозв'язок : учень може мати декілька предметів. Між таблицею дисципліни та таблицею "лектор" існує взаємозв'язок : у вчителя є кілька дисциплін. Після створення концептуальної моделі бази даних ми звернулися до детального проекту підсистеми, що використовується для створення бази дистанційного навчання. Ми відображаємо вибрані об'єкти у таблиці 2.2 а також у таблицях 2.3, 2.4, 2.5, 2.6. можна розглянути детальний опис сутностей.

Зм.	Арк.	№ докум.	Підпис	Дата

Таблиця 2.2 – Список всіх сутностей бази даних Навчальної платформи

Сутність	Переклад
Course	Курс
Lecturer	Лектор
Student	Студент
HomeTask	Завдання

Таблиця 2.3 – Сутність курс (Course)

№	Поле	Атрибут	Тип
1	Id	Є унікальним цілочисельним, яке автоматично генерується це лічильник полів курсів	Int
2	StartDate	Простий атрибут який вказує на початок курсу	DateTime
3	EndDate	Простий атрибут який вказує кінець курсу	DateTime
4	PassCredits	Простий атрибут позначає оцінку	Int

Таблиця 2.3 – Сутність лектор (Lecturer)

№	Поле	Атрибут	Тип
1	Name	Складений атрибут імені лектора	String
2	Id	Лічильник полів лекторів	Int
3	BirthDate	Простий атрибут вказує на дату народження лектора	DateTime

Таблиця 2.4 – Сутність студент (Student)

№	Поле	Атрибут	Тип
1	Name	Складений атрибут імені студента	String
2	Id	Лічильник полів студентів	Int
3	BirthDate	Простий атрибут вказує на дату народження студента	DateTime
4	PhoneNumber	Може реалізовуватися як масив або колекція багатозначних атрибутів тощо	String
5	Email	Для адрес електронної пошти.	String
6	GitHubLink	Простий атрибут для назви групи студента	String
7	Notes	Простий атрибут для додаткової інформації студента	String

Таблиця 2.5 – Сутність домашня робота (HomeTask)

№	Поле	Атрибут	Тип
1	Id	лічильник полів завдань	Int
2	Date	простий атрибут який вказує на кінець завдання	DateTime
3	Title	простий атрибут назва завдання	String
4	Description	простий атрибут позначає максимальну оцінку завдання	String
5	Number	простий атрибут який вказує порядковий номер завдання	Int

Так ми визначили ключові атрибути кожної сутності. Далі розглянемо тип даних кожного атрибута сутності. У таблиці наведено інформацію про типи даних та ключові атрибути.

База даних, створена для впровадження інформаційної системи, може розглядатися як нормалізована, оскільки її таблиця знаходиться у третій нормальній формі. Іншими словами, неклавішні стовпці таблиці не залежать від інших неклавішних стовпців, а лише від первинного ключа. Тому ці таблиці також знаходяться у першій та другій нормальній формі. Загальний формат забороняє дублювати стовпці, забороняє декілька стовпців і вимагає вказати первинний ключ.

## 2.7 Проектування шаблонів підсистеми створення

Для додавання і редагування курсу користувачем представлені наступні моделі. Модель для додавання і редагування курсу від імені адміністратора вказівки дисципліни і дати. Додавання домашньої роботи також залежить від навчального плану предмета і в режимі редагування, старту або неправильній домошній роботі можна замінити. Також можемо вказати лектора. У режимі редагування можна це замінити на інші дані. Шаблон «Додавання навчального курсу» для користувача адміністратор і лекторів представлена на рисунку 2.5.

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

Рисунок 2.5 – Шаблон «створення навчального курсу» для адміністратора та лектора

Модель, що представляє лекторів, які додають та редагують курси, не містить інструкцій для кафедр та дисциплін навчального курсу, оскільки вона автоматично заповнюється на основі особистих даних лектора. Ви можете записати дату. Як репетитори, так і адміністратори користувачів додали лекції, і кількість лекцій залежить від предметного курсу. У режимі редагування, аналогічно користувачеві, адміністратор може замінити одну лекцію на іншу.

Додавання домашньої роботи також залежить від предмету курсу. У режимі редагування ви можете замінити старі або неправильні роботи. Шаблон редагувати навчальний курс для адміністраторів та лекторів відбувається аналогічно до створення курсу .

У режимі перегляду курс містять стовпці зображені, на рисунку 2.6, де всі користувачі можуть переглядати курси, редагувати можуть лишень адміністратори та лектори, студенти можуть приймати курси, та виконувати домашні завдання.



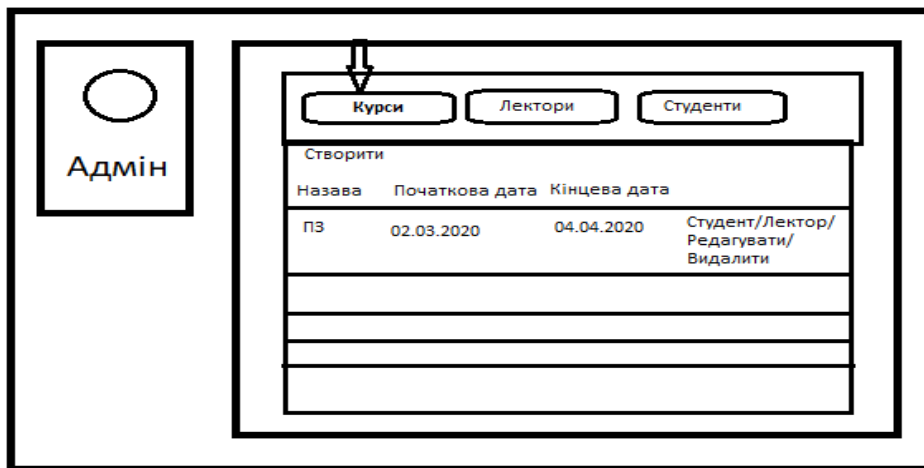


Рисунок 2.6 – Шаблон списку курсу для користувачів

## 2.8 Декомпозиція моделі для Навчальної платформи

При проектуванні складної системи її поділяють на кілька частин, а потім кожен частину розглядають окремо. Існує два способи поділу цього на підсистеми: поділ структури (або функцій) та декомпозиція об'єкта (компонента).

При функціональному розкладі програмної системи її структуру можна описати блок-схемою, вузли блок-схеми є функціями, а зв'язки між вузлами описують рух даних. У системі дистанційного навчання її можна розділити на дві підсистеми: підсистема накопичення освітніх даних та підсистема навчання. У технології програмування процеси, що відбуваються в таких системах, називаються відповідно Front-end і Back-end частинами - це загальні терміни, які відображають початковий та кінцевий стан процесу. Front-end відповідає за отримання будь-якої форми введення (вхідної інформації) від користувача та обробляє отриману інформацію у формі, яка може бути використана Back-end. Фронт - це інтерфейс між користувачем і апаратною частиною [22].

Поєднання цих двох підсистем зможе забезпечити якісне навчання. Подальший поділ підсистем призведе до модульної структури, в якій кожна дія обробляється окремим модулем. Це полегшить оновлення системи. Потрібно

лише відремонтувати модуль, щоб додати нові функції або полагодити дефекти системи.

Поряд з функціями, декомпозиція об'єкта може бути застосована до системи. Як об'єкт доцільно використовувати користувачів системи. Кожен користувач має опис своїх атрибутів та методів організації функцій. Загальна схема зареєстрованих користувачів системи показана на рисунку 2.7.

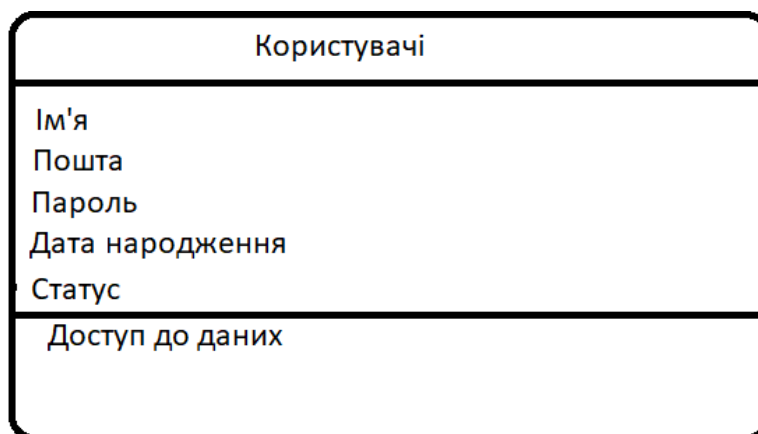


Рисунок 2.7 - Загальна діаграма користувача системи ПН

З малюнка видно, що будь-який користувач має такі атрибути, як ім'я, пошту пароль та дата народження. Атрибут статусу визначає приналежність користувача до таких типів: адміністратор, лектор, студент. Кожний користувач має доступ до даних сайту.

Нова діаграма користувачів додає нові атрибути та методи, необхідні для організаційних функцій. Принципова схема адміністратора показана на рисунку 2.8.



Рисунок 2.8 - Діаграма адміністратора в НП

Крім того, додаються курси редагування, створення та редагування домашніх завдань. Для адміністраторів курсів існує спосіб отримати результати від груп курсів. Крім майна, що визначає членство в організації, студенти також додали методи отримання матеріалів роздачі та домашніх завдань на рисунку 2.9.



Рисунок 2.9 - Діаграма студентів В НП

За допомогою отриманої структури а так же розроблених модулів можна приступити до розробки Навчальної платформи.

## 2.9 Розробка авторизації на основі ролі в ASP.NET Core

Створений сертифікат може належати до однієї або декількох ролей. Наприклад, Трейсі може належати до ролей адміністратора та користувача, тоді як Скотт може належати лише ролі користувача. Створення та управління цими ролями залежить від резервного зберігання процесу авторизації. Роль надається розробнику за допомогою методу `IsInRole` в класі `ClaimsPrincipal`.

У простому випадку аутентифікації ми зазвичай використовуємо ім'я користувача та пароль для входу та виходячи з цього, ми надаємо доступ до програми, але в цьому випадку користувач може отримати доступ до всіх ресурсів програми. Використовуючи процес авторизації на основі політик та ролей, ми можемо надати користувачеві доступ до певної області застосування на основі ролі / політики користувача.

У цій програмі ми автентифікуємо ролі адміністратора, користувача, лектора та студента, але в пз можа створити нові ролі відповідно до вимог адміністратора.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

## 3 РОЗРОБКА ТА ОПИС НАВЧАЛЬНОЇ ПЛАТФОРМИ НА ОСНОВІ ТЕХНОЛОГІЙ .NET

### 3.1 Розробка інтерфейсу в Навчальній платформі

Важливим елементом будь-якої системи, що має безпосередній контакт з користувачем, є інтерфейс. Інтерфейс - це сукупність засобів і методів взаємодії між елементами системи. Цей термін використовується практично у всіх областях науки і техніки. Його значення відноситься до будь-якого сполученню взаємодіючих сутностей.

Програмний компонент системи дистанційного навчання є сайт, який має графічний інтерфейс. При проектуванні графічних інтерфейсів вводиться поняття «юзабіліті». Юзабіліті (usability - дослівно «можливість використання», «корисність») - поняття, що означає підсумковий рівень зручності інтерфейсу для використання в заявлених цілях. На сьогоднішній день проведено безліч досліджень в області розробок.

Тому для забезпечення зручності користувача кожен сайт повинен містити наступні елементи. Блок ідентифікації сайту є частиною графічного інтерфейсу, який дозволяє відвідувачу чітко визначити, що представляє ресурс, який він використовує. Представляє логотип сайту, його назву та можливий короткий опис. Гіперпосилання - частина документа, яка вказує на інший елемент (текст, заголовок, коментар, зображення) у самому документі, інший об'єкт (Файли, каталоги, програми) розташовані на локальній комп'ютері або комп'ютерній мережі або на елементах цього об'єкта. Навігація - елементи інтерфейсу, які дозволяють користувачам переходити до певних структурних частин (сторінок) веб-сайту. Він містить головне меню та додаткове меню і являє собою набір гіперпосилань. Контент - це вміст самого веб-сайту. Це сукупність текстової та графічної інформації. Використовуючи ці поняття, ви можете описати зовнішній вигляд системи, що відповідає вимогам щодо простоти використання. Насправді, проектування графічного інтерфейсу для веб-сайту - це розвиток організації та

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

поява всіх необхідних елементів на сторінці. Сучасні досягнення ергономіки дозволили створити так звані «карти кліків». Карта точно показує, на що звертають увагу відвідувачі сайту. Наводить приклад "кнопки карти" пошукової системи "Google" [18]. На рисунку 3.1.

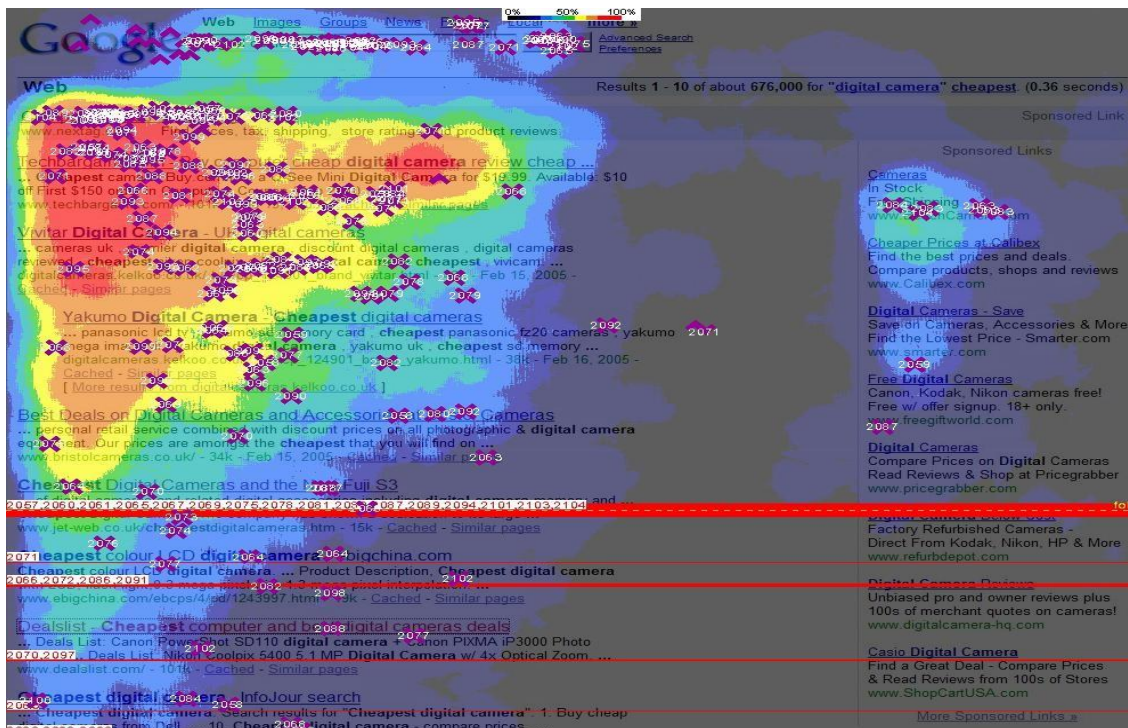


Рисунок 3.1. «Карта кліків» сайт Google.

Трикутник, утворений на місці, де відвідувачі сайту "кліків", найчастіше фахівцями називають "золотим трикутником". Потрібно розмістити головний структурний елемент у цьому місці на сторінці сайту, щоб привернути увагу користувача.

### 3.2 Розробка бази даних для Навчальної платформи

Для зберігання інформації в системі дистанційного навчання передбачається використання баз даних MySQL. Потрібно розробити структуру бази даних, щоб почати процес програмування. База даних має ім'я «AspDemo»

структура і докладний опис сутностей і назви полів так же опис полів наведений в таблиці 3.1.

Таблиця 3.1 – Розроблена база даних для НП

Сутності	Опис сутностей	Назва полів	Опис полів
Users	Інформація про користувача	Id	Ідентифікатор користувача. Значення є унікальним і однозначно визначає користувача.
		Email	Для всіх типів користувачів, крім адміністратора, значенням цього поля є контактна адреса електронної пошти. Так само використовується як логін в процесі авторизації. Значення є унікальним і однозначно визначає користувача.
		Password	Пароль користувача. Використовується в процесі авторизації.
Roles	Інформація про ролі користувачів	Id	Ідентифікатор ролей
		Name	Назва ролей
HomeTasks	Інформація про домашню роботу	Id	Ідентифікатор
		Date	Дата завершення
		Description	Оцінка домашньої роботи
		Title	Номер домашньої роботи
		CourseId	Назва факультету
HomeTaskAssessment	Оцінка домашньої роботи у студентів	Id	Ідентифікатор оцінювання
		IsComplete	Мітка завершення
		Date	Дата закінчення
		HomeTaskId	Ідентифікатор домашньої роботи
		StudentId	Ідентифікатор студента

Courses	Інформаці Про курс	Id	Ідентифікатор курсу
		Name	Назва курсу
		StartDate	Початкова дата курсу
		EndDate	Кінцева дата курсу
		PassCredits	Оцінка за курс
Lecturer	Інформація про лекторів	Id	Ідентифікатор лектора
		Name	Назва лектора
		Date	Дата народження
Student	Інформація про студентів	Id	Ідентифікатор студента
		Name	Прізвище і ім'я студента
		BirthDate	Дата народження
		PhoneNumbet	Номер телефону
		GitHubLink	Назва групи
		Notes	Додаткова інформація про студента
StudentCourse	Інрмація зв'язок між курсом і студентом	StudentId	Ідентифікатор студента
		CourseId	Ідентифікатор курсу
LecturerCourse	Інрмація зв'язок між курсом і лектором	LecturerId	Ідентифікатор лектора
		CourseID	Ідентифікатор курсу

Ви також повинні скасувати пряме з'єднання деяких таблиць. Це робиться за допомогою цих ключів, які використовуються для підтримки цілісності структури бази даних.

Використовуючи таблиці цієї бази даних, ви можете успішно виконувати навчальний процес в НП.

### 3.3 Розробка головного модуля Навчальної платформи

Головний модуль системи дистанційного навчання являє собою сукупність web-додатків, за допомогою якого здійснюється підтримка основних функціональних можливостей: графічний інтерфейс; авторизація користувача; отримання інформації про користувача; настройка користувача; - вихід із системи.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48



Щоб забезпечити вимога подальшої модернізації системи, розумно використовувати об'єктно-орієнтований підхід до програмування моделей. Всі об'єкти в системі повинні представляти собою екземпляри класів. Всі функції системи будуть методами в певних класах.

### 3.4 Графічний опис Навчальної платформи

Для того, щоб користуватися платформою для навчання потрібно авторизуватися всім користувачам . Можемо розглянути авторизацію на рисунку 3.2.

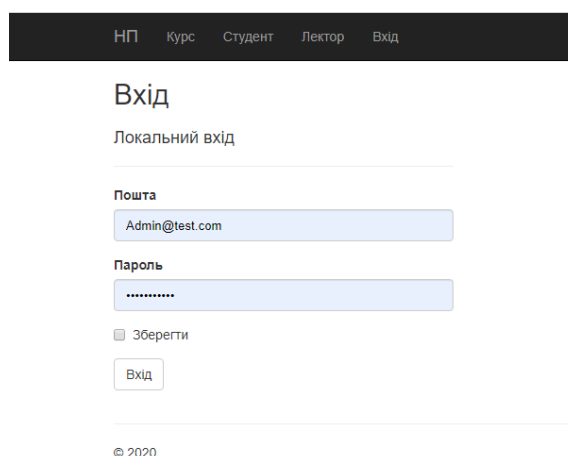


Рисунок 3.2 – графічне зображення авторизації в НП

На цій сторінці ми бачимо що саме потрібно для авторизації користувача. Перше, що потрібно зробити – це ввести пошту і пароль, які вже є в базі даних. Якщо їх ввести неправильно, то авторизація не відбудеться. Другий пункт, збереження пошти і паролю, є необов'язковим. В кінці потрібно клікнути на кнопку «вхід» і, якщо все правильно, тоді акаунт авторизований .

Після проходження авторизації будь-якому користувачу з'являється доступ до перегляду курсів и домашнього завдання також є можливість переглянутися список лекторів та студентів. «Рисунок 3.3 - Графічне представлення головної сторінки в НП».

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

НП	Курс	Студент	Лектор	Вихід
Створити				
Назва	Старт	Кінець	Оцінка	Домашня робота
Хімія	02.03.2020	02.04.2020	2	2
Біологія	12.01.2020	11.01.2020	3	0
Фізика	20.01.2020	02.03.2020	4	0
Філософія	13.05.2020	13.06.2020	50	0
Анатомія	26.03.2020	24.04.2020	100	0

© 2020

Рисунок 3.3 – Графічне представлення головної сторінки в НП

Після входу адміністратора є пару функцій одна з них – це створення ролей для користувачів і їх розподіл. Розташована у головному меню в пункті “Створення ролей”. Функція зображена на рисунку 3.4. Цей функціонал потрібний для того, щоб можна було керувати правильною поведінкою навчальної платформи.

НП	Курс	Студент	Лектор	СтворенняКористувача	Користувачі	Створення Ролі	Вихід
Створити ролі							
Створити							
Ролі							
<input type="text"/>							
<input type="button" value="Створити"/>							

© 2020

Рисунок 3.4 - Графічне представлення створення ролей в НП

На цій сторінці ми можемо бачити, що створення ролі відбувається в один крок. Записати назву ролі і створити її. За замовчуванням стоїть роль “admin”. Після того, як ми створили роль, ми її можемо призначити будь-якому користувачеві. Цю функцію можна розглянути на рисунку 3.5.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

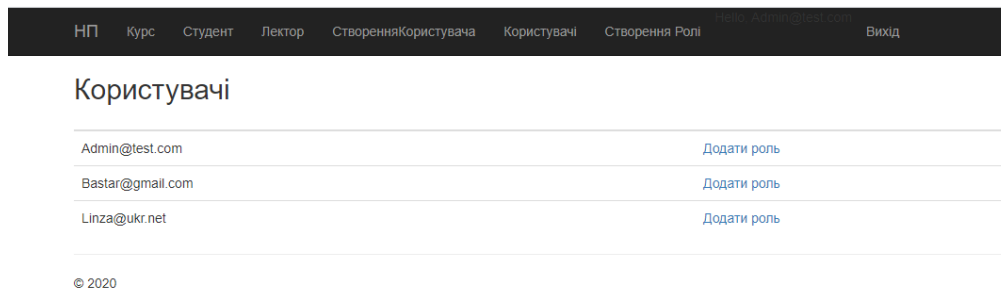


Рисунок 3.5 - Графічне представлення список користувачів в НП

Після чого адміністратор може додавати ролі будь-якому користувачеві на рисунку 3.6.

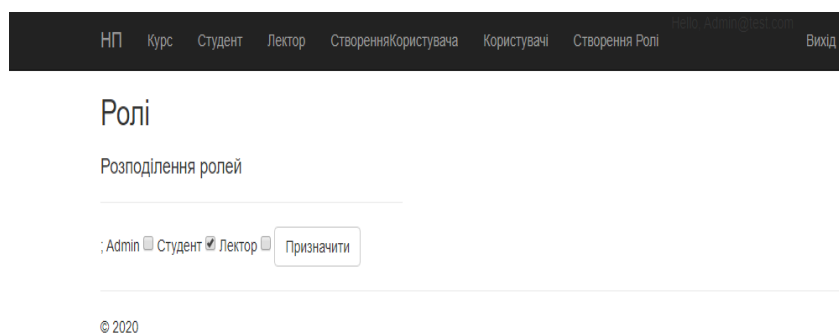


Рисунок 3.6 - Графічне представлення додавання ролей користувачу в НП

Як ми бачимо на сторінці після вибору пункту “додати роль” одному із користувачів є можливість призначати всі створені ролі а також знати їх. А також в головному меню в пункті “СтворенняКористувачів”є ще одна функція – це створення користувачів. Ми її можемо розглянути на рисунку 3.7.

[НП](#)
[Курс](#)
[Студент](#)
[Лектор](#)
[Створення Користувача](#)
[Користувачі](#)
[Створення Ролі](#)
[Вихід](#)

### Створити користувача

Локальний вхід

Пошта

Пароль

Створити

© 2020

Рисунок 3.7 – Графічне представлення створення користувачів в НП

Створення відбувається аналогічно до авторизації: введення пошти та паролю. Після чого створений акаунт додається до бази даних. І адміністратор має можливість ним керувати як завгодно. Також адміністратор створює лекторів для керування і оцінювання курсів на рисунку 3.8.

[НП](#)
[Курс](#)
[Студент](#)
[Лектор](#)
[Створення Користувача](#)
[Користувачі](#)
[Створення Ролі](#)
[Вихід](#)

### Лектори

Ім'я

Дата

Зберегти

© 2020

Рисунок 3.8 - Графічне представлення створення лектора в НП

Для створення лектора адміністратору потрібно вибрати пункт “Лектори”, після чого натиснути “Створити”. Тоді з’явиться 2 пункти, які необхідно заповнити ім’ям і датою. Після того, як ми збережемо, дані буде відтворено так, як показано на рисунку 3.9.

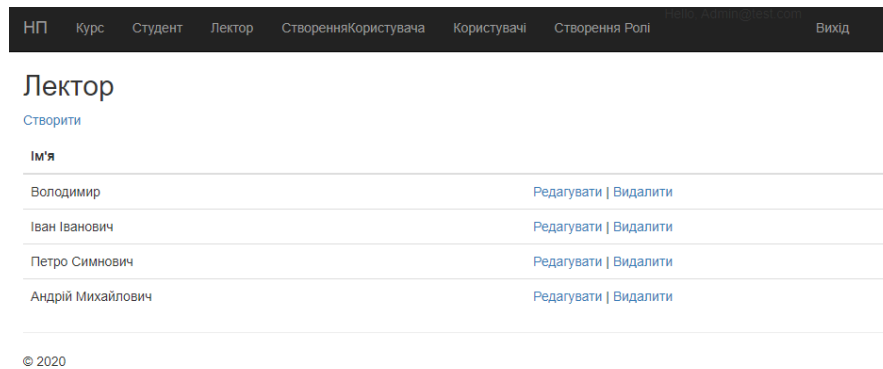


Рисунок 3.9 - Графічне представлення список лекторів в НП

Як можна бачити, дані всіх лекторів можна редагувати та видаляти. Редагування лектора відбувається аналогічно до створення. Ще однією з основних функцій є створення студентів у головному меню пункт “Студент”, яку можна бачити на рисунку 3.10.

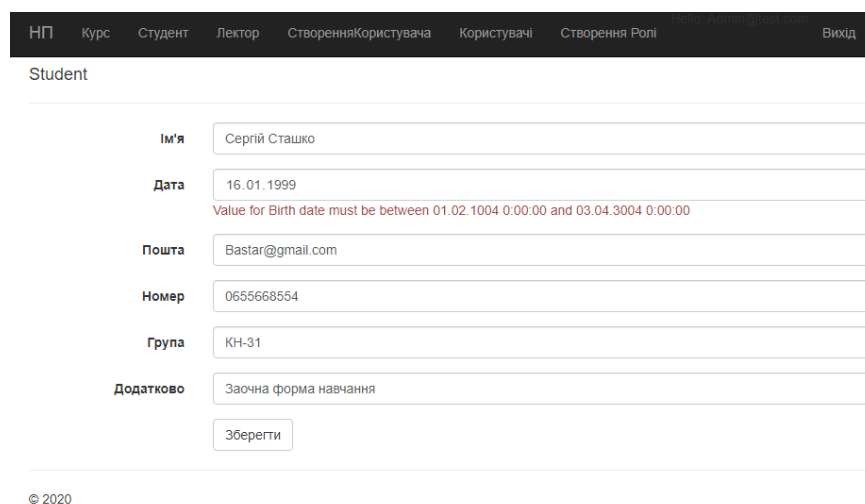


Рисунок 3.10 - Графічне представлення створення студента в НП

При створенні студента потрібно заповнити шість пунктів, три з яких є обов'язковими: ім'я, дата, пошта. Три наступні є не обов'язкові, але заповнити їх потрібно для зручної роботи навчальної платформи. Після того, як ми створимо студента, можна переглянути список, де кожного студента можна редагувати і видаляти ця функція зображена на рисунку 3.11.

НП	Курс	Студент	Лектор	СтворенняКористувача	Користувачі	Створення Ролі	Вихід
Створити							
Ім'я	Дата	Пошта	Телефон	Група	Додатково		
Евгеній Луцьв	06.01.1999	Genalyciv@gmail.com	0674969423		<a href="#">Редагувати</a>   <a href="#">Видалити</a>		
Сергій Сташко	16.01.1999	Bastar@gmail.com	0655668554	КН-31	Заочна форма навчання		<a href="#">Редагувати</a>   <a href="#">Видалити</a>

© 2020

Рисунок 3.11 - Графічне представлення списку студентів НП

Тут можна бачити всі поля, які ми заповняли при створенні. Також виводиться список студентів, де їх можна редагувати і видаляти. Важливим кроком при організації навчання в середовищі є створення курсів. Це обов'язок лекторів. Для того щоб приступити до розробки курсу, необхідно вибрати пункт «Курс» в головному меню. Завантажена після цього сторінка дозволяє управляти вже створеними курсами, а так само розробляти нові на рисунку 3.12.

НП	Курс	Студент	Лектор	СтворенняКористувача	Користувачі	Створення Ролі	Вихід
Створити							
Назва	Старт	Кінець	Оцінка	Домашня робота			
Хімія	02.03.2020	02.04.2020	2	2 <a href="#">Лектори</a>   <a href="#">Студенти</a>   <a href="#">Редагувати</a>   <a href="#">Видалити</a>			
Біологія	12.01.2020	11.01.2020	3	0 <a href="#">Лектори</a>   <a href="#">Студенти</a>   <a href="#">Редагувати</a>   <a href="#">Видалити</a>			
Фізика	20.01.2020	02.03.2020	4	0 <a href="#">Лектори</a>   <a href="#">Студенти</a>   <a href="#">Редагувати</a>   <a href="#">Видалити</a>			
Філософія	13.05.2020	13.06.2020	50	0 <a href="#">Лектори</a>   <a href="#">Студенти</a>   <a href="#">Редагувати</a>   <a href="#">Видалити</a>			
Анатомія	26.03.2020	24.04.2020	100	0 <a href="#">Лектори</a>   <a href="#">Студенти</a>   <a href="#">Редагувати</a>   <a href="#">Видалити</a>			

© 2020

Рисунок 3.12 - Графічне представлення списку курсів в НП

На цій сторінці ми можемо додавати лекторів до певного курсу кнопкою “Лектори” і студентів кнопкою “Студенти” а також редагувати і видаляти. Головною можливістю є створення курсу за допомогою кнопки “Створення” та домашніми роботами. Перше, що ми розглянемо, це створення курсу на рисунку 3.13.

НП Курс Студент Лектор СтворенняКористувача Користувачі Створення Ролі Вихід

Курс

Назва

Старт

Кінець

Оцінка

Назад

© 2020

Рисунок 3.13 - Графічне представлення створення курсів в НП

На цій сторінці потрібно ввести назву курсу, початкову і кінцеву дату, а також вказати максимальну оцінку, яку можна отримати з курсу. Доту можна вибирати в на календарі або вводити в ручку. Після створення курсу з'являється додаткова функція при натисканні кнопки "Створення домашньої роботи". На рисунку 3.14.

НП Курс Студент Лектор СтворенняКористувача Користувачі Створення Ролі Вихід

Домашня робота

Назва

Номер

Оцінка

Дата

© 2020

Рисунок 3.14 - Графічне представлення створення роботи в НП

Тут можна задати номер домашнього завдання і дату, до якої потрібно його виконати, а також максимальну оцінку за нього. Список домашніх робіт можна розглянути на рисунку 3.15.

[НП](#)
[Курс](#)
[Студент](#)
[Лектор](#)
[Створення Користувача](#)
[Користувачі](#)
[Створення Ролі](#)
[Вихід](#)

Course

**Назва**

**Старт**

**Кінець**

**Оцінка**

[Створити домашню роботу](#)

Хімія(Водень)	02.03.2020	2	<a href="#">Оцінювання</a>   <a href="#">Редагувати</a>   <a href="#">Видалити</a>
Хімія(Речовини)	02.01.2020	2	<a href="#">Оцінювання</a>   <a href="#">Редагувати</a>   <a href="#">Видалити</a>

[Назад](#)

Рисунок 3.15 - Графічне представлення створення роботи в НП

Всі домашні роботи можна редагувати, видалити, а також є можливість оцінювати їх . Як відбувається оцінювання можна розглянути на рисунку 3.16.

[НП](#)
[Курс](#)
[Студент](#)
[Лектор](#)
[Створення Користувача](#)
[Користувачі](#)
[Створення Ролі](#)
[Вихід](#)

Оцінювання

Хімія(Водень)

2

02.03.2020 0:00:00

**Студент**

**Перехід**

© 2020

Рисунок 3.16 - Графічне представлення оцінювання в НП

Кожну домашню роботу оцінює лектор, а також помічає, хто пройшов це завдання, а хто ні, і зберігає результат . Після цього пункту можна вийти на те, як додаються студенти і лектори до курсу. На рисунку 3.17, можна розглянути, як додавати лектора до курсу в НП .



НП Курс Студент Лектор СтворенняКористувача Користувачі Створення Ролі Вихід

Додати

Хімія

02.03.2020 0:00:00

02.04.2020 0:00:00

Лектора	Додати
Володимир Іванович	<input checked="" type="checkbox"/>
Іван Іванович	<input checked="" type="checkbox"/>
Петро Симнович	<input type="checkbox"/>
Андрій Михайлович	<input type="checkbox"/>

Зберегти

© 2020

Рисунок 3.17 - Графічне представлення додавання лектора до курсу

Бачимо на зображенні назву, дату початку і кінця курсу, а також список лекторів, яких ми можемо додати до нього, і це все зберегти. ілюстрація на рисунку 3.18, функція додавання студентів до курсу аналогічна до функції додавання лекторів.

НП Курс Студент Лектор СтворенняКористувача Користувачі Створення Ролі Вихід

Додати

Хімія

02.03.2020 0:00:00

02.04.2020 0:00:00

Студента	Додати
Євгеній Луцїв	<input checked="" type="checkbox"/>
Сергій Сташко	<input type="checkbox"/>

Зберегти

© 2020

Рисунок 3.18 - Графічне представлення додавання студентів до курсу

### 3.5 Розробка головного програмного модуля в навчальній платформі

Налаштування стилю НП створення меню сайту. Відкривши «Views-Shared-Layout.cshtml» будемо бачити додані такі пункти меню студенти, курси та лектори. Фрагмент коду зображений на рисунку 3.19, весь код наведений в додатку А.

```
<li><a asp-controller="Course" asp-action="Courses">Курс</a></li>
<li><a asp-controller="Student" asp-action="Students">Студент</a></li>
<li><a asp-controller="Lecturer" asp-action="Lecturers">Лектор</a></li>
@if (!User.Identity.IsAuthenticated)
{
    <li><a asp-controller="Security" asp-action="Login">Вхід</a></li>
}
else
{
    if (User.IsInRole("Admin"))
    {
        <li><a asp-controller="Security" asp-action="CreateUser">Створення Користувача</a></li>
        <li><a asp-controller="Security" asp-action="Users">Користувачі</a></li>
        <li><a asp-controller="Security" asp-action="CreateRole">Створення Ролі</a></li>
    }
    <li><p>Hello, @User.Identity.Name</p></li>
    <li><a asp-controller="Security" asp-action="Logout">Вихід</a></li>
}
```

Рисунок 3.19 – Фрагмент коду пункту меню .

Розроблена модель даних клас об'єктів на рисунку 3.20, для НП. Починається з чотирьох сутностей (студент, оцінювання, завдання, курс).

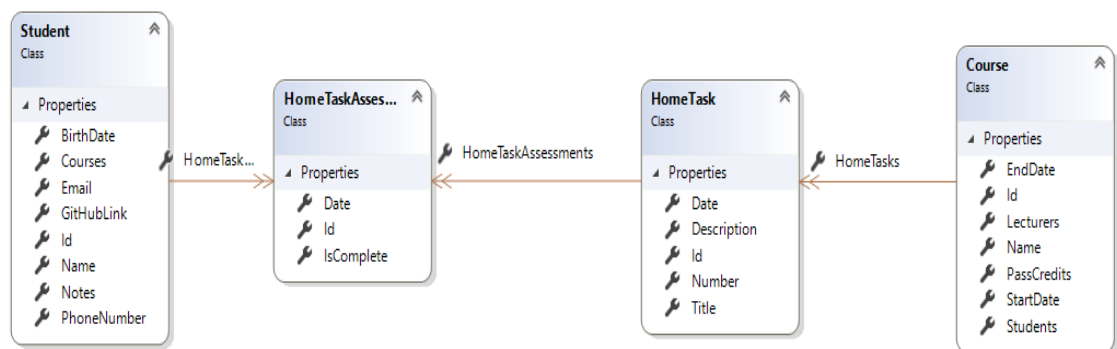


Рисунок 3.20 – Модель даних (студент, завдання та курс)

Існує взаємозв'язок між багатьма Student і HomeTaskAssessment сутностями, і існує відносини один до багатьох між сутностями HomeTask та Course сутностями. Іншими словами, студент може бути зарахований на будь-яку кількість курсів, а на курсі може бути включена будь-яка кількість студентів. Далі показано створені класи для кожної з цих сутностей. Студентська сутність на рисунку 3.21.

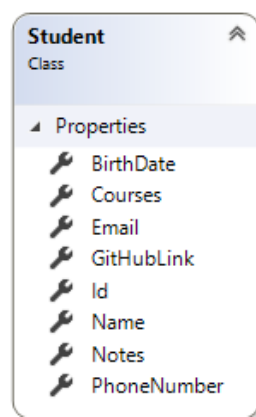


Рисунок 3.21 – Клас Student

У папці моделі створений файл класу під назвою Student.cs фрагмент коду зображений на рисунку 3.22, весь код виведений в додатку Б.

```
28 references
public class Student
{
    [Required(AllowEmptyStrings = false)]

    7 references | 0 exceptions
    public string Name { get; set; }

    7 references | 0 exceptions
    public int Id { get; set; }

    [Display(Name = "Birth date")]
    [DataType(DataType.Date)]
    [Range(typeof(DateTime), "1/2/1004", "3/4/3004",
        ErrorMessage = "Value for {0} must be between {1} and {2}")]
    4 references | 0 exceptions
    public DateTime BirthDate { get; set; }

    [Display(Name = "Phone number")]
    [DataType(DataType.PhoneNumber)]
    [Required(AllowEmptyStrings = false)]
    4 references | 0 exceptions
    public string PhoneNumber { get; set; }
}
```

Рисунок 3.22 – Фрагмент коду сутності Student

ID властивість стане стовпчиком первинного ключа таблиці бази даних , яка відповідає цьому класу. За замовчуванням Entity Framework інтерпретує властивість, яка названа ID або NameID як основний ключ.

Розроблено контекст бази даних основним класом, який координує функціональність Entity Framework для даної моделі даних, є контекстний клас бази даних. Ви створюєте цей клас, виходячи з Microsoft. Entity Framework Core. DbContext класу. У коді вказано, які об'єкти включені в модель даних. Також налаштована поведінка Entity Framework. У папці проекту створена папка з назвою DataAccess.У папці DataAccess створений файл класу під назвою UniversityContext.cs фрагмент коду зображено на рисунку 3.23, повний код розробки знаходиться в додатку Г.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

```

protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    base.OnModelCreating(modelBuilder);
    modelBuilder.Entity<LecturerCourse>().HasKey(k => new { k.LecturerId, k.CourseId });
    modelBuilder.Entity<StudentCourse>().HasKey(k => new { k.CourseId, k.StudentId });
}

0 references | 0 exceptions
public DbSet<Student> Students { get; set; }

0 references | 0 exceptions
public DbSet<Course> Courses { get; set; }

0 references | 0 exceptions
public DbSet<HomeTask> HomeTasks { get; set; }

0 references | 0 exceptions
public DbSet<Lecturer> Lecturers { get; set; }

```

Рисунок 3.23 – Фрагмент коду. Контекст бази даних

Цей код створює DbSet властивість для кожного набору об'єктів. У термінології Entity Framework сукупність об'єктів зазвичай відповідає таблиці бази даних, а сутність відповідає рядку в таблиці.

Коли база даних створена, EF створює таблиці, у яких імена такі ж, як DbSet імена властивостей. Назви властивостей колекцій зазвичай множинні (Students , а не Student). У файлі appsettings.json Доданий рядок з'єднання, як показано на рисунку 3.24.

```

{
  "DefaultConnectionString": "Server=(localdb)\\MSSQLLocalDB;Database=AspDemo;Trusted_Connection=True;",
  "Logging": {
    "IncludeScopes": false,
    "LogLevel": {
      "Default": "Warning"
    }
  }
}

```

Рисунок 3.24 - Підключення SQL Server LocalDB

Рядок підключення вказує базу даних Localdb SQL Server. Localdb - це легка версія двигуна бази даних SQL Server і призначена для розробки додатків,

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

а не виробничого використання. Localdb запускається на вимогу і працює в режимі користувача, тому немає складної конфігурації. За замовчуванням Localdb створює файли бази даних.

Розроблений метод створення курсу у класі CourseController.cs модифікуйте Create метод HttpPost зображений на рисунку 3.25, весь код можна розглянути в додатку В.

```
[HttpPost]
2 references | 0 requests | 0 exceptions
public IActionResult Create(Course courseParameter)
{
    if (courseParameter == null)
    {
        return this.BadRequest();
    }

    if (!ModelState.IsValid)
    {
        ViewData["action"] = nameof(this.Create);

        return this.View("Edit", courseParameter);
    }
    this.courseService.CreateCourse(courseParameter);
    return this.RedirectToAction(nameof(Courses));
}
```

Рисунок 3.25 – Фрагмент коду створення курсу в НП

Цей код створений за допомогою в'язучої моделі ASP.NET Core MVC, а потім зберігає зміни в базі даних. Прив'язка моделі посилається на функціональність ASP.NET Core MVC, що полегшує вам роботу з даними, поданими формою; модельне палітурне перетворення розміщених значень форми у типи CLR та передає їх методу дії в параметри. У цьому випадку, палітурку моделі створює, використовуючи значення властивостей із колекції Form. Введення від користувача не встановлює значення ідентифікатора.

Розроблений метод редагування курсу для навчальної сторінки у CourseController.cs Edit метод HttpGet (який не має HttpPost атрибута) використовує CreateCourse метод для отримання вибраного об'єкту Course, як ви

бачили в Details методі. Вам не потрібно змінювати цей метод. Рекомендований HttpPost Код редагування: читання та оновлення розроблений метод дій HttpPost Edit зображений на рисунку 3.26, фрагментом коду. Повний код можна розглянути в додатку В.

```
[HttpGet]
1 reference | 0 requests | 0 exceptions
public IActionResult Edit(int id)
{
    Course course = this.courseService.GetCourse(id);
    if (course == null)
    {
        return this.NotFound();
    }

    ViewData["action"] = nameof(this.Edit);

    return this.View(course);
}

[HttpPost]
[Authorize(Roles = "Admin")]
1 reference | 0 requests | 0 exceptions
public IActionResult Edit(Course courseParameter)
{
    if (courseParameter == null)
```

Рисунок 3.26 – Фрагмент коду редагування курсу в НП

Розробка реалізує найкращу практику безпеки для запобігання перенапруження. Даний код зчитує наявну сутність та закликає UpdateCourse оновити поля в отриманому об'єкті на основі даних курсу у розміщених даних форми . Автоматичне відстеження змін Entity Framework. Коли викликається, Entity Framework створюються оператори SQL для оновлення рядка бази даних.

Розроблений метод видаляти курс в НП у CourseController.cs код шаблону для Delete методу HttpGet використовує DeleteCourse метод для отримання вибраного об'єкту IP Course фрагмент коду зображено на рисунку 3.27, повний код у додатку В.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

```
[Authorize(Roles = "Admin")]  
0 references | 0 requests | 0 exceptions  
public IActionResult Delete(int id)  
{  
    this.courseService.DeleteCourse(id);  
    return RedirectToAction("Courses");  
}
```

Рисунок 3.27 – Фрегмент коду метод видалення курсу в НП

Цей код приймає необов'язковий параметр, який вказує, чи був викликаний метод після відмови збереження змін. Цей параметр помилковий, коли Delete методHttpGet викликається без попереднього збою. Коли він викликається методомHttpPost Deleteу відповідь на помилку оновлення бази даних, параметр є істинним, і повідомлення про помилку передається в подання. Повний код можна розглянути в додатку В.

### 3.6 Організація компонентів в Навчальній платформі

За замовчуванням структура папки програми ASP.NET Core включає контролери, представлення даних та звичайну модель перегляду. Код клієнта, який підтримує такі структури на стороні сервера, зазвичай зберігається окремо в папціwwwroot. Однак великі програми можуть зіткнутися з організаційною проблемою такого типу, оскільки вам може знадобитися перемикатися між цими папками, щоб використовувати конкретні компоненти. Зі збільшенням кількості файлів і підпапок у кожній папці складність лише збільшуватиметься, тому вам доведеться витратити більше часу на перегляд у браузері рішення. Одним із способів вирішити цю проблему може бути сортування коду програми за компонентами, а не за типом файлу. Ця організація часто називається папками компонентів або фрагментами.

					ДП.ІПЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64



З цієї причини модель MVC ASP.NET Core підтримує області. Ви можете використовувати домени для створення окремих наборів контролера та перегляду папок (та пов'язаних з ними моделей) у кожній папці домену. На малюнку. На рисунку 3.28 показаний приклад структури папок області використання.

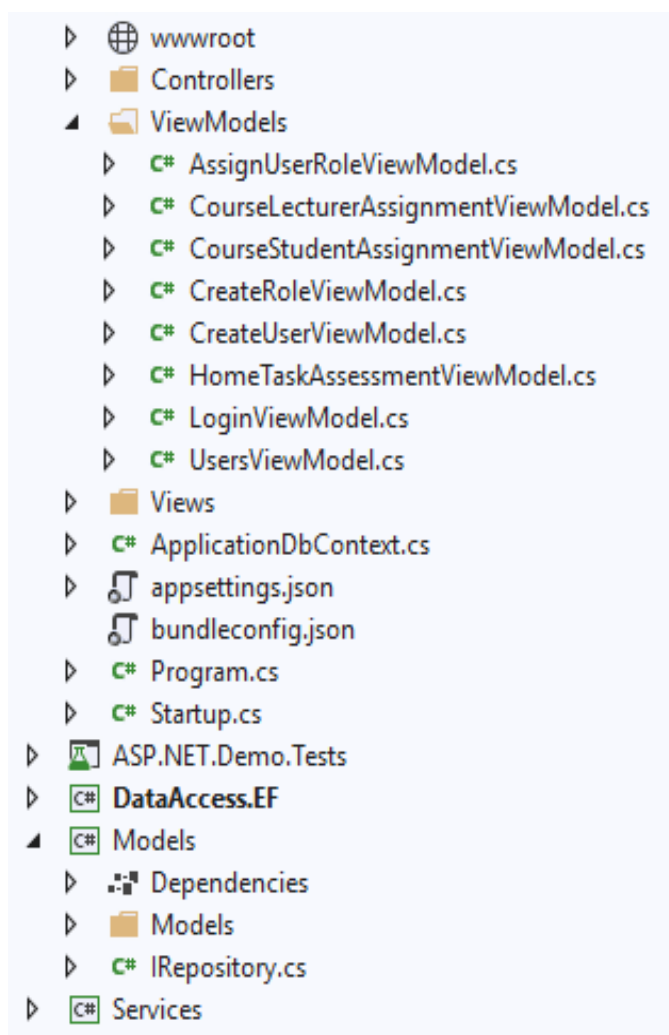


Рисунок 3.28 - Приклад регіональної організації

## 4 БІЗНЕС ПЛАН ДО НАВЧАЛЬНОЇ ПЛАТФОРМИ

### 4.1 Резюме

Основні тези: реалізація дистанційного навчання; цільова аудиторія проекту студенти, викладачі, учні, вчителі, корпорації, люди які здобувають знання в інтернеті, об'єм фінансування від 50 тисяч до 100 тисяч; планова виручка за 1 рік приблизно становить 25 тисяч; до проекту буде залучено 5-6 людей;

### 4.2 Маркетинг Навчальної платформи на .Net

Вид послуг: систему дистанційного навчання; онлайн навчання, онлайн оцінювання ; рентабельність послуги 66%, патент відсутній, досвід реалізації послуги присутній; сфери використання послуги Інтернет; унікальна торгова позиція полягав в забезпеченні якісного та простого у використанні продукту; характеристики(простий у використанні, українізований, загальнодоступний). Ринок збуту: географічне положення Україна; тип покупців всі хто цікавиться дистанційним викладанням матеріалу або навчання; тип послуги забезпечення онлайн навчанням

Джерела, які використовуються для вивчення ринку: Положення про дистанційне навчання [13]; звіти у періодичній пресі [14];

Оцінка кількості клієнтів на послугу: попит може наростати як у ВНЗ так і школах України; потенційні клієнти учні віком від 9 до 17 років , студенти віком від 17 років, викладачі, лектори, вчителі віком від 23 років.

Маркетингова підтримка: клієнти зможуть дізнатися про послугу в рекламі яку буде знаходитися в соціальних мережах; на рекламу заплановані затрати від 2-х тисяч за місяць ; подання даної послуги буде відбуватися через ілюстраційно-описові зображення.

					ДП.ІПЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

План збуту: власними силами а також через посередників; послуга подається в роздріб; попереднє замовлення не передбачається.

Підбір персоналу та оплата праці: розробник бек-енд , фронт-енд , адміністратор пз, піар менеджер , менеджер проекту, директор; можливо найти усіх співробітників в ВНЗ студентів 4-5 курсу, а також в Інтернет пошуку вакансій; необхідний режим роботи працівників з 9.00 до 17.00; оплата працівникам відбувається погодинно.

#### **4.3 Тип розроблюваних Навчальної платформи організаційний план .**

Розробка Навчальної платформи:

- привітальна сторінка або «сторінка-заглушка» потрібна перед відкриттям основного сайту проекту або в перші місяці роботи компанії. Сторінка розповість відвідувачам про проект і датою повноцінного запуску. Форма підписки допоможе зібрати адреси або телефони зацікавлених відвідувачів, щоб безпосередньо розповісти їм про відкриття основного сайту, вчасно повідомити про акції і знижки. Привітальна сторінка - хороший спосіб зібрати базу потенційних клієнтів ще до запуску основного проекту. термін 2-4 тижні;

- якщо потрібно перевірити бізнес-гіпотезу, обмежений бюджет проекту або потрібен сайт з базовими функціями, наприклад, Лендінгем. Конструктор сайтів дозволяє зробити багатосторінковий сайт, витративши мінімум часу і сил на верстку і інтеграцію з системою управління. Ціна – 45 000 гривень, термін 4-8 тижнів;

- якщо для проекту треба структурувати велику базу даних і впровадити складні функції, то необхідно розробити повноцінний сайт на власній системі управління з будь-якими заданими функціями і дизайном. Система управління підбирається під конкретний проект. Ціна – 75 000 гривень, термін 8-12 тижнів;

					ДП.ІПЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

#### 4.4 Зведений календарний план реалізації Навчальної платформи

Календарний план складається з етапів та їх назви, термінів виконання, вартість у таблиці 4.1. Доцільно складати календарний план на перший рік роботи.

Крім календарного наводиться кошторис витрат: закупка обладнання в офіс в таблиці 4.3; ОФП наведений в таблиці 4.2; послуги зв'язку мобільні оператори; оплата послуг сторонніх організацій відсутня; оренда офісу; витрати на організацію; інші витрати.

Таблиця 4.1 – Календарний план

Назва	Терміни виконання	Вартість
Розробка плану Навчальної системи	1-2 місяці	500-1000 грн.
Аналіз літератури та дослідження особливостей аналогічних проектів.	1-2 місяці	2000-3000 грн.
Аналіз характеристик та розробка моделей	2-3 місяці	8000-12000 грн.
Розробка на основі технологій .Net	4-5 місяці	15000-25000 грн
Запуск на хостинг	2-3 тижні	1000-3000 грн

Таблиця 4.2 - Фінансова модель. Фонд оплати праці, грн.

Посада	Кількість співробітників	Середня з/п в місяць на співробітника
Директор	1	6600
Менеджер проектів	1	5200
Розробник фронт-енд	1	8000
Піар менеджер	1	3500
Розробник бек-енд	1	8800
Страхові внески		5000
Разом ФОП	5	37 100

Повний розрахунок ФОП на перший місяці фінансовій моделі.

Таблиця 4.3 - Витрати на купівлю обладнання та додаткових приладів

Інвестиції на відкриття	Сума в грн
Рєєстрація, виготовлення печатки	1250
Розробка логотипу та фірмового стилю	2000
Вивіска	5000
Рекламна кампанія	8000
Закупівля обладнання	12125
Інше	5 000
Разом	33 375

Таблиця 4.4 – Витрати на організацію Навчальної платформи

Щомісячні витрати	Сума в грн.
ФОП ( без відрахування)	37 100
Оренда	10 000
Амортизація	2 850
Комунальні послуги	1 750
Реклама	2 000
Бухгалтерія (віддалена)	2 500
Інші витрати	5 000
Разом	61 200

#### 4.5 Склад фінансового бюджету

Визначення джерел фінансування: інвестиції від навчальних закладів а також від установ які бажають проводити навчальні курси; позикові кошти у сторонніх організацій; кредитні кошти у банків.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

Зведений кошторис витрат до отримання перших прибутків . Залучення додаткових коштів може відбуватися як позикові кошти або кредитні. Очікування в витрат на оренду та закупівлю устаткування приблизно 40 000 грн.

План очікування прибутків та збитків проекту; прибуток від реалізації послуги приблизно становить 10 % від платних курсів а також 250 грн за послугу безплатних курсів або від розробки власних платних курсів ціна становить від 800 до 3000 грн. Загальна сума в місяць приблизно 78 000 грн. Чистий прибуток становить 22 000 грн.

План руху грошових коштів: джерела з яких будуть отримані кошти (власний заробіток 22 000 при повному запуску проекту, кошти від позик на перших кроках становить 80 000 грн., кредитні кошти 30 000 грн., інвестиції 15 000 грн.

Розрахунок показників проекту: загальна сума чистого прибутку за перший рік приблизно становить 90 000 грн., рентабельність яка визначається за наступною формулою: сума чистого прибутку за 1 рік, поділена на об'єм вкладень власних коштів, та помножити на 100% [17]. 72% рентабельності.

Термін окупності, який визначається як період часу у місяцях, протягом якого накопичена сума чистого прибутку повністю покриває загальні суму початкових особистих фінансових вкладень орієнтовно 14-18 місяць після запуску проекту.

#### **4. 6 Оцінка можливих ризиків підприємства**

Низький рівень технічних фахівців. Необхідно ретельно відбирати членів команди, заохочувати постійний розвиток, прагнення до нових викликів.

Недостатній попит. Зміцнення репутації, високий рівень сервісу, рішення задач точно в строк, налагодження довгострокових відносин із замовниками.

Велика кількість правок від замовника невдоволення результатом. Потрібно проводити роз'яснювальну роботу, тому що клієнти не є

					ДП.ІПЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		70

фахівцями і їм важко сформулювати свої вимоги. Після складається детальне технічне завдання, яке закріплює домовленості і підтверджує, що сторони порозумілися.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71

## ВИСНОВКИ

Ціль даної роботи заключалася в аналізі можливості використання сучасних технологій в освітньому процесі, розробка навчальної платформи на основі технології .NET та організація роботи з навчальними курсами.

Реалізовано наступний функціонал:

1. Лектору та адміністратору надана можливість додавання, редагування і видалення навчального курсу.
2. Права доступу розмежовані: користувач-адміністратор здійснює додавання, редагування і видалення всіх навчальних курсів, а користувач-лектор – тільки свого.
3. Користувач-студент вивчає матеріали і виконує домашню роботу свого навчального курсу.
4. Інтерфейс системи доступний будь-якому користувачеві.
5. Система при роботі – стійка та безпечна.
6. Робота в системі є зручною для використання.
7. Об'єктно-орієнтований підхід, використаний в процесі розробки, дозволяє здійснювати подальшу модернізацію системи.

Навчальна платформа реалізує основні задачі для дистанційного навчання.

Проведено також обґрунтування економічної доцільності розробки.

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72



## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

### REFERENCES

1. Learning Management System. URL: <https://atutor.github.io/> (дата звернення: 11.03.2020).
2. Open Source Learning Management System. URL: <https://sourceforge.net/projects/claroline/> (дата звернення: 15.03.2020).
3. Get started with Office 365 for free. URL: <https://www.microsoft.com/en-us/education/products/office?tab=schools&CorrelationId=01069668-7ebd-4604-9446-8fc6ef6ce60c> (дата звернення: 01.04.2020).
4. Efront. URL: <https://www.efront.com/> (дата звернення: 13.04.2020).
5. Moodle. URL: <https://moodle.org/> (дата звернення: 15.04.2020).
6. SHAREPOINT LMS is now LMS365. URL: <https://www.sharepointlms.com/> (дата звернення: 17.04.2020).
7. Learningapps. URL: <https://learningapps.org/> (дата звернення: 15.04.2020).
8. Навчально-науковий центр якості надання освітніх послуг і дистанційного навчання. URL: <https://d-learn.pnu.edu.ua/> (дата звернення: 15.04.2020).
9. Концепция разработки MVC — Model, View, Controller. URL: <https://rtfm.co.ua/django-book-model-razrabotki-mtc-model-view-controller/> (дата звернення: 21.04.2020).
10. Microsoft представила обновления и новые инструменты для разработчиков на Azure, Visual Studio. URL: <https://itc.ua/news/microsoft-predstavila-obnovleniya-novyie-instrumentyi-dlya-razrabotchikov-na-azure-visual-studio-ne-tolko/> (дата звернення: 01.05.2020).
11. Прайс Марк ДЖ. С # 7 і .NET Core. Крос-платформна розробка для професіоналів: навч. посіб 3-тє вид. Пітер: Пітер, 2018. 640 с.

					ДП.ІІЗ-09.ІІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

- 12.Суб'єкти. URL: <https://osvita.ua/vnz/reports/law/9516/> (дата звернення: 15.04.2020).
- 13.Положення про дистанційне навчання. URL: <http://vnz.org.ua/dystantsijna-osvita/polozhennja-kmu?start=1> (дата звернення: 15.05.2020).
- 14.Час дистанційного навчання. URL: <https://www.unn.com.ua/uk/news/1868007-u-mon-rozpovili-yak-otsinyuyut-uchniv-pid-chas-distantiynogo-navchannya> (дата звернення: 18.05.2020).
- 15.Романов А.Н., Торопцова В.С. Технология дистанционного обучения - М: навч. посіб. Юнити-Дана, 2000, 304 с.
- 16.Харрингтон Д. Проектирование объектно-ориентированных баз данных: навч. посіб. Москва: ДМК Пресс, 2012, 272 с.
- 17.Економіка підприємств. URL: [https://osvita.ua/vnz/reports/econom\\_pidpr/19538/](https://osvita.ua/vnz/reports/econom_pidpr/19538/) (дата звернення: 18.05.2020).
- 18.Моніторинг сайту. URL: <https://code.tutsplus.com/articles/50-ways-to-get-your-site-noticed--net-15620> (дата звернення: 18.05.2020).
- 19.Дистанційне навчання: цілі, принципи, особливості, вимоги. URL: <http://albeskaa.blogspot.com/2012/12/blog-post.html> (дата звернення: 15.05.2020).
- 20.Дидактичні принципи дистанційного навчання. URL: <http://elar.khnu.km.ua/jspui/bitstream/123456789/3525/1/%D0%B4%D0%B8%D0%B0%D0%BA%D1%82%D0%B8%D1%87%D0%BD%D1%96%20%D0%BF%D1%80%D0%B8%D0%BD%D1%86%D0%B8%D0%BF%D0%B8%20%D0%B4%D0%B8%D1%81%D1%82%D0%B0%D0%BD%D1%86%D1%96%D0%B9%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F.pdf> (дата звернення: 13.05.2020).
- 21.П. Федорук і М. Дутчак, “Побудова бази знань адаптивних систем дистанційного навчання на основі фреймової та продукційної моделей представлення знань,” Управляючі системи і машини (УСiМ), №5, с.35-42, 2012.

					ДП.ІПЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

- 22.М. Дутчак, "Моделювання процесу автоматизованої побудови індивідуалізованого навчання в інтелектуальних освітніх онлайн системах," Прикладні науково-технічні дослідження: матеріали ІV міжнар. наук.-практ. конф., Івано-Франківськ, 2020, т.1., с. 47-51.
- 23.Джеймс Ч., Дэвид П., Смот Т. ASP.NET Core. Разработка приложений: навч. посіб. Питер: Серия «Для профессионалов»). 2018. 464с.
- 24.Троелсен Эндрю, Джепикс Филипп. Язык программирования С# 7 и платформы .NET и .NET Core: навч. посіб. Парал: Диалектика, 2018. 1328 с.

					ДП.ІІЗ-09.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

## Додаток А

### Сирцевий код Навчальної платформи

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
  <title>@ViewData["Title"]НП</title>

  <environment include="Development">
    <link rel="stylesheet"
href="~/lib/bootstrap/dist/css/bootstrap.css" />
    <link rel="stylesheet" href="~/css/site.css" />
  </environment>
  <environment exclude="Development">
    <link rel="stylesheet"
href="https://ajax.aspnetcdn.com/ajax/bootstrap/3.3.7/css/bootstra
p.min.css"
      asp-fallback-
href="~/lib/bootstrap/dist/css/bootstrap.min.css"
      asp-fallback-test-class="sr-only" asp-fallback-test-
property="position" asp-fallback-test-value="absolute" />
    <link rel="stylesheet" href="~/css/site.min.css" asp-
append-version="true" />
  </environment>
</head>
<body>
  <nav class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
          <span class="sr-only">Навігація</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a asp-area="" asp-controller="Home" asp-
action="Index" class="navbar-brand">НП</a>
      </div>
      <div class="navbar-collapse collapse">
        <ul class="nav navbar-nav">

          <li><a asp-controller="Course" asp-
action="Courses">Курс</a></li>
          <li><a asp-controller="Student" asp-
action="Students">Студент</a></li>

```

## Продовження Додатку А

```

        <li><a asp-controller="Lecturer" asp-
action="Lecturers">Лектор</a></li>
        @if (!User.Identity.IsAuthenticated)
        {
            <li><a asp-controller="Security" asp-
action="Login">Вхід</a></li>
        }
        else
        {
            if (User.IsInRole("Admin"))
            {
                <li><a asp-controller="Security" asp-
action="CreateUser">Створення Користувача</a></li>
                <li><a asp-controller="Security" asp-
action="Users">Користувачі</a></li>
                <li><a asp-controller="Security" asp-
action="CreateRole">Створення Ролі</a></li>
            }
            <li><p>Hello, @User.Identity.Name</p></li>
            <li><a asp-controller="Security" asp-
action="Logout">Вихід</a></li>
        }
    </ul>
</div>
</div>
</nav>
<div class="container body-content">
    @RenderBody()
    <hr />
    <footer>
        <p>&copy; 2020</p>
    </footer>
</div>

<environment include="Development">
    <script src="~/lib/jquery/dist/jquery.js"></script>
    <script
src="~/lib/bootstrap/dist/js/bootstrap.js"></script>
    <script src="~/js/site.js" asp-append-
version="true"></script>
</environment>
<environment exclude="Development">
    <script
src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-2.2.0.min.js"
asp-fallback-src="~/lib/jquery/dist/jquery.min.js"
asp-fallback-test="window.jQuery"
crossorigin="anonymous"
integrity="sha384-
K+ctZQ+LL8q6tP7I94W+qzQsfRV2a+AfHII9k8z8l9ggpc8X+Ytst4yBo/hH+8Fk">
</script>

```

## Продовження Додатку А

```
<script
src="https://ajax.aspnetcdn.com/ajax/bootstrap/3.3.7/bootstrap.min
.js"
    asp-fallback-
src="~/lib/bootstrap/dist/js/bootstrap.min.js"
    asp-fallback-test="window.jQuery &&
window.jQuery.fn && window.jQuery.fn.modal"
    crossorigin="anonymous"
    integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNIpG9mGCD8wGNicPD7Txa">
</script>
<script src="~/js/site.min.js" asp-append-
version="true"></script>
</environment>

@RenderSection("Scripts", required: false)
</body>
</html>
```

## Додаток Б

### Сирцевий код Навчальної платформи

```

namespace Models.Models
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel.DataAnnotations;

    public class Student
    {
        [Required(AllowEmptyStrings = false)]

        public string Name { get; set; }

        public int Id { get; set; }

        [Display(Name = "Birth date")]
        [DataType(DataType.Date)]
        [Range(typeof(DateTime), "1/2/1004", "3/4/3004",
            ErrorMessage = "Value for {0} must be between {1} and
{2}")]
        public DateTime BirthDate { get; set; }

        [Display(Name = "Phone number")]
        [DataType(DataType.PhoneNumber)]
        [Required(AllowEmptyStrings = false)]
        public string PhoneNumber { get; set; }

        [DataType(DataType.EmailAddress)]
        [Required(AllowEmptyStrings = false)]
        public string Email { get; set; }

        public string GitHubLink { get; set; }

        public string Notes { get; set; }

        public virtual List<HomeTaskAssessment>
HomeTaskAssessments { get; set; }

        public virtual List<StudentCourse> Courses { get; set; }
    }
}

```

## Додаток В

### Сирцевий код Навчальної платформи

```
using Microsoft.AspNetCore.Mvc;

namespace ASP.NET.Demo.Controllers
{
    using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Net;

    using ASP.NET.Demo.ViewModels;

    using Microsoft.AspNetCore.Authorization;

    using Models.Models;
    using Services;

    public class CourseController : Controller
    {
        private readonly CourseService courseService;
        private readonly StudentService studentService;
        private readonly LecturerService lecturerService;

        public CourseController(CourseService courseService,
StudentService studentService, LecturerService lecturerService)
        {
            this.courseService = courseService;
            this.studentService = studentService;
            this.lecturerService = lecturerService;
        }

        // GET
        public IActionResult Courses()
        {
            return View(this.courseService.GetAllCourses());
        }

        [Authorize(Roles = "Admin")]
        public IActionResult Delete(int id)
        {
            this.courseService.DeleteCourse(id);

            return RedirectToAction("Courses");
        }

        [Authorize(Roles = "Admin")]
        public IActionResult Create()
        {

```



**Продовження Додатку В**

```
        ViewData["action"] = nameof(this.Create);
        return this.View("Edit", new Course());
    }

    [HttpGet]
    [Authorize(Roles = "Admin")]
    public IActionResult Edit(int id)
    {
        Course course = this.courseService.GetCourse(id);
        if (course == null)
        {
            return this.NotFound();
        }

        ViewData["action"] = nameof(this.Edit);

        return this.View(course);
    }

    [HttpPost]
    [Authorize(Roles = "Admin")]
    public IActionResult Edit(Course courseParameter)
    {
        if (courseParameter == null)
        {
            return this.BadRequest();
        }
        courseService.UpdateCourse(courseParameter);
        return this.RedirectToAction(nameof(Courses));
    }

    [Authorize(Roles = "Admin")]
    [HttpPost]
    public IActionResult Create(Course courseParameter)
    {
        if (courseParameter == null)
        {
            return this.BadRequest();
        }

        if (!ModelState.IsValid)
        {
            ViewData["action"] = nameof(this.Create);

            return this.View("Edit", courseParameter);
        }
        this.courseService.CreateCourse(courseParameter);
        return this.RedirectToAction(nameof(Courses));
    }
}
```

## Продовження Додатку В

```

[Authorize(Roles = "Admin,Lecturer")]
[HttpGet]
public IActionResult AssignStudents(int id)
{
    var allStudents =
this.studentService.GetAllStudents();
    var course = this.courseService.GetCourse(id);
    CourseStudentAssignmentViewModel model = new
CourseStudentAssignmentViewModel();

    model.Id = id;
    model.EndDate = course.EndDate;
    model.Name = course.Name;
    model.StartDate = course.StartDate;
    // model.PassCredits = course.PassCredits;
    model.Students = new List<StudentViewModel>();

    foreach (var student in allStudents)
    {
        bool isAssigned = course.Students.Any(p =>
p.StudentId == student.Id);
        model.Students.Add(new StudentViewModel() {
StudentId = student.Id, StudentFullName = student.Name, IsAssigned
= isAssigned });
    }

    return this.View(model);
}

[Authorize(Roles = "Admin,Lecturer")]
[HttpPost]
public IActionResult
AssignStudents(CourseStudentAssignmentViewModel
assignmentViewModel)
{

this.courseService.SetStudentsToCourse(assignmentViewModel.Id,
assignmentViewModel.Students.Where(p =>
p.IsAssigned).Select(student => student.StudentId));

    return RedirectToAction("Courses");
}

[HttpGet]
[Authorize(Roles = "Admin,Lecturer")]
public IActionResult AssignLecturers(int id)
{
    var allLecturers =
this.lecturerService.GetAllLecturers();
    var course = this.courseService.GetCourse(id);

```

**Продовження Додатку В**

```

        CourseLecturerAssignmentViewModel model = new
CourseLecturerAssignmentViewModel();

        model.Id = id;
        model.EndDate = course.EndDate;
        model.Name = course.Name;
        model.StartDate = course.StartDate;
        model.PassCredits = course.PassCredits;
        model.Lecturers = new List<LecturersViewModel>();

        foreach (var lecturer in allLecturers)
        {
            bool isAssigned = course.Lecturers.Any(p =>
p.LecturerId == lecturer.Id);
            model.Lecturers.Add(new LecturersViewModel() {
LecturerId = lecturer.Id, Name = lecturer.Name, IsAssigned =
isAssigned });
        }

        return this.View(model);
    }

    [HttpPost]
    [Authorize(Roles = "Admin,Lecturer")]
    public IActionResult
AssignLecturers(CourseLecturerAssignmentViewModel model)
    {
        this.courseService.SetLecturersToCourse(model.Id,
model.Lecturers.Where(p => p.IsAssigned).Select(lecturer =>
lecturer.LecturerId));

        return RedirectToAction("Courses");
    }
}
}

```

## Додаток Г

### Сирцевий код Навчальної платформи

```

namespace DataAccess.EF
{
    using Microsoft.EntityFrameworkCore;
    using Microsoft.Extensions.Options;
    using Models.Models;

    public class UniversityContext : DbContext
    {
        private readonly IOptions<RepositoryOptions> options;

        public UniversityContext(IOptions<RepositoryOptions>
options)
        {
            this.options = options;
        }

        protected override void
OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            base.OnConfiguring(optionsBuilder);

optionsBuilder.UseSqlServer(options.Value.DefaultConnectionString)
;
            optionsBuilder.UseLazyLoadingProxies();
        }

        protected override void OnModelCreating(ModelBuilder
modelBuilder)
        {
            base.OnModelCreating(modelBuilder);
            modelBuilder.Entity<LecturerCourse>().HasKey(k => new
{ k.LecturerId, k.CourseId });
            modelBuilder.Entity<StudentCourse>().HasKey(k => new {
k.CourseId, k.StudentId });
        }

        public DbSet<Student> Students { get; set; }

        public DbSet<Course> Courses { get; set; }

        public DbSet<HomeTask> HomeTasks { get; set; }

        public DbSet<Lecturer> Lecturers { get; set; }
    }
}

```