

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

**Кваліфікаційна робота
на здобуття ступеня магістра**

за спеціальністю 122 Комп'ютерні науки
на тему:

**АВТОМАТИЗАЦІЯ ПРОЦЕСУ РОЗРОБКИ РОБОЧИХ ПРОГРАМ
НАВЧАЛЬНИХ ДИСЦИПЛІН ДЛЯ ЗАКЛАДІВ ВИЩОЇ ОСВІТИ**

Виконав студент 2-го курсу магістратури
Мітічкін Дмитро Сергійович



(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Омельчук Людмила Леонідівна



(підпис)

Засвідчую, що в цій роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент



(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри теорії та технології
програмування

«27» квітня 2021 р., протокол № 10

Завідувач кафедри

М. С. Нікітченко

(підпис)

РЕФЕРАТ

Обсяг роботи 83 сторінки, 32 ілюстрації, 21 таблиця, 17 джерел посилань, 5 додатків.

АВТОМАТИЗАЦІЯ НАВЧАЛЬНОГО ПРОЦЕСУ, РОЗРОБКА РОБОЧИХ ПРОГРАМ, ВИМОГИ ДО СИСТЕМИ, РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ, БАЗИ ДАНИХ, РОЗРОБКА ВЕБЗАСТОСУНКУ.

Об'єктом даної роботи є процес розробки робочих програм навчальних дисциплін для закладів вищої освіти. Предметом роботи є програмний засіб для автоматизації процесу розробки робочих програм навчальних дисциплін для закладів вищої освіти.

Метою роботи є проектування та розробка програмного засобу для автоматизації процесу розробки робочих програм навчальних дисциплін.

Методи розробки: проектування бази даних, розробка програмного продукту на основі еволюційної методології. Інструменти розробки: СКБД Microsoft SQL Server Management Studio 18, середовище розробки Microsoft Visual Studio Community 2019 16.9.3, технології Entity Framework Core (EF Core) Database First — ORM Framework, ASP.NET MVC, Bootstrap. Мови: SQL, C#, HTML, CSS. Бібліотеки: Xceed DocX.

Результати роботи: розглянуто процес розробки робочих програм навчальних дисциплін, сформована інструкція до заповнення форми робочої програми, визначені способи автоматизувати даний процес, досліджено питання наявності програмних засобів для вирішення аналогічних задач, висунуто вимоги до програмного засобу автоматизації розробки робочої програми, досліджено технології для його розробки, спроектовано та реалізовано базу даних з інформацією про освітні програми на факультетах вищих навчальних

закладів, дисципліни, що до них входять, та відповідності їм програмних результатів та компетентностей з освітніх програм, розроблено вебзастосунок, який дозволить працювати з базою даних та сформувавши для кожної дисципліни шаблон робочої програми, заповнений інформацією з бази.

Розроблений програмний засіб може застосовуватись викладачами при розробці робочих програм своїх дисциплін. До бази адміністратором централізовано вносяться зміни освітніх програм та навчальних планів, а викладачі отримують шаблони робочих програм, які містять інформацію з освітньої програми з усіма внесеними змінами, без потреби звіряти ці дані вручну.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 НАЯВНІСТЬ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ВИРІШЕННЯ АНАЛОГІЧНИХ ЗАДАЧ.....	12
РОЗДІЛ 2 ПІДГОТОВКА ДО РОЗРОБКИ	17
2.1 Процес розробки робочої програми навчальної дисципліни	17
2.2 Задача автоматизації заповнення форми робочої програми.....	26
2.3 Огляд та вибір технологій	28
РОЗДІЛ 3 ПРИЗНАЧЕННЯ І ВИМОГИ ДО ЗАСТОСУНКУ	31
3.1 Цілі створення застосунку.....	31
3.2 Вимоги до застосунку	32
РОЗДІЛ 4 РЕАЛІЗАЦІЯ ЗАСТОСУНКУ	36
4.1 Структура бази даних	36
4.2 Структура системи вебзастосунку	44
РОЗДІЛ 5 ІНСТРУКЦІЯ З ВИКОРИСТАННЯ ЗАСТОСУНКУ	53
5.1 Інструкція для користувача.....	53
5.2 Інструкція для адміністратора	61
ВИСНОВКИ.....	65
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	67
ДОДАТОК А.....	70
ДОДАТОК Б	71

ДОДАТОК В	72
ДОДАТОК Г	74
ДОДАТОК Д.....	77

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

БД – база даних;

ВСК – визначена спеціальністю компетентність;

ЗВО – заклад вищої освіти;

ЗК – загальна компетентність;

ІС – інформаційна система;

ІКТ – інформаційно-комунікаційні технології;

МОН – Міністерство освіти і науки;

НП – навчальний план;

ОС – операційна система;

ООП – об'єктно-орієнтоване програмування;

ОП – освітня програма;

ПЗ – програмне забезпечення;

ПІБ – прізвище, ім'я, по-батькові;

ПРН – програмні результати навчання;

РПНД – робоча програма навчальної дисципліни;

СК – спеціальна компетентність;

СКБД – система керування базами даних;

ФК – фахова компетентність;

ASP.NET – Active Server Pages (активні серверні сторінки) .NET;

ECTS – European Credit Transfer and Accumulation System (Європейська кредитна трансферно-накопичувальна система);

EF – Entity Framework (фреймворк сутностей);

MVC – Model View Controller (модель–представлення–контролер);

ORM – Object-relational mapping (об’єктно-реляційна проекція);

SQL – Structured Query Language (структурна мова запитів).

ВСТУП

Однією з основних задач закладів вищої освіти (ЗВО) є ефективна організація навчально-пізнавальної та наукової діяльності. З розвитком інформаційних технологій, зростанням їх загальної доступності, та зростанням за рахунок цього темпу розвитку суспільства, постає питання про застосування інновацій в сфері інформатики для підвищення якості організації навчального процесу, а саме: зменшення кількості помилок, більш ефективного використання часу викладачами, методистами та студентами, підвищення зручності доступу до інформації.

Підвищення ефективності вирішення організаційних питань сприяє підвищенню рівня освіти в цілому: за рахунок більш ефективного розподілу часу та сил виникає можливість змістити фокус ближче до самого навчання та наукової діяльності, та робити кроки щодо покращення процесу навчання замість використання тих самих сил виключно для підтримання існуючого рівня.

Актуальність роботи та підстави для її виконання. Однією із задач, яка виникає в рамках глобальної задачі організації навчального процесу, є задача складання робочих програм навчальних дисциплін. Ця задача включає в себе заповнення низки позицій з інформацією про дисципліну, для якої складається програма, таких як: інформація про освітню програму (ОП), навчальні цілі, схема формування оцінки, структура дисципліни тощо. При заповненні цих позицій виникає потреба звертатись до двох інших нормативних документів: до опису ОП, до якої входить дисципліна, та до навчального плану (НП), який містить інформацію про кількість кредитів ECTS та кількості годин, які виділяються на складові даної дисципліни.

Через це під час заповнення викладачем форми робочої програми навчальної дисципліни (РПНД) він має постійно звертатись до документів, шукати дані в матрицях та таблицях, та кожного разу перевіряти переписані з документів дані. Це підвищує ймовірність допущення помилок при розробці робочої програми, збільшує час, витрачений на її розробку, і зміщує фокус викладача з планування освітньої діяльності на коректне заповнення документів.

Також зі стрімким розвитком галузей знань, особливо галузей, пов'язаних з комп'ютерними науками, гостро постає питання регулярної реорганізації навчального процесу для відповідності сучасним технологіям та тенденціям. Університет, який бажає не відставати від сучасної ситуації в сферах, за якими здійснюється підготовка фахівців, має все частіше вносити зміни до ОП та НП, а з ними — і до РПНД. А це означає, що ручне заповнення документів в майбутньому вимагатиме все більше і більше сил та часу. Таким чином, застосування програмних засобів для автоматизації різних аспектів даного процесу є вкрай важливим і актуальним питанням.

Мета і завдання роботи. Метою даної кваліфікаційної роботи є проектування та розробка програмного засобу для автоматизації процесу розробки РПНД. Ця мета передбачає виконання наступних завдань:

- Розглянути процес розробки РПНД, сформулювати інструкцію до заповнення форми робочої програми, визначити способи автоматизувати даний процес.
- Дослідити питання наявності програмних засобів для вирішення аналогічних задач, висунути вимоги до програмного засобу автоматизації розробки робочої програми, дослідити технології для його розробки.

- Спланувати та реалізувати базу даних (БД) з інформацією про ОП на факультетах ЗВО, дисципліни, що до них входять, та відповідності їм програмних результатів та компетентностей з ОП.
- Розробити вебзастосунок, який дозволить працювати з БД та сформувати для кожної дисципліни шаблон робочої програми, заповнений інформацією з бази.

Об’єкт та предмет роботи, засоби і методи розробки. Об’єктом даної роботи є процес розробки РПНД для освітніх програм в ЗВО. Предметом роботи є програмний засіб для автоматизації процесу розробки РПНД для освітніх програм в ЗВО.

Розробка програмного засобу складалася з двох основних етапів: реалізація БД освітніх програм та розробка вебзастосунку, який відображає дані з бази і дозволяє знайти дисципліну та сформувати для неї шаблон робочої програми.

Методологія розробки — еволюційна. Вона полягає в тому, що після розробки версії продукту вона демонструється замовнику (науковому керівнику), після чого розробляється наступна версія з урахуванням побажань замовника. Даний процес повторюється, поки не буде створено продукт, який задовольняє всі потреби замовника.

Для бази даних було розгорнуто SQL-сервер, її розробка велась за допомогою системи керування базами даних Microsoft SQL Server Management Studio 18. Для розробки вебзастосунку було обрано технології: Entity Framework Core (EF Core) Database First — ORM Framework, ASP.NET MVC, Bootstrap. Розробка велась у середовищі розробки Microsoft Visual Studio 2019. Версія — безкоштовна Visual Studio Community 2019 16.9.3. Для роботи з документами у форматі .docx було використано бібліотеку DocX від Xceed Software.

Даний набір технологій дозволяє ефективно працювати з базами даних, передавати дані з них до представлень, та обробляти запити користувачів, які працюють в своєму браузері з представленнями вебзастосунку.

Розроблений програмний засіб може застосовуватись викладачами при розробці РПНД. До бази централізовано вносяться зміни ОП та НП, а викладачі отримують шаблони робочих програм, які містять інформацію з освітньої програми з усіма внесеними змінами, без потреби звіряти ці дані вручну.

РОЗДІЛ 1

НАЯВНІСТЬ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ВИРІШЕННЯ АНАЛОГІЧНИХ ЗАДАЧ

Ідея застосування програмного забезпечення (ПЗ) для підвищення ефективності процесу організації навчальної та наукової діяльності не є новою. Системи автоматизації застосовуються повсюдно в закладах вищої освіти. Якщо система автоматизації не застосовується, опис ОП та НП, як правило, можна знайти на вебсайті факультету. Але вони не завжди оновлюються і можуть бути застарілими, і можлива ситуація, коли актуальні матеріали просто передаються викладачами один одному через електронну пошту, без використання будь-яких засобів автоматизації. Системи автоматизації, впроваджені в різних університетах, мають різний функціонал. Розглянемо деякі з них.

Одна з таких систем — система автоматизації Київського національного університету імені Тараса Шевченка, або Triton [1]. Для студентів за умови їх реєстрації сервіс Triton Student надає можливість здійснювати вибір дисциплін, вибір спеціалізації, переглядати свої оцінки, графік контролю з дисциплін, та переглядати план навчального процесу (див. рис.1.1).

Triton Student Новини Вибір Навчальний процес Пошта dmitchkin

Навчальний план: 2017, Інформатика

В - всього
ВН - всього навчальних
Л - лекції
К - консультації
С - семінари
ЛЗ - лабораторні заняття
ПЗ - практичні заняття
ІЗ - індивідуальні заняття
СР - самостійна робота
Н/ВП - навчальні та виробничі практики

Код	Назва	Курс	Семестр	Контроль	Кредити ЕСТS	Години										
						Навчальні заняття										
						з них:										
В	ВН	Л	К	С	ЛЗ	ПЗ	ІЗ	СР	Н/ВП							
1. Обов'язкові навчальні дисципліни																
ННД.09	Інформаційна безпека та криптографія	1	1	Екзамени	4	120	40	26	2	0	12	0	0	80	0	
ННД.08	Інформаційні мережі	1	1	Екзамени	4	120	40	26	2	0	12	0	0	80	0	
ННД.07	Сучасні операційні системи	1	1	Екзамени	4	120	40	26	2	0	12	0	0	80	0	
ННД.06	Теорія обчислень та комп'ютерні технології	1	1	Екзамени	5	150	50	26	2	0	22	0	0	100	0	
ННД.10	Штучний інтелект	1	1	Екзамени	4	120	40	26	2	0	12	0	0	80	0	
ННД.01	Методологія та організація наукових досліджень з основними інтелектуальною власністю	1	1	Заліки	3	90	30	14	2	14	0	0	0	60	0	
ННД.03	Психологія вищої школи	1	1	Заліки	3	90	28	26	2	0	0	0	0	62	0	
ННД.11	Актуальні проблеми обробки інформації в комп'ютерних системах	1	2	Екзамени	5	150	48	46	2	0	0	0	0	102	0	
ННД.12	Формальні методи розробки програмних систем	1	2	Екзамени	4	120	40	38	2	0	0	0	0	80	0	

Рисунок 1.1 — Інтерфейс системи Triton Student, перегляд навчального плану

Також в системі Triton наявний функціонал для створення та узгодження НП, генерації файлів з НП, формування оперативних планів, статистика про успішність студентів, інформація про навчальні аудиторії (рис. 1.2, 1.3).

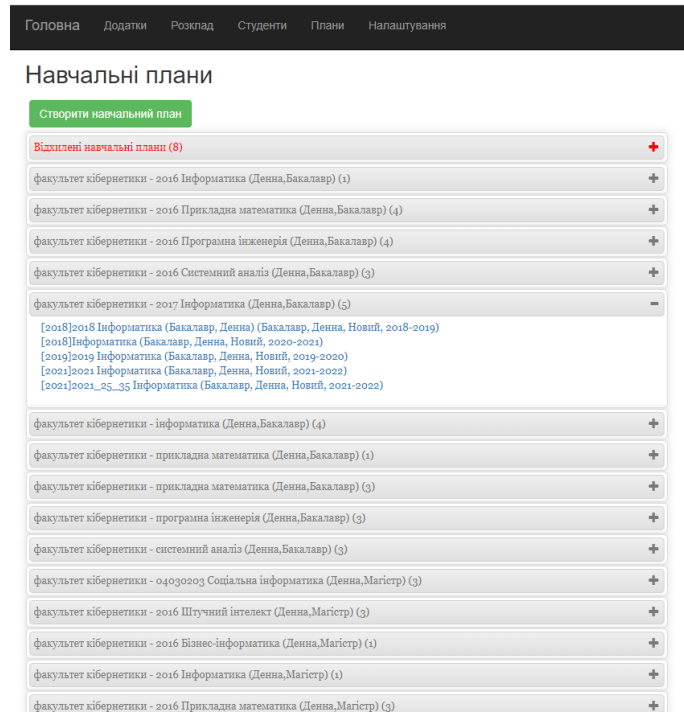


Рисунок 1.2 — Функціонал системи Triton для роботи з НП

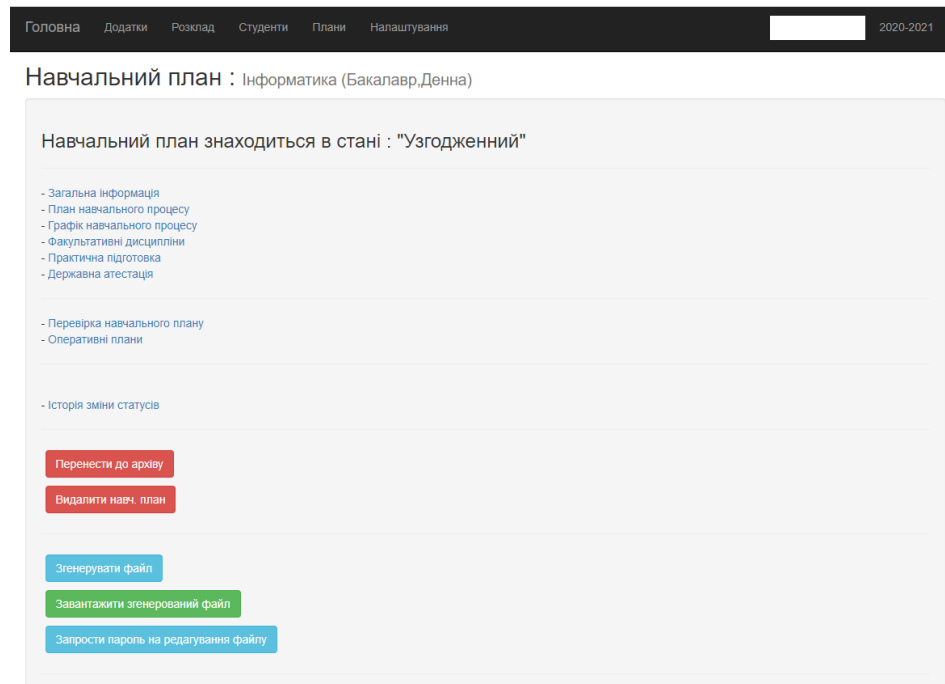


Рисунок 1.3 — Функціонал системи Triton для роботи з НП

Описи ОП та НП факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка можна знайти на сайті факультету [2], а РПНД — на сайтах кафедр [3, 4]. Triton дозволяє формувати та перевіряти НП, і дані документи далі завантажуються на сайти в ручному режимі, без використання систем автоматизації.

Даний засіб є зручним, дозволяє переглядати актуальну інформацію, і містить всі дані з навчального плану. Його задача полягає в полегшенні процесу розробки НП і донесення інформації до студентів, однак він не містить інформації з опису ОП, та не містить посилань на робочі програми навчальних дисциплін.

Розглянемо засоби автоматизації організації освітнього процесу в Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського». Конструкторським бюро інформаційних систем розробляється єдине інформаційне середовище університету, яке включає в себе ряд підсистем для автоматизованого ведення баз даних документів, пов'язаних з організацією освітнього процесу, наукової, адміністративної та іншими видами діяльності [5]. Для підтримки освітнього процесу було створено систему «Електронний кампус», яка почала функціонувати в 2012 році. Вона надає централізований доступ до інформаційних ресурсів, імпортованих з інших автоматизованих систем університету, і забезпечує комунікацію між студентами і викладачами, розповсюджує інформацію про майбутні події і заходи, і надає доступ до відомостей про діяльність ректорату і інших підрозділів університету [6].

Для підтримки освітнього процесу в університеті в майбутньому заплановано ввести підсистеми, які дозволяють:

- автоматизовано формувати навчальні плани за напрямом підготовки, спеціальністю, освітньо-кваліфікаційним рівнем тощо;
- автоматизовано формувати плани роботи кафедр та індивідуальних планів роботи викладачів;
- автоматизовано складати розклад занять та сесій;
- автоматизовано створювати та вести бази даних типових організаційно-методичних документів, методичних рекомендацій по проведенню занять, автоматизовано формувати та вести інформаційне та науково-методичне забезпечення дисциплін;
- автоматизовано формувати звіти про успішність студентів, додатки до дипломів, сертифікати та інші документи для студентів і випускників [5,7].

Розглянувши системи автоматизації різних ЗВО, можна зробити висновок, що хоч більшість з них надає доступ до описів ОП, розкладу та НП, а деякі надають можливість автоматизовано формувати НП, не було знайдено жодної системи, яка наразі має функціонал для автоматичного заповнення позицій в РПНД.

РОЗДІЛ 2

ПІДГОТОВКА ДО РОЗРОБКИ

В цьому розділі розглянемо існуючий, неавтоматизований, процес розробки РПНД. Для цього спиратимемося на форму РПНД, затверджену рішенням Науково-методичної ради Київського національного університету імені Тараса Шевченка від 30.06.2017 [8] та рекомендацій щодо її заповнення. Далі поставимо задачу розробки програмного засобу для автоматизації даного процесу.

2.1 Процес розробки робочої програми навчальної дисципліни

Процес складання РПНД починається з заповнення титульного аркушу. Спочатку заповнюються позиції **«Галузь знань»**, **«Спеціальність»**, **«Освітній рівень»** та **«Освітня програма»**. Якщо дисципліна входить до вибіркового блоку дисциплін, заповнюється позиція **«Спеціалізація»**. В залежності від того, коли запроваджена навчальна програма, вказуються різні шифри і назви галузі знань та спеціальності. Якщо програма запроваджена із 2016 року, шифри і назви галузі знань та спеціальності вказуються у відповідності до Постанови Кабінету Міністрів України від 29.04.15 р. за № 266 **«Про затвердження переліку галузей знань і спеціальностей, за якими здійснюється підготовка здобувачів вищої освіти»** (зі змінами) [9]. Наприклад, це буде актуальним в рамках програми **«Інформатика»** спеціальності **«Комп'ютерні науки»** для магістратури та 1–3 курсу бакалаврату.

Далі відповідно до інформації з НП заповнюються позиції **«Вид дисципліни»** (обов'язкова, вибіркова чи факультативна), **«Форма навчання»**

(денна/заочна), «**Навчальний рік**», «**Семестр**», «**Кількість кредитів ECTS**», «**Мова викладання, навчання і оцінювання**», «**Форма заключного контролю**», та «**Викладачі**» (всі науково-педагогічні працівники, які забезпечують викладання даної дисципліни в даному навчальному році).

На другій сторінці робочої програми автори робочої програми вказуються в позиції «**Розробник(и)**», де вказуються їх ПІБ, науковий ступінь, вчене звання, посада і кафедра. Остаток сторінки заповнюється вже по затвердженню робочої програми, яка розглядається на засіданні кафедри та науково-методичної комісії факультету/інституту. Вона підписується відповідно завідувачем кафедри та головою науково-методичної комісії факультету/інституту, і затверджується заступником декана/директора інституту з навчальної роботи. Якщо дисципліна викладатиметься на іншому факультеті/інституті, необхідно ще погодити робочу програму із науково-методичною комісією цього факультету/інституту. Якщо дисципліна не є профільною для факультету/інституту, також необхідне погодження з науково-методичною комісією профільного факультету, і в такому випадку робиться додатковий запис про погодження. Для економічних та юридичних наук погодження робиться з предметною комісією з економічних та юридичних наук при Науково-методичній раді Університету.

Перейдемо до основної частини робочої програми.

Перша позиція — «**1. Мета дисципліни**». Обсяг тексту в даній позиції не має перевищувати 300 символів.

Друга позиція, «**2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності)**», може бути заповнена в двох форматах: «1. Успішне опанування курсу _____ ; 2. Знання теоретичних основ _____ », або «1. Знати: _____ ; 2. Вміти: _____ ; 3. Володіти елементарними навичками: _____ ». Інформація для заповнення позиції береться з наявної

структурно-логічної схеми освітньої програми (див. додаток А), яку можна знайти в пункті 2.2 відповідної освітньої програми. За схемою визначається, від яких інших дисциплін залежить дисципліна, для якої складається робоча програма. В якості попередніх вимог до опанування або вибору навчальної дисципліни можна вписати результати навчання з таблиці в позиції 5 робочої програми дисципліни, від якої залежить дана дисципліна.

Для позиції «**3. Анотація навчальної дисципліни**» обсяг тексту не має перевищувати 700 символів, для «**4. Завдання (навчальні цілі)**» — 500 символів. В завданнях вказуються компетентності випускника, на досягнення яких спрямована дана дисципліна. Загальні компетентності (ЗК), спеціальні або фахові компетентності (СК або ФК), та визначені спеціалізаціями компетентності (ВСК), які відповідають дисципліні, можна знайти у відповідному описі ОП — у пункті 4 «Матриця відповідності загальних та спеціальних програмних компетентностей компонентам освітньої програми». Текстовий опис компетентностей також наявний в навчальному плані, в пункті 1.6. Загальні компетентності вказуються у відповідності до переліку, затвердженого наказом МОН України від 01.06.2016 р. за №600 [10].

Наступна позиція — «**5. Результати навчання за дисципліною**». Тут необхідно заповнити таблицю (див. табл. 2.1).

Таблиця 2.1 — Позиція РПНД «Результати навчання за дисципліною»

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1.1	Знати....	лекція	Тест, 60% правильних відповідей	15%
1.2	Знати....	лекція	--/--	15%
2.1	Вміти	лабораторна робота	Звіт по лабораторній роботі	30%
4.1	Прийняти і обґрунтувати рішення....	лекція, семінарське заняття, самостійна робота	Кейс-задача	40%

Заплановані для дисципліни результати навчання треба описувати з детальністю, достатньою для розробки відповідних заходів оцінювання. Код результатів визначається у відповідності до належності результатів навчання до знань (1.х), вмінь (2.х), комунікації (3.х), автономності та відповідальності (4.х). В залежності від дисципліни, можуть бути відсутніми результати, які належать до спрямування комунікації і автономії та відповідальності. Формулювання результатів навчання здійснюється із використанням таких дієслів, як «знати», «вміти», «прийняти рішення», «використати» тощо. Детальні рекомендації можна подивитись в «Розроблення освітніх програм. Методичні рекомендації» [11] (затверджено наказом МОН України від 01.06.2016 за №600 [10]).

В колонці **«Форми (та/або методи і технології) викладання і навчання»** вказуються форми занять, навчальні заходи, методи, технології тощо, які будуть використані викладачем для забезпечення здобуття студентом відповідних результатів навчання: лабораторна робота, лекція, семінарське заняття, кейс-задача тощо.

В колонці **«Методи оцінювання та пороговий критерій оцінювання (за необхідності)»** вказується вид діяльності студента, за яким буде оцінюватись рівень досягнення результатів навчання: тест, задача, контрольна робота, реферат, звіт по лабораторній роботі тощо. В разі потреби вказується мінімальний пороговий рівень позитивної оцінки.

В колонці **«Відсоток у підсумковій оцінці з дисципліни»** вказується частка оцінки у відсотках за вказаний результат навчання у підсумковій оцінці з дисципліни. Сума за всіма результатами навчання має дорівнювати 100%

Наступна позиція — **«6. Співвідношення результатів навчання дисципліни із програмними результатами навчання»** (див. табл. 2.2), є обов'язковою для вибіркових дисциплін, які не входять до блоків спеціалізації. Результати навчання дисципліни беруться з попереднього пункту. Програмні результати навчання (ПРН), що відповідають дисципліні, беруться з пункту 5 **«Матриця забезпечення результатів навчання (ПРН) відповідними компонентами освітньої програми»** освітньої програми. Їх текстовий опис наявний в пункті 1.7 освітньої програми.

Таблиця 2.2 — Позиція РПНД «Співвідношення результатів навчання дисципліни із програмними результатами навчання»

Результати навчання дисципліни	1.1	1.2	2.1	4.1	4.2
Програмні результати навчання					
<i>(з опису освітньої програми)</i>		+			
			+		
			+		

Відміткою «+» показується, яким саме результатам навчання освітньої програми відповідають результати навчання дисципліни. Усі програмні результати та результати навчання дисципліни мають мати принаймні один «+» у відповідному рядку/стовпчику.

Сьома позиція «**7. Схема формування оцінки**», має три підпункти.

При заповненні позиції «**7.1 Форми оцінювання студентів**» зазначається перелік видів робіт та форм їх контролю та оцінювання із зазначенням результатів навчання, які на них мають бути оцінені, а також кількість балів/відсоток у підсумковій оцінці з дисципліни, пороговий рівень позитивної оцінки. Слід враховувати відповідні показники з таблиці в позиції 5 — результати навчання, форми викладання, порогові критерії оцінювання і відсоток у підсумковій оцінці.

Спочатку описується семестрове оцінювання. Наприклад:

Семестрове оцінювання:

1. *Контрольна робота з першої теми (тест): РН 1.1 — 10 балів/6 балів.*
2. *Реферат: РН 1.1, РН 2.1 — 20 балів/12 балів.*

3. *Контрольна робота з другої теми (тест): РН 1.2 — 10 балів/6 балів.*

4. *Лабораторні роботи (5 робіт): РН 2.2 — 4 бали/2 бали за кожну.*

Описується і підсумкове оцінювання — у формі іспиту, комплексного іспиту чи диференційованого заліку (залік виставляється за результатами роботи впродовж семестру і не передбачає додаткових заходів оцінювання для успішних студентів, а у випадку комплексного іспиту слід вказати питому вагу його складових). Фіксується форма оцінювання, максимальна кількість балів, які можуть бути отримані студентом (зазвичай 40 балів по 100-бальній шкалі), результати навчання, які будуть оцінюватись, форма проведення, види завдань та їх частка в сукупній оцінці, та мінімальний пороговий рівень екзаменаційної оцінки, за якої іспит вважається складеним. У випадку, коли студент на іспиту набрав менше вказаної кількості балів, вони не додаються до семестрової оцінки (незалежно від кількості балів, отриманих під час семестру), в екзаменаційній відомості у колонці «бали за іспит» ставиться «0», а в колонку «результуюча оцінка» переноситься лише кількість балів, отриманих під час семестру. Також слід прописати умови, які висуваються викладачами даної дисципліни як необхідна умова допуску до іспиту — наприклад, кількість балів, набрана студентом під час семестру. Мінімальна кількість балів не може бути меншою, ніж різниця рівня порогової оцінки (60 балів) і кількості балів, винесених на іспит (зазвичай 40) — якщо у студента менше 20 балів, він не може отримати позитивну оцінку. Викладач може визначити і вищий рівень мінімальної оцінки (як правило до 36 балів), якщо це аргументовано результатами навчання, які не виносяться на іспит. Якщо серед результатів навчання дисципліни є такі, які не можуть бути перевірені на іспиті, формулюються додаткові вимоги, наприклад: «Студент допускається до іспиту за умови виконання всіх (або деякий відсоток) передбачених планом лабораторних робіт».

В позиції «**7.2 Організація оцінювання**» зазначається порядок організації передбачених РПНД форм оцінювання із зазначенням орієнтовного графіку оцінювання. Коли дисципліна викладається два і більше семестрів, слід чітко вказати порядок формування заключної оцінки, у тому числі співвідношення між підсумковими формами контролю в семестрах.

Позиція «**7.3 Шкала відповідності оцінок**» (див. табл. 2.3).

Таблиця 2.3 — Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59
Зараховано / Passed	60-100
Не зараховано / Fail	0-59

При заповненні позиції «**8. Структура навчальної дисципліни. Тематичний план лекцій і семінарських/практичних/лабораторних занять**» (див. таблицю 2.4) слід вказати, про які саме заняття йде мова (лекцій і семінарських, лекцій і лабораторних, тощо). Можливий поділ дисципліни на розділи або частини. Крім тем лекційних/практичних/семінарських занять і назв лабораторних робіт вказуються також теми, винесені на самостійне вивчення. Вказуються також реферати та інші контрольні заходи. За умови наявності в навчальному плані може зазначатися курсова робота. Для лекцій, семінарських, практичних і лабораторних занять зазначається реальна, кратна двом, кількість годин. Час, відведений на інші види навчального навантаження студента, може бути кратний одній годині.

Таблиця 2.4 — Позиція «Структура навчальної дисципліни. Тематичний план занять»

№ п/п	Назва теми	Кількість годин		
		лекції	семінари/ практичні/ лабораторн і <i>вибрати необхідне</i>	Самостійна робота
<i>Назва розділу чи частини 1 (якщо здійснюється поділ)</i>				
1	Вступ. Тема 1 Назва. Самостійна робота:	2		2
2	Тема 2. Назва Самостійна робота:	3		2
	...			
	Тема XX. Назва Самостійна робота:	-	-	7
	<i>Контрольна робота 1</i>		X	
<i>Частина 2 Назва (за наявності)</i>				
	Тема XXX. Назва Самостійна робота:			
	<i>Підсумкова контрольна робота</i>	x	або x	
	ВСЬОГО	34	16	57

В позиції «9. Рекомендовані джерела» вказується до 10 фундаментальних, базових джерел, та, як правило, до 20 додаткових.

В «10. Додаткові ресурси» додаються посилання на електронні ресурси (не тільки відкриті), на яких розміщено додаткову інформацію щодо дисципліни

— приклади контрольних і екзаменаційних завдань, тематика рефератів, методичні вказівки по виконанню лабораторних робіт тощо.

По заповненню і затвердженню цих десяти позицій розробку РПНД можна вважати завершеною.

2.2 Задача автоматизації заповнення форми робочої програми

Запишемо позиції, заповнення яких передбачається при розробці РПНД. Автоматизація заповнення тих, що стосуються конкретики викладання даної дисципліни даним викладачем, неможлива. Відмітимо ті позиції, що передбачають звернення до інших документів:

1. Галузь знань — з ОП;
2. Спеціальність — з ОП;
3. Освітній рівень — з ОП;
4. Освітня програма — з ОП;
5. Спеціалізація — з ОП;
6. Вид дисципліни — з ОП;
7. Форма навчання — з ОП;
8. Навчальний рік — з НП;
9. Семестр — з НП;
10. Кількість кредитів ECTS — з НП;
11. Мова викладання;
12. Форма контролю — з ОП;
13. Викладачі;
14. Розробники;
15. Мета дисципліни;

16. Попередні вимоги;
17. Анотація;
18. Завдання (навчальні цілі) — з ОП (компетентності);
19. Результати навчання — з ОП (ПРН);
20. Співвідношення результатів навчання з програмними результатами — з ОП (ПРН);
21. Форма оцінювання студентів;
22. Організація оцінювання;
23. Схема формування оцінки — з ОП (залежить від форми оцінювання);
24. Структура навчальної дисципліни;
25. Кількість годин (лекцій, семінарів, самостійної роботи) — з НП;
26. Рекомендовані джерела;
27. Додаткові ресурси.

Позиції, що передбачають звернення до ОП чи НП — це позиції, заповнення яких можна автоматизувати, адже дана інформація не стосується конкретики викладання дисципліни, а фіксується при затвердженні ОП та НП, і від викладача, який розробляє робочу програму, не залежить.

Отже, найбільший можливий ступінь автоматизації — це отримання викладачем шаблону робочої програми для своєї дисципліни, який би вже містив всю інформацію, зафіксовану в ОП та НП.

Для того, щоб цього досягти, необхідні:

1. Дані з ОП та НП. Вони мають зберігатися в БД з можливістю редагування та оновлення у випадку внесення змін до цих двох документів.
2. Можливість завантажити шаблон робочої програми з даними з бази. Це передбачає створення додатку з інтерфейсом, через який викладачі

зможуть бачити структуру ОП та знайти там свою дисципліну. Шаблон робочої програми повинен бути коректно оформлений і завантажений в зручному для роботи форматі, такому як .docx.

2.3 Огляд та вибір технологій

Задача передбачає дві частини: створення БД та створення вебзастосунку для роботи з БД. Для цього необхідна архітектура, яка дала б змогу відокремити три компоненти: дані, логіку та інтерфейс програми. Для цієї мети ідеально підходить архітектурний шаблон «Модель-представлення-контролер» (Model View Controller, або MVC). Його суть полягає в розподілі системи на три компоненти — на моделі, представлення та на контролери (див. рис. 2.1):

- Модель — структура даних, керування ними, логіка додатку;
- Представлення — інтерфейс, вивід інформації від моделі до користувача;
- Контролер — ввід даних, керування моделлю через команди. [12]

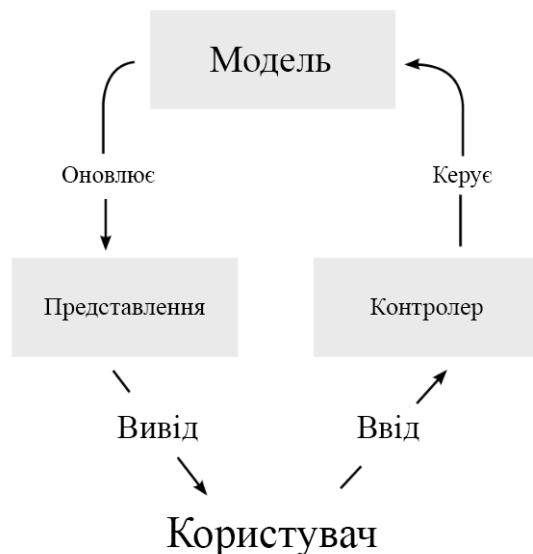


Рисунок 2.1 — Архітектурний шаблон MVC

Серед розповсюджених реалізацій MVC є такі фреймворки Java, як Struts, Spring MVC, JavaServer Faces, і фреймворк ASP.NET MVC, створений Microsoft.

Оскільки структура предметної області не є складною, вся необхідна інформація може бути записана у вигляді таблиць, та існує необхідність в незалежності даних від реалізації, для цієї задачі підходить реляційна модель даних. [13] На даний момент реляційна модель даних фактично є стандартом, а для керування реляційними базами даних широко застосовується мова Structured Query Language (SQL). SQL — друга найбільш розповсюджена у світі мова програмування (після JavaScript), її використовують 50% всіх розробників, і після 47 років використання цієї мови досвід роботи та підтримка спільноти є дуже широкими [14]. Тому обиратимемо систему керування базами даних (СКБД), яка дозволить розгорнути SQL-сервер. Дві найбільш розповсюджені СКБД для цього — MySQL та Microsoft SQL Server [15].

Через часткове попереднє знайомство та, головним чином, зручність інтеграції, було обрано технології з єдиним розробником — Microsoft SQL Server та ASP.NET MVC. Робота на ОС Windows 10 з використанням СКБД Microsoft SQL Server Management Studio 18 та середовища розробки Microsoft Visual Studio 2019 забезпечить максимальну сумісність, зручність і найменшу ймовірність помилок. Версія Visual Studio — безкоштовна Visual Studio Community 2019 16.9.3. Для обгортання БД в класи і взаємодію через .NET було застосовано технологію Entity Framework Core (EF Core) Database First — ORM Framework. ORM (Object Relational Mapping — об'єктно-реляційна проєкція) дозволяє автоматично згенерувати код, який відображає структуру існуючої БД, і працювати вже з ним для подальшої реалізації додатку на принципах об'єктно-орієнтованого програмування. Такий підхід називається Database First («Спершу база даних») [16]. Для роботи з документами у форматі .docx було використано бібліотеку DocX, розроблену Xceed Software. Вона є безкоштовною, відкритою версією бібліотеки Xceed Words, інтегрується у Visual Studio за допомогою вбудованого NuGet Package Manager, і містить всі методи, необхідні для виконання поставленої задачі [17]. Також, на відміну від Microsoft.Office.Interop.Word, вона не вимагає прив'язки до конкретної версії Microsoft Office 2013 та обов'язкового встановлення його на комп'ютер.

РОЗДІЛ 3

ПРИЗНАЧЕННЯ І ВИМОГИ ДО ЗАСТОСУНКУ

3.1 Цілі створення застосунку

Застосунок, призначений для автоматизації процесу розробки РПНД в ЗВО розробляється з метою:

- Забезпечення можливості адміністраторів централізовано редагувати оновлювати інформацію з ОП та НП.
- Забезпечення доступу користувачів до інформації з ОП та НП.
- Забезпечення можливості користувачів завантажувати шаблони для заповнення РПНД.

До інформації з ОП та НП входять:

- Структура: університет, факультети в університеті, ОП на факультетах, вибіркові блоки і дисципліни в ОП.
- Інформація про ОП: назва, галузь знань, спеціальність, рівень освіти.
- Програмні результати навчання та компетентності в освітній програмі, типи компетентностей (загальна/фахова).
- Інформація про дисципліну: код, назва, вибірковий блок, тип підсумкового контролю, семестр, кількість кредитів, загальна кількість годин, що виділяється, кількість годин на складові дисципліни (лекції/семінари/самостійна робота).
- ПРН та компетентності з ОП, що відповідають дисципліні.

Опишемо заплановану поведінку застосунку за допомогою діаграми прецедентів (рис. 3.1):



Рисунок. 3.1 — Діаграма прецедентів

3.2 Вимоги до застосунку

Застосунок, призначений для автоматизації процесу розробки РПНД в ЗВО має бути реалізований у вигляді вебзастосунку, підключеного до SQL-серверу, на якому зберігаються дані. БД має дозволяти додавати, редагувати і видаляти дані. При видаленні даного, на яке існує посилання в інших таблицях, відповідні

записи також мають бути видалені. Структура БД на SQL-сервері має бути відображена в коді програми: кожній таблиці відповідатиме клас з полями, що відповідають полям таблиці, а структура класів і зв'язки між ними мають відповідати зв'язкам між таблицями та їх властивостям.

Кожен клас матиме контролер, який відповідатиме за зв'язок між класом та представленнями вебсторінок, за обробку даних від адміністратора, та за обробку запитів користувачів. Кожен контролер матиме відповідні представлення: індекс (відображення записів таблиці), додавання, редагування, деталі, видалення запису. У індексі мають бути посилання на сторінки додавання, редагування та видалення записів з таблиці, доступні тільки адміністратору, та на деталі і індекси підпорядкованих їй таблиць, доступ до яких буде як у адміністратора, так і у користувача. Представлення повинні мати зручний і зрозумілий україномовний інтерфейс. За наявності зв'язку підпорядкування кожен рядок старшої таблиці (наприклад, ОП) містить кнопку переходу до молодшої (наприклад, вибіркового блоку), і при її натисканні демонструються тільки ті записи, що відповідають рядку, з якого був здійснений перехід (тобто буде отримано таблицю вибіркового блоку, що відповідають обраній ОП).

Індекс таблиці дисциплін матиме поруч із кожною дисципліною кнопку для завантаження шаблону робочої програми. При натисканні на неї буде здійснений запис інформації про дисципліну у коректно оформлений документ у форматі .docx, і завантажений даний файл.

Передбачається наявність підсистеми для автентифікації та авторизації. Користувачі зможуть створювати облікові записи, автентифікуватися, та керувати обліковими записами. Адміністратор повинен мати можливість надавати роль адміністратора іншим користувачам. Наявність двох ролей

передбачає наявність двох функціональних підсистем: адміністраторської та користувацької.

Отже, запишемо функції і задачі, що виконуються програмним засобом (див. табл. 3.1):

Таблиця 3.1 — Функції і задачі, що виконуються програмним засобом

Функція	Задача
Підсистема користувача	
Реєстрація	Введення інформації про себе
Реєстрація	Оновлення реєстрації
Навігація	Знаходження шуканої інформації в таблицях
Робота з БД	Перегляд інформації в таблицях, перегляд деталей про записи в таблицях
Робота з робочими програмами	Завантаження шаблону робочої програми для обраної дисципліни
Підсистема адміністратора	
Реєстрація	Введення інформації про себе
Реєстрація	Оновлення реєстрації
Керування роботою користувачів	Надання прав користувачам

Продовження таблиці 3.1

Функція	Задача
Підсистема адміністратора	
Керування роботою користувачів	Вилучення прав у користувачів
Навігація	Знаходження шуканої інформації в таблицях
Робота з БД	Перегляд інформації в таблицях, перегляд деталей про записи в таблицях
Робота з БД	Додавання інформації до таблиць
Робота з БД	Редагування інформації в таблицях
Робота з БД	Видалення інформації з таблиць
Робота з робочими програмами	Завантаження шаблону робочої програми для обраної дисципліни

Сформулюємо тепер технічні вимоги. Браузери: Вебзастосунок повинен коректно відображатися в інтернет-браузерах MS Internet Explorer 5.5 і вище, Mozilla 1.7 і вище, Opera 7.54 і вище, Google Chrome 53 і вище.

На довільні некоректні дії користувача, пов'язані з введенням невірних даних, незаповненням обов'язкових полів введення в формах та інші дії, які можуть бути оброблені застосунком, генеруються відповідні повідомлення про помилки українською мовою в межах загального дизайну сайту.

РОЗДІЛ 4

РЕАЛІЗАЦІЯ ЗАСТОСУНКУ

4.1 Структура бази даних

Розробка БД велась за допомогою Microsoft SQL Management Studio. Було спроектовано БД з логічною структурою, яка відповідає структурі ОП, відображає всі необхідні взаємозв'язки і включає всю інформацію з ОП та НП, що планується вносити в робочу програму (див. додаток Б).

БД містить таблиці, опис яких наведено в табл. 4.1:

Таблиця 4.1 — Перелік таблиць в базі даних

Таблиця	Опис
Universities	Університети
Faculties	Факультети
BranchesOfKnowledge	Галузі знань
Specializations	Спеціальності
EducationProgramsTypes	Рівні підготовки (бакалавр/магістр/тощо)
EducationPrograms	Освітні програми
Competences	Компетентності
CompetenceTypes	Тип компетентності (загальна/фахова)
EducationProgramsCompetences	Відповідності між компетентностями та освітніми програмами

Продовження таблиці 4.1

Таблиця	Опис
SelectiveBlocks	Вибіркові блоки
Subjects	Дисципліни
ControlTypes	Види заключного контролю
ProgramResults	Програмні результати навчання
SubjectProgramResults	Відповідності між дисциплінами і програмними результатами
SubjectCompetences	Відповідності між дисциплінами і компетентностями

Розглянемо детальний опис структури таблиць (див. табл. 4.2 – 4.16):

Таблиця 4.2 — Поля таблиці Universities

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Name	nvarchar(300)	Назва університету

Таблиця 4.3 — Поля таблиці Faculties

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Name	nvarchar(300)	Назва факультету
UniversityId	int	Ідентифікатор університету, до якого відноситься факультет

Таблиця 4.4 — Поля таблиці BranchesOfKnowledge

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Name	nvarchar(300)	Назва галузі знань

Таблиця 4.5 — Поля таблиці Specializations

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Name	nvarchar(300)	Назва спеціальності
BranchOfKnowledgeId	int	Ідентифікатор галузі знань

Таблиця 4.6 — Поля таблиці EducationProgramsTypes

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Name	nvarchar(300)	Назва рівня освіти

Таблиця 4.7 — Поля таблиці EducationPrograms

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Name	nvarchar(300)	Назва освітньої програми
SpecializationId	int	Ідентифікатор спеціальності
EducationProgramTypesId	int	Ідентифікатор рівня освіти

Продовження таблиці 4.7

Атрибут	Тип	Опис
FacultyId	int	Ідентифікатор факультету, до якого відноситься освітня програма

Таблиця 4.8 — Поля таблиці Competences

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Description	nvarchar(500)	Опис компетентності
CompetenceTypeId	int	Ідентифікатор типу компетентності

Таблиця 4.9 — Поля таблиці CompetenceTypes

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Type	nvarchar(300)	Тип компетентності

Таблиця 4.10 — Поля таблиці EducationProgramsCompetences

Атрибут	Тип	Опис
Id	int	Ідентифікатор
EducationProgramId	int	Ідентифікатор освітньої програми
CompetenceId	int	Ідентифікатор компетентності

Таблиця 4.11 — Поля таблиці SelectiveBlocks

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Name	nvarchar(300)	Назва вибіркового блоку
EducationProgramId	int	Ідентифікатор освітньої програми

Таблиця 4.12 — Поля таблиці Subjects

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Name	nvarchar(300)	Назва дисципліни
ControlTypeId	int	Ідентифікатор типу контролю
SelectiveBlockId	int	Ідентифікатор вибіркового блоку
Credits	int	Кількість кредитів ECTS
Semester	int	Семестр
Lectures	int	Години лекцій
Seminars	int	Години семінарів
SelfWork	int	Години самостійної роботи

Таблиця 4.13 — Поля таблиці ControlTypes

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Name	nvarchar(300)	Назва виду контролю

Таблиця 4.14 — Поля таблиці ProgramResults

Атрибут	Тип	Опис
Id	int	Ідентифікатор
Description	nvarchar(500)	Опис програмного результату
EducationProgramId	int	Ідентифікатор освітньої програми

Таблиця 4.15 — Поля таблиці SubjectProgramResults

Атрибут	Тип	Опис
Id	int	Ідентифікатор
SubjectId	int	Ідентифікатор дисципліни
ProgramResultId	int	Ідентифікатор програмного результату

Таблиця 4.16 — Поля таблиці SubjectCompetences

Атрибут	Тип	Опис
Id	int	Ідентифікатор
SubjectId	int	Ідентифікатор дисципліни
EducationProgramCompetenceId	int	Ідентифікатор зв'язку «освітня програма – компетентність»

Взаємозв'язки між таблицями є природними: наприклад, факультет належить до університету, ОП — до факультету, вибірковий блок — до ОП і так далі. Проте деякі зв'язки варто обґрунтувати. Наприклад, зв'язок між таблицею EducationPrograms та таблицею Competences реалізовано через проміжну таблицю EducationProgramsCompetences, на відміну від зв'язку між EducationPrograms та ProgramResults, який є безпосереднім. Це пояснюється тим, що ПРН, як правило, є унікальними для стандарту спеціальності, але загальні компетентності частіше можуть повторюватись між різними стандартами спеціальностей, тому для них було обрано зв'язок, який дозволить додавати ті самі компетентності до різних спеціальностей. Так само через проміжні таблиці реалізовано зв'язки між дисциплінами та компетентностями і ПРН, що їм відповідають: Subjects — SubjectProgramResults — ProgramsResults та Subjects — SubjectCompetences — EducationProgramsCompetences. Таким чином при додаванні різних дисциплін можна вибирати відповідні їм компетентності та ПРН, наявні в ОП.

При реалізації БД важливо врахувати питання видалення записів з таблиць, на які існують посилання в інших таблицях. В мові SQL можна вказати дію при

видаленні таблиці-предку: ON DELETE CASCADE (при видаленні таблиці-предку видалити його нащадків), ON DELETE SET NULL (замінити значення ключа таблиці-предку в таблиці-нащадку на порожнє значення — за умови, що це поле дозволяє порожні значення), ON DELETE SET DEFAULT (замінити значення ключа на наперед задане) та відсутність дії. При використанні опції ON DELETE CASCADE необхідно враховувати наявність циклів в структурі бази даних, адже це може привести до помилки, якщо декілька шляхів каскадного видалення сходяться. В Microsoft SQL Server Management Studio це контролюється автоматично, однак там не враховується характер зв'язків — за наявності будь-якого циклу в структурі БД програма з міркувань безпеки заборонить для кожної таблиці використати опцію каскадного видалення. Наприклад, хоч записи в таблицях SubjectProgramResults та SubjectCompetences можна видаляти (бо ключі даних таблиць не містяться в інших таблицях, тобто вони не мають нащадків), програма рахує їх зв'язки як цикл і заперечує каскадне видалення. Тому для реалізації каскадного видалення записів в таблицях для кожної таблиці було написано тригер, в якому вказано дії при видаленні — видалити всіх нащадків, після цього видалити саму таблицю. Наприклад, до таблиці EducationPrograms було додано наступний тригер:

```
CREATE TRIGGER dbo.DELETE_EducationProgram
    ON dbo.EducationPrograms
    INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;
    DELETE FROM SelectiveBlocks WHERE EducationProgramId IN (SELECT
    Id FROM DELETED)
    DELETE FROM EducationProgramsCompetences WHERE EducationProgramId
    IN (SELECT Id FROM DELETED)
    DELETE FROM ProgramResults WHERE EducationProgramId IN (SELECT Id
    FROM DELETED)
    DELETE FROM EducationPrograms WHERE Id IN (SELECT Id FROM
    DELETED)
END
```

Тобто порядок видалення ОП: видалити всі її вибіркові блоки (які мають аналогічні тригери для видалення всіх підпорядкованих дисциплін), видалити її зв'язки з компетентностями, видалити її ПРН, потім видалити саму ОП.

4.2 Структура системи вебзастосунку

Архітектура вебзастосунку розроблялась за шаблоном «Модель — представлення — контролер» (Model View Controller, або MVC).

В якості моделей в структурі вебзастосунку виступають класи, які відповідають таблицям з БД — 15 класів (відповідні файли наявні на рис. 4.1).

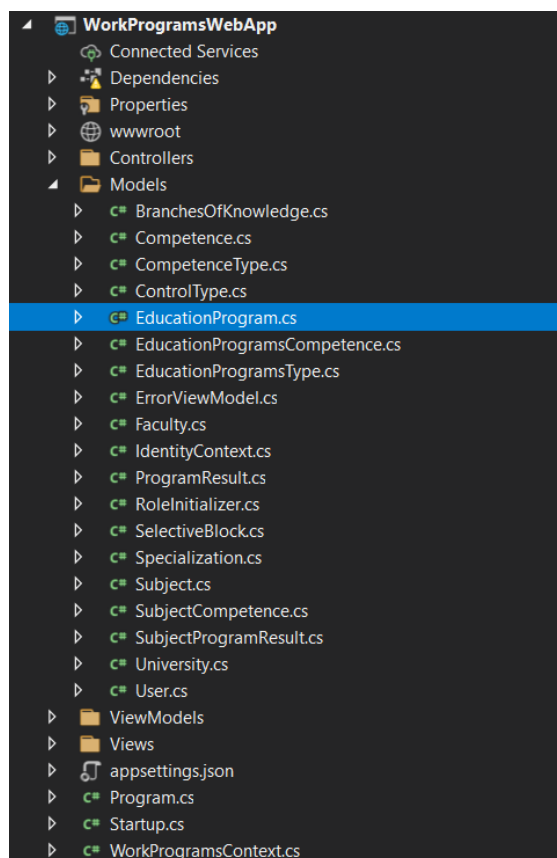


Рисунок 4.1 — Файлова структура проекту

Розглянемо, наприклад, клас EducationProgram, що відповідає таблиці EducationPrograms з БД:

```
namespace WorkProgramsWebApp
{
    public partial class EducationProgram
    {
        public EducationProgram()
        {
            ProgramResults = new HashSet<ProgramResult>();
            SelectiveBlocks = new HashSet<SelectiveBlock>();
            EducationProgramsCompetences = new
HashSet<EducationProgramsCompetence>();
        }

        [Required(ErrorMessage = "Поле не повинно бути порожнім")]
        [Display(Name = "Освітня програма")]
        public int Id { get; set; }

        [Display(Name = "Освітня програма")]
        [Required(ErrorMessage = "Поле не повинно бути порожнім")]
        public string Name { get; set; }

        [Display(Name = "Спеціальність")]
        [Required(ErrorMessage = "Поле не повинно бути порожнім")]
        public int SpecializationId { get; set; }

        [Display(Name = "Рівень")]
        [Required(ErrorMessage = "Поле не повинно бути порожнім")]
        public int EducationProgramsTypesId { get; set; }

        [Display(Name = "Факультет")]
        [Required(ErrorMessage = "Поле не повинно бути порожнім")]
        public int FacultyId { get; set; }

        public virtual EducationProgramsType
EducationProgramsTypes { get; set; }

        public virtual Faculty Faculty { get; set; }

        public virtual Specialization Specialization { get; set; }

        public virtual ICollection<ProgramResult> ProgramResults {
get; set; }
    }
}
```

```
        public virtual ICollection<SelectiveBlock> SelectiveBlocks
    { get; set; }

        public virtual ICollection<EducationProgramsCompetence>
    EducationProgramsCompetences { get; set; }

    }
}
```

Поля даного класу відповідають полям таблиці EducationPrograms: Id, Name, SpecializationId, EducationProgramsTypesId, FacultyId. Дані поля є обов'язковими, тому за допомогою анотацій вказано, що користувач не зможе залишити поле порожнім, бо отримає помилку «Поле не повинно бути порожнім». Також наявні підписи з назвами полів українською для коректного відображення у представленнях. Поля EducationProgramsTypes, Faculty, Specialization відповідають зв'язкам, у яких EducationProgram є нащадком, а колекції ProgramResults, SelectiveBlocks та EducationProgramsCompetences відповідають нащадкам.

Кожен клас має набір відповідних представлень (рис. 4.2).

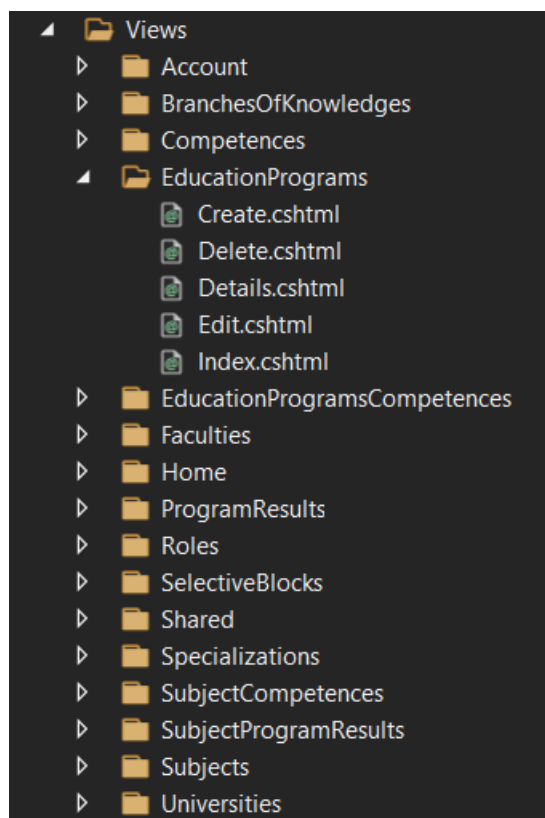


Рисунок 4.2 — Представлення у проєкті

Кожне представлення відповідає за окрему функцію:

- Index — відображення записів у таблиці;
- Create — додавання запису до відповідної таблиці;
- Edit — редагування записів;
- Details — відображення детальної інформації про запис;
- Delete — вилучення записів з таблиці.

Вебзастосунок організовано таким чином, що користувач починає з представлення Index таблиці-предку, і біля кожного запису в ній є кнопка переходу на аналогічне представлення Index таблиці-нащадку. Тобто починаючи з представлення Index моделі університетів, можна перейти від будь-якого університету в таблиці до списку його факультетів, від списку факультетів — до списку ОП кожного факультету і так далі.

Розглянемо, наприклад, представлення Index моделі EducationProgram, код якого можна знайти в додатку В. На екран користувача виводиться таблиця з назвами ОП, їх рівнем, спеціальністю, та кнопка переходу на представлення Index моделі SelectiveBlocks, яке видасть наявні в даній ОП вибіркові блоки (рис. 4.3).

Університети Галузі знань Компетентності Аккаунт

Освітні програми: Факультет комп'ютерних наук та кібернетики

Компетентності за освітніми програмами

Освітня програма	Рівень	Спеціальність	
Інформатика	бакалавр	122 «Комп'ютерні науки»	Вибіркові блоки Програмні результати
Інформатика	магістр	122 «Комп'ютерні науки»	Вибіркові блоки Програмні результати

Рисунок 4.3 — представлення Index моделі EducationProgram для ОП на факультеті комп'ютерних наук та кібернетики (вигляд для користувача)

Аналогічно є посилання на таблицю з усіма ПРН, що відповідають даній ОП, та на таблицю з відповідностями різних ОП компетенціям. Якщо сторінку переглядає адміністратор, на екран також виводяться опції додавання нових ОП та редагування і видалення наявних ОП (рис. 4.4).

Освітні програми: Факультет комп'ютерних наук та кібернетики

[Компетентності за освітніми програмами](#)
[Додати освітню програму](#)

Освітня програма	Рівень	Спеціальність	
Інформатика	бакалавр	122 «Комп'ютерні науки»	<ul style="list-style-type: none"> ·Вибіркові блоки ·Програмні результати ·Редагувати ·Видалити
Інформатика	магістр	122 «Комп'ютерні науки»	<ul style="list-style-type: none"> ·Вибіркові блоки ·Програмні результати ·Редагувати ·Видалити

Рисунок 4.4 — представлення Index моделі EducationProgram для ОП на факультеті комп'ютерних наук та кібернетики (вигляд для адміністратора)

Дії користувача обробляються наступною складовою архітектури — контролером. Контролери містять відповідні до представлень методи: Index, Create, Edit, Details, Delete. Проект містить контролери на кожну з моделей (рис. 4.5).

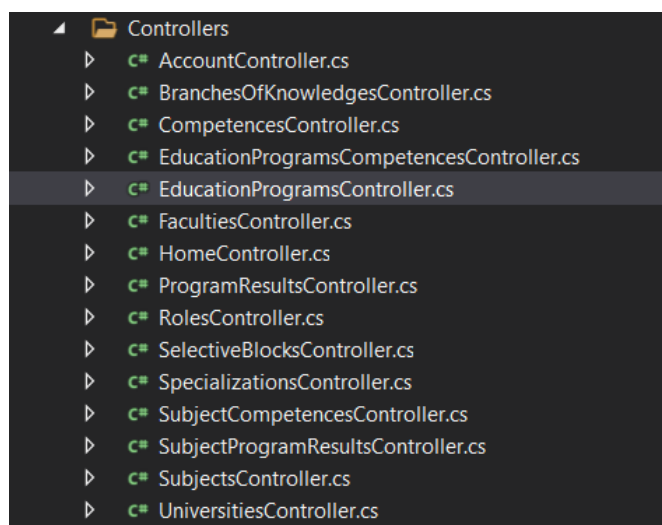


Рисунок 4.5 — Контролери у проекті

Розглянемо, наприклад, метод `Index` контролера `EducationProgramsController`:

```
public async Task<IActionResult> Index(int? id, string? name)
{
    if (id == null) return RedirectToAction("Index",
"Faculties");
    ViewBag.FacultyId = id;
    ViewBag.FacultyName = name;
    var educationProgramsByFaculty =
_context.EducationPrograms.Where(e => e.FacultyId == id).Include(e
=> e.EducationProgramsTypes).Include(e => e.Faculty).Include(e =>
e.Specialization);
    var ep = from e in educationProgramsByFaculty
            select e;
    ep = ep.OrderBy(e => e.Name);
    return View(await ep.AsNoTracking().ToListAsync());
}
```

Цей метод викликається для одного з факультетів. Він приймає на вхід параметри `id` та `name` — відповідно ідентифікатор та назва факультету, працює з моделлю `EducationProgram`, і повертає представлення, яке містить список ОП на цьому факультеті, відсортований за назвою ОП. При виклику з порожнім `id` він повертає представлення `Index` факультетів. Схожа поведінка у методу `Create`: він викликається для додавання ОП до певного факультету, і дозволяє заповнити всі поля новоствореної ОП, крім поля факультету, яке автоматично заповнюється ідентифікатором факультету, з якого було викликано метод.

Генерація документу з шаблоном РПНД відбувається завдяки методу `ExportToWord` контролера `SubjectsController`. Його код винесено в додаток Г. Метод копіює заздалегідь підготовлений шаблон РПНД, в якому позиції, які мають бути заповнені, записані у вигляді тегів `<НАЗВАПОЗИЦІЇ>`. Метод звертається до БД і замінює теги на текст з відповідними даними. Компетентності дисципліни вставляються у вигляді списку, а для вставки ПРН у таблицю відповідності результатів навчання ПРН додаються нові рядки.

Приклад шаблону, згенерованого для дисципліни «Інструментальні засоби та технології програмування» ОП «Інформатика» факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка, можна знайти в додатку Д.

Важливою складовою вебзастосунку є система автентифікації і авторизації. Вона дозволяє вести облік користувачів і надавати адміністраторські права доступу тим, хто вповноважений змінювати дані в БД. Для її реалізації було використано вбудований в ASP.NET функціонал для автентифікації і авторизації, і створено окрему БД для зберігання інформації про зареєстрованих користувачів (рис. 4.6).

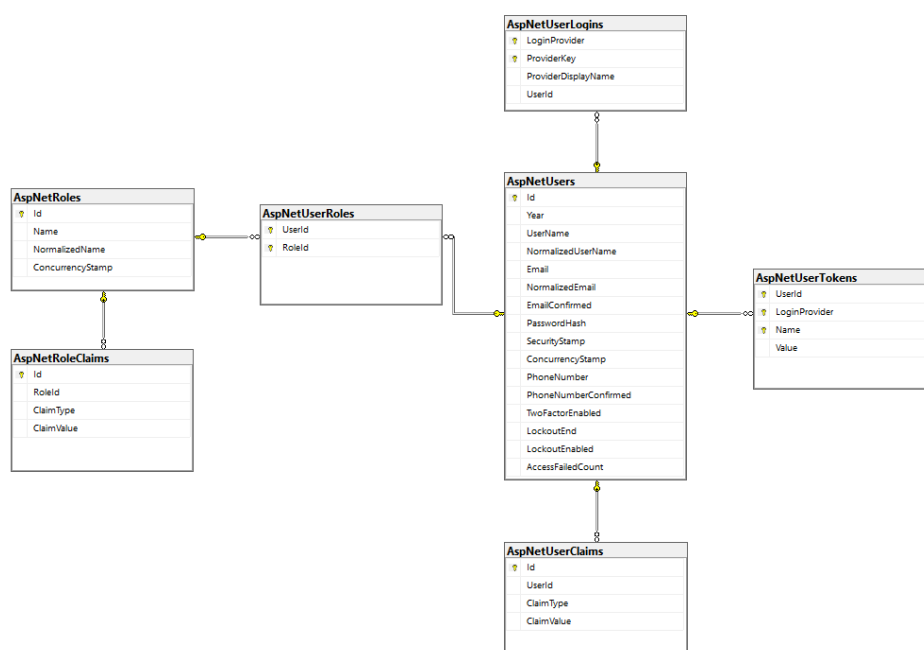


Рисунок 4.6 — БД для системи автентифікації і авторизації

Було створено моделі: `User`, `IdentityContext` та `ErrorViewModel`; представлення: `Login`, `Register`, `AccessDenied (Account)`, `Edit`, `Index`, `UserList (Roles)`; контролери: `AccountController`, `RolesController`. Вони взаємодіють з БД для безпечного зберігання даних окремо від інтерфейсу — наприклад, паролі

зберігаються у БД у вигляді хешів. Доступ до кожного контролера прописується залежно від ролі — наприклад, доступ до контролера RolesController, який відповідає за можливість редагування адміністратором прав доступу користувачів, є тільки у адміністратора, а до всіх інших контролерів є доступ і у користувача, і у адміністратора, але не у аноніма (незарєєстрованого користувача), і не у користувача, позбавленого адміністратором будь-яких прав. Проте користувач, хоч і матиме доступ до контролерів, не зможе викликати методи, пов'язані зі зміною даних в БД — це доступно тільки адміністратору.

РОЗДІЛ 5

ІНСТРУКЦІЯ З ВИКОРИСТАННЯ ЗАСТОСУНКУ

5.1 Інструкція для користувача

Вебзастосунок для автоматизації розробки РПНД вимагає реєстрації для користування. При відкритті вебсторінки незареєстрований користувач бачить опції увійти чи зареєструватись (рис. 5.1).

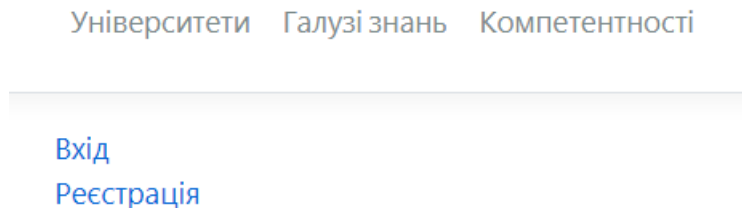


Рисунок 5.1 — Головна сторінка для незареєстрованого користувача

Якщо він намагатиметься перейти за посиланнями верхнього меню, він буде перенаправлений назад. Сторінку реєстрації та сторінку входу для зареєстрованого користувача можна побачити на рис. 5.2 та рис. 5.3 відповідно.

Реєстрація нового користувача

Email

Пароль

Підтвердження паролю

Рисунок 5.2 — Вікно реєстрації

Вхід

Email

Пароль

 Запам'ятати[Реєстрація](#)

Рисунок 5.3 — Вікно входу

Для реєстрації необхідна дійсна електронна пошта, а пароль має містити одну велику літеру, одну цифру та один символ, який не є буквено-цифровим. Після реєстрації, для роботи з застосунком користувач має очікувати надання адміністратором прав доступу. За відсутності прав користувач при спробі відкрити посилання з меню побачить повідомлення про відсутність прав (рис. 5.4).

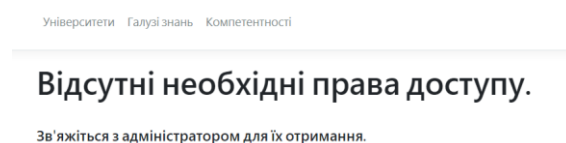


Рисунок 5.4 — повідомлення про відсутність прав доступу

Для виходу з зареєстрованого облікового запису необхідно перейти за посиланням «Акаунт» і натиснути на кнопку «Вихід» (рис. 5.5).

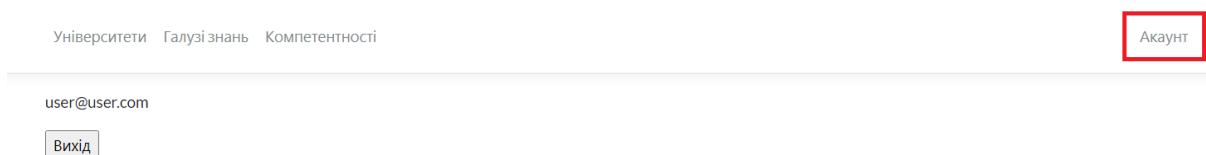


Рисунок 5.5 — Вікно «Акаунт»

При переході за посиланням «Галузь знань» верхнього меню відкриється список галузей знань, за якими здійснюється підготовка фахівців у наявних в базі освітніх програмах (рис. 5.6).

Галузі знань	
Галузь знань	Спеціальності
11 «Математика та статистика»	Спеціальності
12 «Інформаційні технології»	Спеціальності

Рисунок 5.6 — Список галузей знань

Біля кожної галузі знань є посилання на сторінку зі списком спеціальностей, які входять до даної галузі знань (рис. 5.7).

12 «Інформаційні технології»: Спеціальності	
Спеціальність	
121 «Інженерія програмного забезпечення»	
122 «Комп'ютерні науки»	
123 «Комп'ютерна інженерія»	
124 «Системний аналіз»	
125 «Кібербезпека»	

Рисунок 5.7 — Список спеціальностей для галузі 12 «Інформаційні технології»

При переході за посиланням «Компетентності» верхнього меню відкриється загальний список з усіма компетентностями, наявними в базі (рис. 5.8).

Компетентності

Компетентності за освітніми програмами

Опис	Тип
ЗК1. Здатність до абстрактного мислення, аналізу та синтезу	Загальна
ЗК2. Здатність застосовувати знання у практичних ситуаціях	Загальна
ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності	Загальна
ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово	Загальна
ЗК5. Здатність спілкуватися іноземною мовою	Загальна
ЗК6. Здатність вчитися й оволодівати сучасними знаннями	Загальна
ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел	Загальна

Рисунок 5.8 — Загальний список компетентностей

Для перегляду списку компетентностей за ОП (рис. 5.9) необхідно перейти за відповідним посиланням над загальним списком компетентностей. Натискання на «Освітня програма» та «Компетентність» встановлює порядок сортування за відповідним полем.

Компетентності за освітніми програмами

Загальний список компетентностей

Освітня програма	Рівень	Компетентність
Інформатика	бакалавр	ЗК1. Здатність до абстрактного мислення, аналізу та синтезу
Інформатика	бакалавр	ЗК2. Здатність застосовувати знання у практичних ситуаціях
Інформатика	бакалавр	ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності
Інформатика	бакалавр	ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово
Інформатика	бакалавр	ЗК5. Здатність спілкуватися іноземною мовою
Інформатика	бакалавр	ЗК6. Здатність вчитися й оволодівати сучасними знаннями
Інформатика	бакалавр	ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел
Інформатика	бакалавр	ЗК8. Здатність генерувати нові ідеї (креативність)

Рисунок 5.9 — Список компетентностей за освітніми програмами

Пошук факультетів, освітніх програм, вибіркового блоків та дисциплін починається зі сторінки «Університети» (рис. 5.10).

Університети	Галузі знань	Компетентності	Акаунт
Університети			
Назва університету			
Київський національний університет імені Тараса Шевченка			Факультети
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»			Факультети

Рисунок 5.10 — Список університетів

Кожен університет має посилання на список його факультетів (рис. 5.11), кожен факультет у списку має посилання на список його ОП (рис. 5.12), кожна ОП — на список її вибіркових блоків (рис. 5.13) та список її ПРН (рис. 5.14).

Університети	Галузі знань	Компетентності	Акаунт
Факультети: Київський національний університет імені Тараса Шевченка			
Назва факультету			
Факультет комп'ютерних наук та кібернетики			Освітні програми
Механіко-математичний факультет			Освітні програми

Рисунок 5.11 — Список факультетів Київського національного університету імені Тараса Шевченка

Університети Галузі знань Компетентності Акаунт

Освітні програми: Факультет комп'ютерних наук та кібернетики

Компетентності за освітніми програмами

Освітня програма	Рівень	Спеціальність	
Інформатика	бакалавр	122 «Комп'ютерні науки»	Вибіркові блоки Програмні результати
Інформатика	магістр	122 «Комп'ютерні науки»	Вибіркові блоки Програмні результати

Рисунок 5.12 — Список освітніх програм факультету комп'ютерних наук та кібернетики

Університети Галузі знань Компетентності Акаунт

Вибіркові блоки: Інформатика

Вибірковий блок

Обов'язкові компоненти ОП	Дисципліни
Інтелектуальні Інформаційні технології	Дисципліни
Теорія та технологія програмування	Дисципліни
Інформаційні технології та системи	Дисципліни

Рисунок 5.13 — Список вибірових блоків ОП «Інформатика»

Університети Галузі знань Компетентності Акаунт

Програмні результати навчання: Інформатика

Програмний результат навчання

ПРН1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.
ПРН2. Використовувати сучасний математичний апарат неперервного та дискретного аналізу, лінійної алгебри, аналітичної геометрії, в професійній діяльності для розв'язання задач теоретичного та прикладного характеру в процесі проектування та реалізації об'єктів інформатизації.
ПРН3. Використовувати знання закономірностей випадкових явищ, їх властивостей та операцій над ними, моделей випадкових процесів та сучасних програмних середовищ для розв'язування задач статистичної обробки даних і побудови прогнозних моделей.
ПРН4. Використовувати методи обчислювального інтелекту, машинного навчання, нейромережевої та нечіткої обробки даних, генетичного та еволюційного програмування для розв'язання задач розпізнавання, прогнозування, класифікації, ідентифікації об'єктів керування тощо.

Рисунок 5.14 — Список ПРН ОП «Інформатика»

Кожен вибірковий блок має список його дисциплін (рис. 5.15), а кожна дисципліна має посилання на списки її ПРН та компетентностей з ОП.

Університети Галузі знань Компетентності Акаунт

Дисципліни у блоці: Обов'язкові компоненти ОП

Дисципліна	Семестр	Форма контролю	Кредити ECTS	Кількість годин	Лекції	Семинари	Самостійна робота	
Інструментальні середовища та технології програмування	4	іспит	5	150	36	38	76	Програмні результати Компетентності Шаблон робочої програми

Рисунок 5.15 — Список дисциплін у блоці обов'язкових компонент ОП «Інформатика»

Біля кожної дисципліни є кнопка «Шаблон робочої програми» — при її натисканні браузер завантажить файл формату .docx з шаблоном РПНД, в якому будуть заповнені всі поля, що спираються на ОП та НП (рис. 5.16). Далі розробник РПНД самостійно заповнює позиції, що безпосередньо стосуються учбового матеріалу та викладання даної дисципліни.

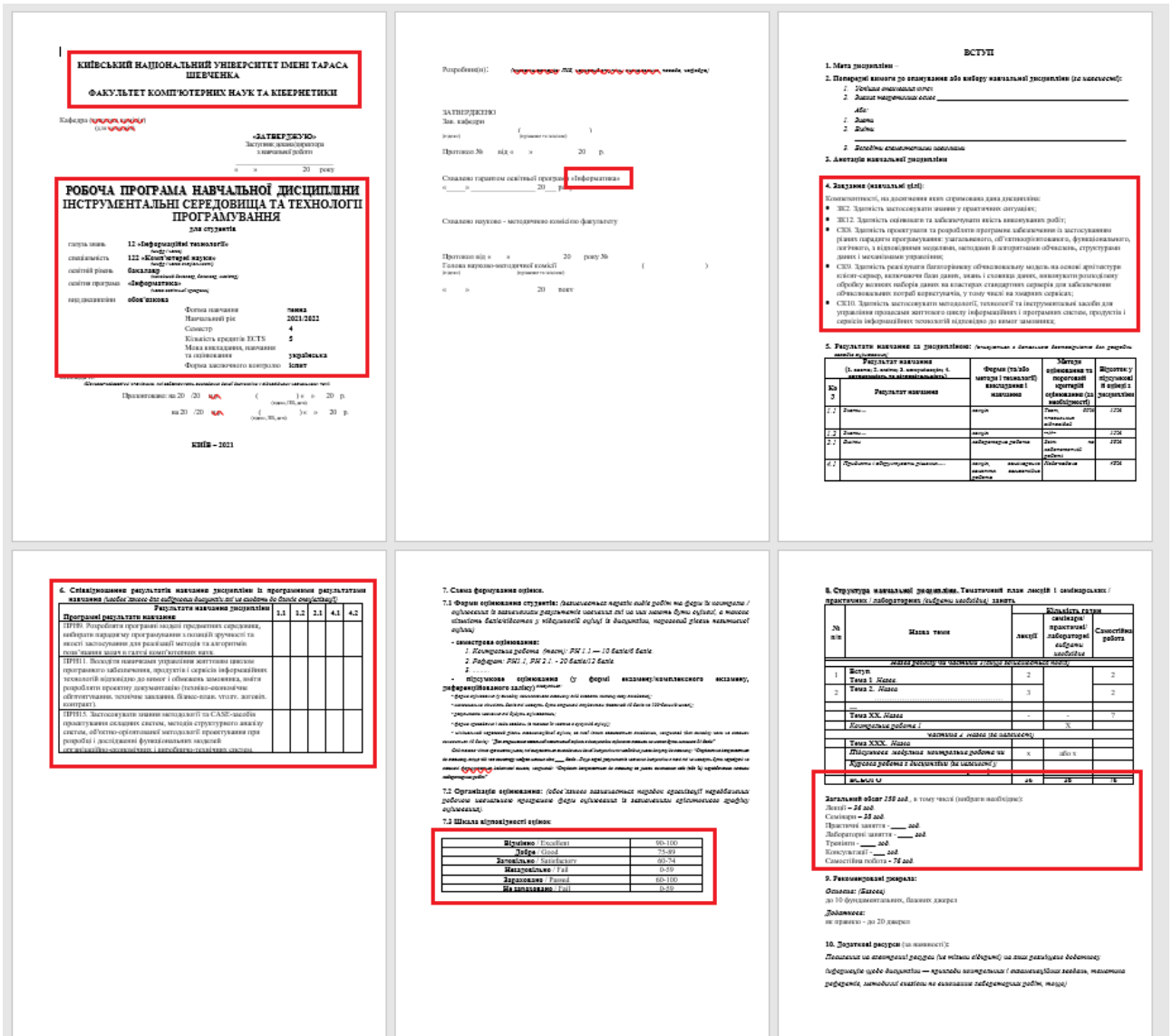


Рисунок 5.16 — Поля РПНД «Інструментальні середовища та технології програмування», заповнені автоматично

5.2 Інструкція для адміністратора

Після отримання прав доступу адміністратора, зареєстрований користувач побачить у мені посилання «Користувачі» на сторінку зі списком користувачів (рис. 5.17).

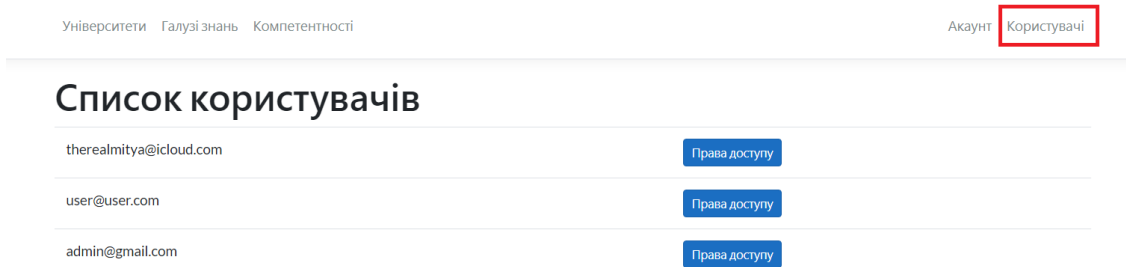


Рисунок 5.17 — Список користувачів, який бачить адміністратор

Біля адреси кожного користувача є кнопка «Права доступу», яка відкриває сторінку для контролю його прав доступу (рис. 5.18).

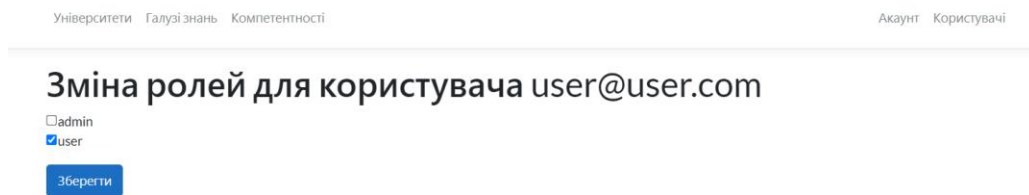


Рисунок 5.18 — Вікно зміни ролей користувача з адресою user@user.com

Користувач з роллю user може переглядати всі дані і завантажувати шаблони РПНД, а користувач з роллю admin має всі можливості звичайного користувача, і також може вносити зміни в БД та керувати ролями інших користувачів. Користувач без ролі не має доступу до жодної сторінки, крім сторінки свого акаунту з кнопкою виходу.

Навігація по сайту у адміністратора така сама, як і у звичайного користувача. Однак адміністратор побачить біля кожного запису в таблиці посилання «Редагувати» та «Видалити», а також посилання над кожною таблицею для додавання запису (рис. 5.19).

Університети Галузі знань Компетентності Акаунт Користувачі

Освітні програми: Факультет комп'ютерних наук та кібернетики

[Компетентності за освітніми програмами](#)
[Додати освітню програму](#)

Освітня програма	Рівень	Спеціальність	
Інформатика	бакалавр	122 «Комп'ютерні науки»	<ul style="list-style-type: none"> ·Вибіркові блоки ·Програмні результати ·Редагувати ·Видалити
Інформатика	магістр	122 «Комп'ютерні науки»	<ul style="list-style-type: none"> ·Вибіркові блоки ·Програмні результати ·Редагувати ·Видалити

Рисунок 5.19 — Елементи адміністратора роботи з БД

При додаванні запису до списку нащадків певного запису в таблиці-предку, поле з предком заповнюється автоматично. Тобто, наприклад, при додаванні ОП до списку ОП на факультеті комп'ютерних наук та кібернетики, поле «факультет» заповнюється автоматично і не показується у вікні створення (рис. 5.20).

Університети Галузі знань Компетентності Акаунт Користувачі

Додати освітню програму до: Факультет комп'ютерних наук та кібернетики

Освітня програма

Спеціальність

Рівень

[Назад до списку](#)

Рисунок 5.20 — Вікно створення ОП на факультеті комп'ютерних наук та кібернетики

Якщо це поле необхідно змінити, тобто, наприклад, перенести ОП з одного факультету на інший, це можна зробити через вікно редагування (рис. 5.21). Всі інші поля адміністратор заповнює вручну або вибирає опцію зі списку. При додаванні та редагуванні записів обов’язково мають бути заповненими всі поля, крім полів таблиці дисциплін з інформацією про кількість годин — їх можна залишити порожніми.

Університети Галузі знань Компетентності Акаунт Користувачі

Редагувати освітню програму

Освітня програма

Спеціальність

Рівень

Факультет

[Назад до списку](#)

Рисунок 5.21 — Вікно редагування ОП

ВИСНОВКИ

Метою даної кваліфікаційної роботи було проектування та розробка програмного засобу для автоматизації процесу розробки РПНД, в рамках загальної мети спрощення та оптимізації процесу організації освітньої діяльності університету. Для досягнення цієї мети було успішно виконано наступні завдання:

- Розглянуто питання наявності засобів автоматизації процесу розробки РПНД в інших ЗВО України;
- Розглянуто процес розробки РПНД, сформовано інструкцію до заповнення форми робочої програми, визначено способи автоматизувати даний процес;
- Висунуто вимоги до програмного засобу автоматизації розробки РПНД, обрано технології для його розробки;
- Спроектовано та програмно реалізовано БД з інформацією про ОП на факультетах ЗВО, дисципліни, що до них входять, та відповідності їм програмних результатів та компетентностей з ОП;
- Створено вебзастосунок для роботи з БД та генерації для кожної дисципліни шаблону робочої програми, заповненого інформацією з бази.

Розроблений програмний засіб може застосовуватись викладачами при розробці робочих програм своїх дисциплін. До бази адміністратором централізовано вносяться зміни освітніх програм та навчальних планів, а викладачі отримують шаблони робочих програм, які містять інформацію з освітньої програми з усіма внесеними змінами, без потреби звіряти ці дані вручну. З пришвидшенням темпів розвитку суспільства, для відповідності сучасним стандартам індустрії університети мають все частіше проводити

реорганізацію освітнього процесу, вносячи зміни в ОП, НП та РПНД. Використання такого програмного засобу значно зекономить кількість часу та зусиль, необхідних для створення РПНД та перевірки коректності заповнених РПНД, що сприятиме більшій концентрації розробників на самій суті дисципліни та організації її викладання, що, в свою чергу, підвищить рівень якості освіти.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

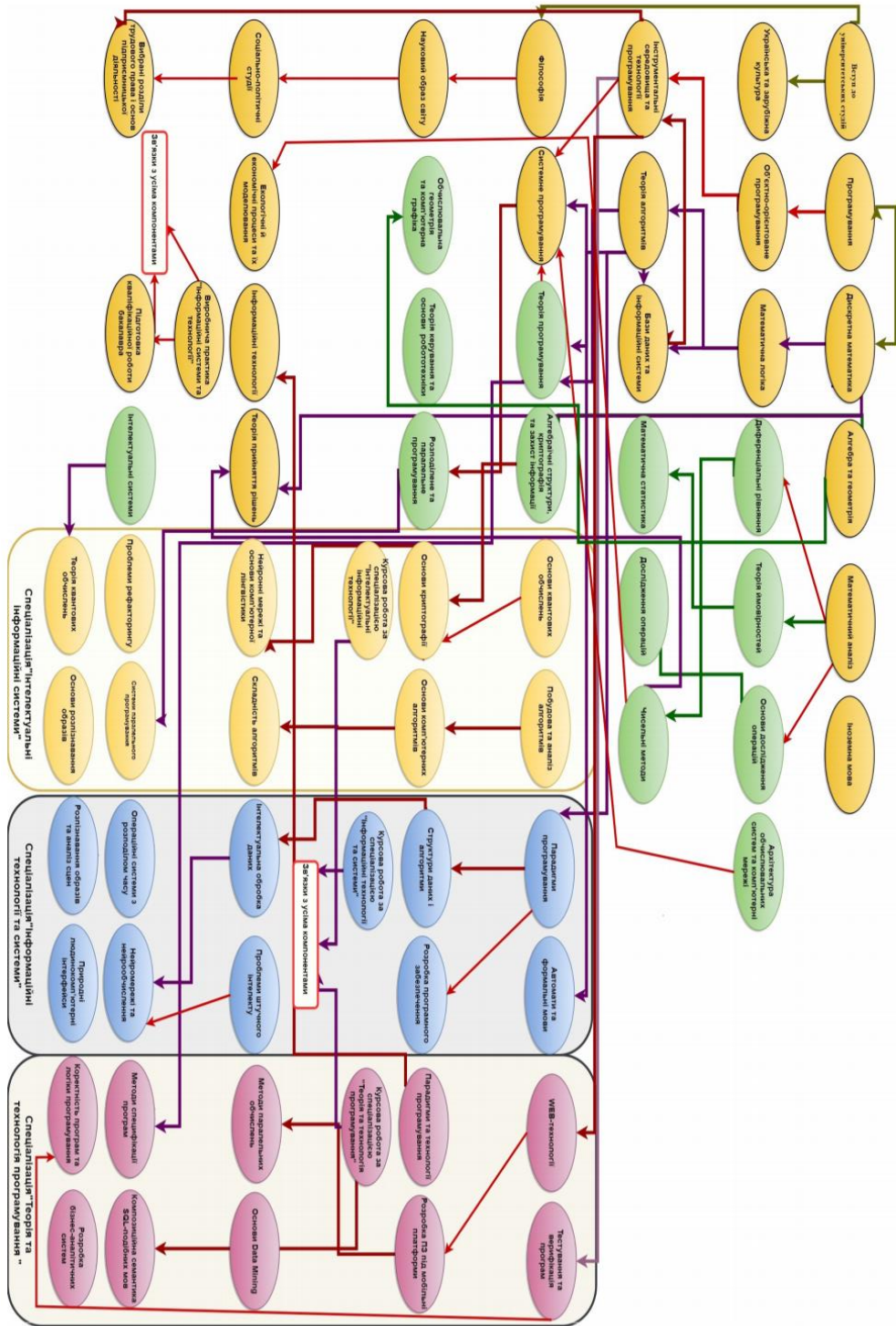
1. Система автоматизації Київського національного університету імені Тараса Шевченка (Triton) [Електронний ресурс]. Режим доступу до ресурсу: <https://student.triton.knu.ua/>
2. Вебсайт факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка, розділ «Навчальні плани» [Електронний ресурс]. Режим доступу до ресурсу: <http://csc.knu.ua/uk/curriculum>
3. Вебсайт кафедри теорії та технології програмування факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка, розділ «Освітні програми» [Електронний ресурс]. Режим доступу до ресурсу: <https://tp.infosoft.ua/%d0%9e%d1%81%d0%b2%d1%96%d1%82%d0%bd%d1%96-%d0%bf%d1%80%d0%be%d0%b3%d1%80%d0%b0%d0%bc%d0%b8/>
4. Вебсайт кафедри прикладної статистики програмування факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка, розділ «Навчальний процес» [Електронний ресурс]. Режим доступу до ресурсу: <http://applstat.univ.kiev.ua/ukr/?templ=studies>
5. Вебсайт Конструкторського бюро інформаційних систем [Електронний ресурс] / Конструкторське бюро інформаційних систем Національного технічного університету України «Київський Політехнічний Інститут» – 2021. – Режим доступу до ресурсу: <http://kbis.kpi.ua/>.
6. Інформаційно-телекомунікаційна система «Електронний кампус» [Електронний ресурс] / Конструкторське бюро інформаційних систем

- Національного технічного університету України «Київський Політехнічний Інститут». – 2021. – Режим доступу до ресурсу: <https://ecampus.kpi.ua/>.
7. Електронний кампус: стан і перспективи [Електронний ресурс] / О.Д.Фіногенов, директор КБ ІС, к.т.н., доц.. – 2019. – Режим доступу до ресурсу: <https://kpi.ua/2019-kp20-2>.
 8. Наказ ректора від 11.08.2017 р. за № 729-32 «Про запровадження в освітній та інформаційний процес форм опису освітньо-професійної (освітньо-наукової) програми, структурних вимог до інформаційного пакету, форм робочої навчальної програми дисципліни і форми представлення інформації про кваліфікацію науково-педагогічного працівника». Режим доступу до ресурсу: http://nmc.univ.kiev.ua/docs/Nakaz_Form_Doc-729-32_11-08-2017.pdf
 9. Постанова Кабінету Міністрів України від 29.04.15 р. за № 266 «Про затвердження переліку галузей знань і спеціальностей, за якими здійснюється підготовка здобувачів вищої освіти». Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/266-2015-%D0%BF>
 10. Наказ Міністерства освіти і науки України від 01.06.2016 р. за № 600 «Про затвердження та введення в дію Методичних рекомендацій щодо розроблення стандартів вищої освіти». Режим доступу до ресурсу: https://osvita.ua/legislation/Vishya_osvita/51506/
 11. «Методичні рекомендації щодо розроблення стандартів вищої освіти». Режим доступу до ресурсу: <https://osvita.ua/doc/files/news/515/51506/Recom.doc>

12. Reenskaug T. The Model-View-Controller (MVC) Its Past and Present. / Trygve Reenskaug. – 2003. Режим доступа до ресурсу: https://folk.universitetetioslo.no/trygver/2003/javazone-jaoo/MVC_pattern.pdf
13. Codd E. F. A Relational Model of Data for Large Shared Data Banks / Edgar Frank Codd. // Communications of the ACM. – 1970. – С. 377–387. Режим доступа до ресурсу: <https://web.archive.org/web/20070612235326/http://www.acm.org/classics/nov95/toc.html>
14. SQL is 43 years old - here's 8 reasons we still use it today [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <https://blog.sqlizer.io/posts/sql-43/>.
15. MySQL vs. MSSQL—Performance and Main Differences Between Database and Servers [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.dnsstuff.com/mysql-vs-mssql-performance>.
16. Entity Framework Core [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://habr.com/ru/company/otus/blog/500012/>.
17. Xceed Software — DocX Library [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://github.com/xceedsoftware/DocX>

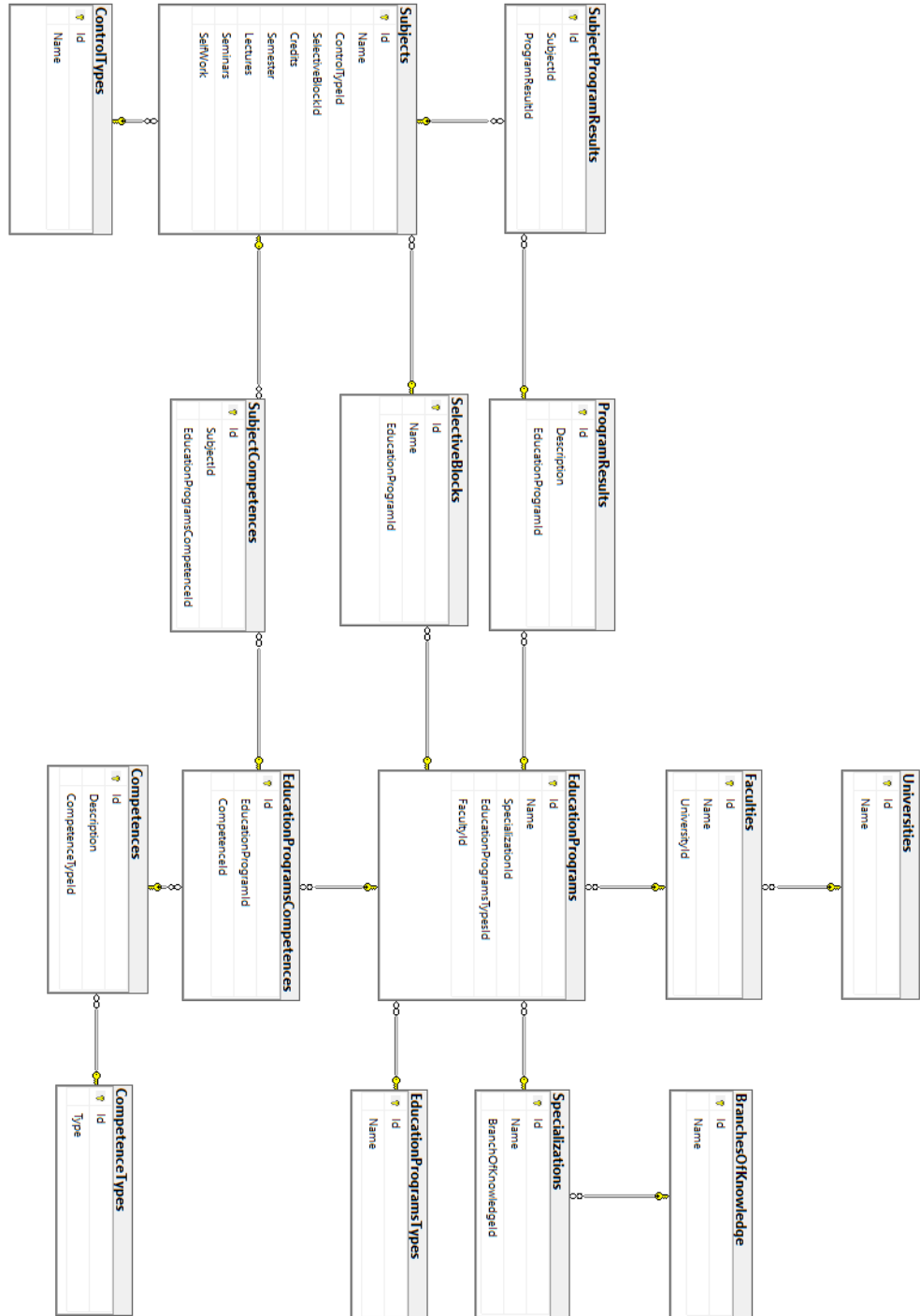
ДОДАТОК А

Структурно-логічна схема освітньо-професійної програми «Інформатика»



ДОДАТОК Б

Структурна схема бази даних освітніх програм



ДОДАТОК В

Код представлення Index моделі EducationProgram

```

@model IEnumerable<WorkProgramsWebApp.EducationProgram>

@{
    ViewData["Title"] = "Освітні програми";
}

<h2>Освітні програми:<br>@ViewBag.FacultyName</h2>

<p>
    @Html.ActionLink("Компетентності за освітніми програмами",
    "Index", "EducationProgramsCompetences") <br />
    @if (ViewContext.HttpContext.User.IsInRole("admin"))
    {
        @Html.ActionLink("Додати освітню програму", "Create", new
        { facultyId = @ViewBag.FacultyId })
    }
</p>

<style>
    table th, td
    {
        display: table-cell;
        text-align: center;
    }
    .left
    {
        text-align: left;
    }
</style>

<table class="table">
    <thead>
        <tr>
            <th style="vertical-align:middle">
                @Html.DisplayNameFor(model => model.Name)
            </th>

            <th style="vertical-align:middle">
                @Html.DisplayNameFor(model =>
                model.EducationProgramsTypesId)
            </th>

            <th style="vertical-align:middle">

```

```

        @Html.DisplayNameFor(model =>
model.SpecializationId)
        </th>

        <th></th>
    </tr>
</thead>
<tbody>
@foreach (var item in Model) {
    <tr>
        <td style="vertical-align:middle">
            @Html.DisplayFor(modelItem => item.Name)
        </td>

        <td style="vertical-align:middle">
            @Html.DisplayFor(modelItem =>
item.EducationProgramsTypes.Name)
        </td>

        <td style="vertical-align:middle">
            @Html.DisplayFor(modelItem =>
item.Specialization.Name)
        </td>

        <td class="left">
            <a asp-action="Details" asp-route-
id="@item.Id">·Вибіркові блоки</a><br />
            <a asp-action="Details2" asp-route-
id="@item.Id">·Програмні результати</a>

            @if (ViewContext.HttpContext.User.IsInRole("admin"))
            {
                <br /><a asp-action="Edit" asp-route-
id="@item.Id">·Редагувати</a><br />
                <a asp-action="Delete" asp-route-
id="@item.Id">·Видалити</a>
            }
        </td>
    </tr>
}
</tbody>
</table>

```

ДОДАТОК Г

Метод `ExportToWord` класу `SubjectsController`

```

public async Task<IActionResult> ExportToWord(int? id)
{
    if (id == null)
    {
        return NotFound();
    }
    var subject = await _context.Subjects.FindAsync(id);
    string templatePath =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "Word",
"template.docx");

    DocX doc = DocX.Load(templatePath);

    var selectiveBlock = _context.SelectiveBlocks.Where(s
=> s.Id == subject.SelectiveBlockId).FirstOrDefault();
    var educationProgram =
_context.EducationPrograms.Where(s => s.Id ==
selectiveBlock.EducationProgramId).FirstOrDefault();
    var faculty = _context.Faculties.Where(s => s.Id ==
educationProgram.FacultyId).FirstOrDefault();
    var university = _context.Universities.Where(s => s.Id
== faculty.UniversityId).FirstOrDefault();
    var specialization = _context.Specializations.Where(s
=> s.Id == educationProgram.SpecializationId).FirstOrDefault();
    var branchOfKnowledge =
_context.BranchesOfKnowledges.Where(s => s.Id ==
specialization.BranchOfKnowledgeId).FirstOrDefault();
    var educationProgramType =
_context.EducationProgramsTypes.Where(s => s.Id ==
educationProgram.EducationProgramsTypesId).FirstOrDefault();
    var controlType = _context.ControlTypes.Where(s =>
s.Id == subject.ControlTypeId).FirstOrDefault();
    var subjectProgramResults = from p in
_context.SubjectProgramResults.Where(s => s.SubjectId ==
subject.Id) select p;
    var subjectCompetences = from c in
_context.SubjectCompetences.Where(s => s.SubjectId == subject.Id)
select c;

    doc.ReplaceText("<UNIVERSITYNAME>",
university.Name.ToUpper());
    doc.ReplaceText("<FACULTYNAME>",
faculty.Name.ToUpper());

```

```

        doc.ReplaceText("<SUBJECTNAME>",
subject.Name.ToUpper());
        doc.ReplaceText("<BRANCHOFKNOWLEDGE>",
branchOfKnowledge.Name);
        doc.ReplaceText("<SPECIALIZATION>",
specialization.Name);
        doc.ReplaceText("<EDUCATIONPROGRAMTYPE>",
educationProgramType.Name);
        doc.ReplaceText("<EDUCATIONPROGRAM>",
educationProgram.Name);
        doc.ReplaceText("<SELECTIVEBLOCK>",
selectiveBlock.Name);
        doc.ReplaceText("<SEMESTER>",
subject.Semester.ToString());
        doc.ReplaceText("<CREDITS>",
subject.Credits.ToString());
        doc.ReplaceText("<CONTROLTYPE>", controlType.Name);
        doc.ReplaceText("<YEAR>",
DateTime.Now.ToString("yyyy"));
        doc.ReplaceText("<YEAR2>",
DateTime.Now.AddYears(1).ToString("yyyy"));
        doc.ReplaceText("<HOURS>", subject.Hours.ToString());
        doc.ReplaceText("<LECTURES>",
subject.Lectures.ToString());
        doc.ReplaceText("<SEMINARS>",
subject.Seminars.ToString());
        doc.ReplaceText("<SELFWORK>",
subject.SelfWork.ToString());

        Formatting format = new Formatting();
        format.Size = 12;
        var bulletedList =
doc.AddList(_context.Competences.Where(s => s.Id ==
_context.EducationProgramsCompetences.Where(s => s.Id ==
subjectCompetences.First().EducationProgramsCompetenceId).FirstOrD
efault().CompetenceId).FirstOrDefault().Description + ";", 0,
ListItemType.Bulleted, formatting: format);
        foreach(var c in subjectCompetences.Skip(1))
        {
            var competenceId =
_context.EducationProgramsCompetences.Where(s => s.Id ==
c.EducationProgramsCompetenceId).FirstOrDefault().CompetenceId;
            doc.AddListItem(bulletedList,
_context.Competences.Where(s => s.Id ==
competenceId).FirstOrDefault().Description+";", 0, formatting:
format);
        }

```

```

        foreach (var paragraph in doc.Paragraphs)
        {
            paragraph.FindAll("<COMPETENCES>").ForEach(index
=> paragraph.InsertListAfterSelf(bulletedList));
        }
        doc.ReplaceText("<COMPETENCES>", "Компетентності, на
досягнення яких спрямована дана дисципліна:");

        int i = 0;
        foreach (var p in subjectProgramResults)
        {
            i++;
            var programResult =
_context.ProgramResults.Where(s => s.Id ==
p.ProgramResultId).FirstOrDefault();
            doc.Tables[1].InsertRow();

doc.Tables[1].Rows[i].Cells[0].Paragraphs[0].Append(programResult.
Description, format);
        }
        doc.Tables[1].Design = TableDesign.TableGrid;

        if(controlType.Id == 2)
        {
            for (int j = 3; j >= 0; j--)
            {
                doc.Tables[2].RemoveRow(j);
            }
        }

        using (var stream = new MemoryStream())
        {
            doc.SaveAs(stream);
            stream.Flush();

            return new FileContentResult(stream.ToArray(),
                "application/vnd.openxmlformats-
officedocument.wordprocessingml.document")
            {
                FileNameDownloadName = subject.Name + " - робоча
програма.docx"
            };
        }
    }
}

```

ДОДАТОК Д

Згенерований вебзастосунком шаблон РПНД дисципліни «Інструментальні засоби та технології програмування» ОП «Інформатика» (бакалаврський рівень) факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра (циклова комісія) _____
(для коледжів)

«ЗАТВЕРДЖУЮ»

Заступник декана/директора
з навчальної роботи

«___» _____ 20__ року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ ІНСТРУМЕНТАЛЬНІ СЕРЕДОВИЩА ТА ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ для студентів

галузь знань **12 «Інформаційні технології»**
(шифр і назва)
спеціальність **122 «Комп'ютерні науки»**
(шифр і назва спеціальності)
освітній рівень **бакалавр**
(молодший бакалавр, бакалавр, магістр)
освітня програма **«Інформатика»**
(назва освітньої програми)
вид дисципліни **обов'язкова**

Форма навчання	денна
Навчальний рік	2021/2022
Семестр	4
Кількість кредитів ECTS	5
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі:

(Науково-педагогічні працівники, які забезпечують викладання даної дисципліни у відповідному навчальному році)

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» __ 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» __ 20__ р.
(підпис, ПІБ, дата)

КИЇВ – 2021

Розробник(и): _____ (вказати авторів: ПІБ, науковий ступінь, вчене звання, посада, кафедра)

ЗАТВЕРДЖЕНО

Зав. кафедри _____

_____ (_____)

(підпис)

(прізвище та ініціали)

Протокол № ____ від « ____ » _____ 20__ р.

Схвалено гарантом освітньої програми «Інформатика» _____

« ____ » _____ 20__ року

Схвалено науково - методичною комісією факультету

Протокол від « ____ » _____ 20__ року № ____

Голова науково-методичної комісії _____ (_____)

(підпис)

(прізвище та ініціали)

« ____ » _____ 20__ року

ВСТУП

1. Мета дисципліни –

2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):

1. Успішне опанування курсу _____
2. Знання теоретичних основ _____
Або:
 1. Знати _____
 2. Вміти _____
3. Володіти елементарними навичками _____

3. Анотація навчальної дисципліни

4. Завдання (навчальні цілі):

Компетентності, на досягнення яких спрямована дана дисципліна:

- ЗК2. Здатність застосовувати знання у практичних ситуаціях;
- ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт;
- СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктноорієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління;
- СК9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах;
- СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника;

5. Результати навчання за дисципліною: (описуються з детальною достовірністю для розробки заходів оцінювання)

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1.1	Знати....	лекція	Тест, 60% правильних відповідей	15%
1.2	Знати....	лекція	--/--	15%
2.1	Вміти	лабораторна робота	Звіт по лабораторній роботі	30%

4.1	Прийняти і обґрунтувати рішення.....	лекція, заняття, робота	семінарське самостійна	Кейс-задача	40%
-----	--------------------------------------	-------------------------------	---------------------------	-------------	-----

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання (необов'язково для вибіркових дисциплін які не входять до блоків спеціалізації)

Результати навчання дисципліни	1.1	1.2	2.1	4.1	4.2
Програмні результати навчання					
ПРН9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.					
ПРН11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).					
ПРН15. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.					

7. Схема формування оцінки.

7.1 Форми оцінювання студентів: (зазначається перелік видів робіт та форм їх контролю / оцінювання із зазначенням результатів навчання які на них мають бути оцінені, а також кількість балів/відсоток у підсумковій оцінці із дисципліни, пороговий рівень позитивної оцінки)

- семестрове оцінювання:

1. Контрольна робота (тест): РН 1.1.— 10 балів/6 балів.

2. Реферат: РН1.1, РН 2.1. - 20 балів/12 балів.

3.

- підсумкове оцінювання (у формі іспиту/комплексного іспиту, диференційованого заліку) вказується:

- форма оцінювання (у випадку комплексного іспиту слід вказати питому вагу складових);

- максимальна кількість балів які можуть бути отримані студентом (зазвичай 40 балів по 100-бальній шкалі);

- результати навчання які будуть оцінюватись;

- форма проведення і види завдань (а також їх частка в сукупній оцінці);

- мінімальний пороговий рівень екзаменаційної оцінки, за якої іспит вважається складеним, наприклад (для випадку коли на іспит виноситься 40 балів): “Для отримання загальної позитивної оцінки з дисципліни оцінка за іспит не може бути меншою 24 балів”

Слід також чітко прописати умови, які висуваються викладачами даної дисципліни як необхідна умова допуску до іспиту: “Студент не допускається до іспиту, якщо під час семестру набрав менше ніж ____ балів . Якщо серед результатів навчання дисципліни є такі які не можуть бути перевірені на іспиті формулюються додаткові вимоги, наприклад: “Студент допускається до іспиту за умови виконання всіх (або %) передбачених планом лабораторних робіт”

7.2 Організація оцінювання: (обов’язково зазначається порядок організації передбачених робочою навчальною програмою форм оцінювання із зазначенням орієнтовного графіку оцінювання).

7.3 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій і семінарських / практичних / лабораторних (вибрати необхідне) занять

№ п/п	Назва теми	Кількість годин		
		лекції	семінари/ практичні/ лабораторні вибрати необхідне	Самостійна робота
<i>Назва розділу чи частини 1 (якщо здійснюється поділ)</i>				
1	Вступ. Тема 1 Назва.	2		2
2	Тема 2. Назва	3		2
	...			
	Тема XX. Назва	-	-	7
	<i>Контрольна робота 1</i>		X	
<i>Частина 2 Назва (за наявності)</i>				
	Тема XXX. Назва			
	<i>Підсумкова модульна контрольна робота чи</i>	x	або x	
	<i>Курсова робота з дисципліни (за наявності у навчальному плані)</i>			
	ВСЬОГО	36	38	76

Загальний обсяг 150 год., в тому числі (вибрати необхідне):

Лекції – **36 год.**

Семінари – **38 год.**

Практичні заняття - ____ год.

Лабораторні заняття - ____ год.

Тренінги - ____ год.

Консультації - ____ год.

Самостійна робота - **76 год.**

9. Рекомендовані джерела:

Основна: (Базова)

до 10 фундаментальних, базових джерел

Додаткова:

як правило - до 20 джерел

10. Додаткові ресурси (за наявності):

Посилання на електронні ресурси (не тільки відкриті) на яких розміщено додаткову інформацію щодо дисципліни — приклади контрольних і екзаменаційних завдань, тематика рефератів, методичні вказівки по виконанню лабораторних робіт, тощо)