

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»

Директор інституту (декан факультету)

Андрій ФОРСЮК

(підпис)

(прізвище та ініціали)

«06» червня 2025 р.

«До захисту допущено»

Завідувач кафедри

Сергій ГРИБКОВ

(підпис)

(прізвище та ініціали)

«06» червня 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА зі спеціальності 122 комп'ютерні науки

(код і назва спеціальності)

освітньо-професійної програми Комп'ютерні науки

на тему: «Розроблення інформаційної системи формування контрактів для вступників НУХТ»

Виконав: здобувач 4 курсу, групи КН-4-2.

Паламар Роман Володимирович

(прізвище, ім'я та по батькові повністю)

(підпис)

Керівник Грибков Сергій Віталійович

(прізвище, ім'я та по батькові повністю)

(підпис)

Консультанти Сергій ГРИБКОВ

(ім'я та прізвище)

(підпис)

(ім'я та прізвище)

(підпис)

(ім'я та прізвище)

(підпис)

Рецензент

(ім'я та прізвище)

(підпис)

Я як здобувач Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав і не одержував недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач

(підпис)

Київ — 2025р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра інформаційних технологій, штучного інтелекту і кібербезпеки

Освітній ступінь бакалавр

Спеціальність 122 комп'ютерні науки

(код і назва)

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційних
технологій, штучного інтелекту і

кібербезпеки _____ **Сергій ГРИБКОВ**

«28» _____ квітня 2025 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Паламара Романа Володимировича

(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення інформаційної системи формування контрактів для вступників НУХТ»,

керівник роботи Грибков Сергій Віталійович, професор, доктор технічних наук,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «28» квітня 2025 р. № 254-кс

2. Строк подання здобувачем роботи: 30.05.2025 р.

3. Вихідні дані до роботи: 1. Нормативні документи НУХТ

2. Правила прийому до НУХТ

3. Інструкції щодо роботи приймальної комісії

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

1. Загальна характеристика університету

2. Організаційна структура приймальної комісії

3. Аналіз стану автоматизації структурного підрозділу

4. Аналіз і моделювання бізнес-процесів

5. Огляд існуючих програмних рішень

6. Постановка задачі розробки системи

7. Проектування бази даних

8. Реалізація програмного забезпечення

9. Тестування та інструкція користувача

5. Перелік графічного матеріалу:

Схеми організаційної структури НУХТ, функціональні моделі, моделі та схеми бази даних, інтерфейс інформаційної системи, фрагменти коду програми

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1,2,3	Грибков С.В , завідувач кафедри		

7. Дата видачі завдання: _____ 28 квітня 2025 року _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі. Визначення цілей, завдань та вимог до системи	02.05.2025	Виконано
2	Аналіз предметної області. Огляд аналогічних рішень	4.05.2025	Виконано
3	Проектування функціональної структури та логіки роботи системи	5.05.2025	Виконано
4	Реалізація основних модулів системи та проектування бази даних	15.05.2025	Виконано
5	Підготовка пояснювальної записки, презентації	19.05.2025	Виконано
6	Оформлення відгуку, рецензії, подання роботи	29.05.2025	Виконано

Здобувач _____

Роман ПАЛАМАР _____

Керівник роботи _____

Сергій ГРИБКОВ _____

АНОТАЦІЯ

Кваліфікаційна робота на тему «Розроблення інформаційної системи формування контрактів для вступників НУХТ» складається з 3 розділів обсягом 86 сторінок, містить 42 рисунка, 13 таблиць та 30 посилань на використані джерела.

Робота присвячена дослідженню предметної області процесу формування контрактів для вступників НУХТ та розробці веб-застосунку для автоматизації цього процесу. У роботі проаналізовано організаційну структуру університету, досліджено нинішній стан комп'ютеризації та виявлено недоліки існуючого процесу формування контрактів. Проведено функціональне моделювання бізнес-процесів, розглянуто існуючі рішення для автоматизації та обґрунтовано доцільність розробки власної системи. Розроблена інформаційна система дозволяє скоротити час формування контрактів, зменшити ймовірність помилок та підвищити загальну ефективність роботи приймальної комісії.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, ПРИЙМАЛЬНА КОМІСІЯ, ФОРМУВАННЯ КОНТРАКТІВ, АВТОМАТИЗАЦІЯ БІЗНЕС-ПРОЦЕСІВ, НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ, ВСТУПНА КАМΠΑНІЯ, ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ, ТЕХНІКО-ЕКОНОМІЧНЕ ОБґРУНТУВАННЯ, ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ, БАЗА ДАНИХ

SUMMARY

The qualification work on the topic «Development of an information system for the formation of contracts for NUFT entrants» consists of 3 chapters with a volume of 86 pages, contains 42 images, 13 tables and 30 references to the sources used.

The work is devoted to the study of the subject area of the process of forming contracts for NUFT entrants and the development of a web application to automate this process. The paper analyzes the organizational structure of the university, examines the current state of computerization, and identifies the shortcomings of the existing contract formation process. The functional modeling of business processes was carried out, existing automation solutions were considered, and the feasibility of developing our own system was substantiated. The developed information system allows to reduce the time of contract formation, reduce the likelihood of errors, and increase the overall efficiency of the admission committee.

Keywords: INFORMATION SYSTEM, ADMISSION COMMITTEE, CONTRACT FORMATION, BUSINESS PROCESS AUTOMATION, NATIONAL UNIVERSITY OF FOOD TECHNOLOGIES, ADMISSION CAMPAIGN, ECONOMIC EFFICIENCY, FEASIBILITY STUDY, FUNCTIONAL MODELING, DATABASE

ЗМІСТ

АНОТАЦІЯ	4
SUMMARY	5
ЗМІСТ	6
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	8
ВСТУП	9
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	11
1.1. Загальна характеристика Національного університету харчових технологій	11
1.2. Організаційна структура Національного університету харчових технологій	12
1.3. Аналіз нинішнього стану комп'ютеризації університету.....	14
1.4. Функціональне моделювання та аналіз існуючих бізнес-процесів	16
1.5. Огляд існуючих рішень для розв'язання виявлених проблем	21
1.6. Розрахунок техніко-економічного обґрунтування впровадження створюваного програмного забезпечення	24
1.7. Обґрунтування доцільності проектування й розроблення інформаційної системи формування контрактів для вступників НУХТ	26
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ.....	28
2.1. Загальні положення.....	28
2.2. Призначення і цілі створення системи.....	28
2.3. Характеристика об'єкта автоматизації.....	28
2.4. Вимоги до системи.....	29

2.5. Склад і зміст робіт по створенню системи	38
2.6. Порядок контролю і приймання системи	39
2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію .	40
2.8. Вимоги до документації	40
РОЗДІЛ 3. СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	42
3.1. Опис та обґрунтування вибору програмно-технічних засобів розроблення програмного продукту	42
3.2. Проектування та створення бази даних	45
3.3 Реалізація функцій системи	47
3.4. Інструкція користувача.....	58
3.5 Тестування програмного продукту	78
ВИСНОВКИ.....	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	84
ДОДАТКИ.....	87
Додаток А. Схеми організаційної структури НУХТ	87
Додаток Б. Функціональні моделі	88
Додаток В. Моделі та схеми Бази Даних	92
Додаток Г. Інтерфейс Інформаційної Системи.....	93
Додаток Д. Фрагменти коду програми.....	106

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

- НУХТ - Національний університет харчових технологій
- БД - база даних
- СУБД - система управління базами даних
- ПЗ - програмне забезпечення
- МЗ - математичне забезпечення
- ІЗ - інформаційне забезпечення
- ЛЗ - лінгвістичне забезпечення
- API - Application Programming Interface (програмний інтерфейс)
- UI/UX - User Interface/User Experience (інтерфейс користувача/досвід користувача)
- ДСТУ - Державний стандарт України
- ДБН - Державні будівельні норми
- ЄДЕБО - ЄДИНА ДЕРЖАВНА ЕЛЕКТРОННА БАЗА З ПИТАНЬ ОСВІТИ

ВСТУП

В умовах сьогодення, з стрімким розвитком інформаційних технологій та глобальною цифровізацією, одним із пріоритетних напрямків модернізації навчальних закладів є цифрова трансформація різних процесів. Цифровізація освітніх установ дозволяє суттєво підвищити ефективність та швидкість їх роботи а також знизити вплив людського фактору на різноманітні діяльності.

Особливо актуальним питанням є автоматизація документообігу під час вступної кампанії, коли навчальні заклади стикаються з необхідністю обробки великих обсягів даних та формування значної кількості однотипних документів у стислі терміни. Особливо це стосується документів, що є внутрішніми, адже усі дії про вступну компанію вносяться до ЄДЕБО. Національний університет харчових технологій, як один із провідних закладів вищої освіти України, щороку приймає тисячі нових студентів, і процес оформлення документів для вступників потребує значних людських та часових ресурсів.

На сьогоднішній день процес формування контрактів для вступників НУХТ здійснюється переважно вручну, що супроводжується низкою проблем: високою вірогідністю помилок при заповненні документів, значними часовими витратами на підготовку кожного контракту, складністю внесення змін у вже створені документи та обмеженою можливістю контролю за процесом з боку керівництва. Це призводить до зниження ефективності роботи приймальної комісії, збільшення черг вступників та створення додаткового навантаження на працівників університету.

Впровадження спеціалізованої інформаційної системи для автоматизації процесу формування контрактів є одним із ключових кроків у цифровій трансформації НУХТ. Така система дозволить не лише прискорити та спростити роботу приймальної комісії, але й покращити досвід взаємодії вступників з університетом, що є важливим фактором конкурентоспроможності навчального закладу на ринку освітніх послуг.

Актуальність розробки інформаційної системи формування контрактів для вступників НУХТ обумовлена необхідністю оптимізації та прискорення процесу підготовки документів під час вступної кампанії, необхідністю створення централізованої бази даних вступників для зручного доступу, аналізу та звітності;

Використання сучасних веб-технологій та фреймворків дозволяє створити зручний у використанні застосунок, що забезпечить не лише ефективне вирішення поточних проблем, а й створить основу для подальшого розвитку інформаційних систем НУХТ.

Метою даної кваліфікаційної роботи є розробка інформаційної системи, яка дозволить автоматизувати процес формування контрактів для вступників НУХТ, забезпечить зручне зберігання та управління даними, а також спростить взаємодію між членами приймальної комісії під час вступної кампанії.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальна характеристика Національного університету харчових технологій

Національний університет харчових технологій - один з закладів вищої освіти України, що спеціалізується на підготовці фахівців для харчової, фармацевтичної та агропромислових галузей, а також різних інших спеціальностей. Університет має довгу, цікаву історію, що бере свій початок з 1884 року, коли було засновано Смілянське технічне училище цукрової промисловості. У 1930 році на базі факультету цукрової промисловості Київського політехнічного інституту було створено Київський інститут цукрової промисловості, який згодом трансформувався у Київський технологічний інститут харчової промисловості. У 1993 році заклад став університетом, а в 2002 році він отримав статус національного університету.

Головною метою НУХТ є підготовка висококваліфікованих фахівців рівня бакалавр, магістр та докторів філософії технічних спеціальностей пов'язаних з харчовою та переробною виробництвами, проведення наукових досліджень, впровадження інноваційних технологій у виробництва, а також забезпечення якісної освіти відповідно до міжнародних та всеукраїнських стандартів.

Сфера діяльності університету охоплює:

- освітню діяльність;
- науково-дослідну роботу;
- міжнародну співпрацю;
- інноваційну діяльність;
- культурно-виховну роботу.

Особливістю НУХТ є його тісний зв'язок з різними виробництвами, що дозволяє здобувачам освіти отримувати не лише теоретичні знання в університеті, але й практичні навички на виробничих практиках, на виробництвах-партнерів.

Університет має розвинену інфраструктуру, що включає декілька навчальних корпусів, лабораторій, бібліотеку, кілька гуртожитків, спортивний зал.

1.2. Організаційна структура Національного університету харчових технологій

1.2.1. Загальна схема організаційної структури університету

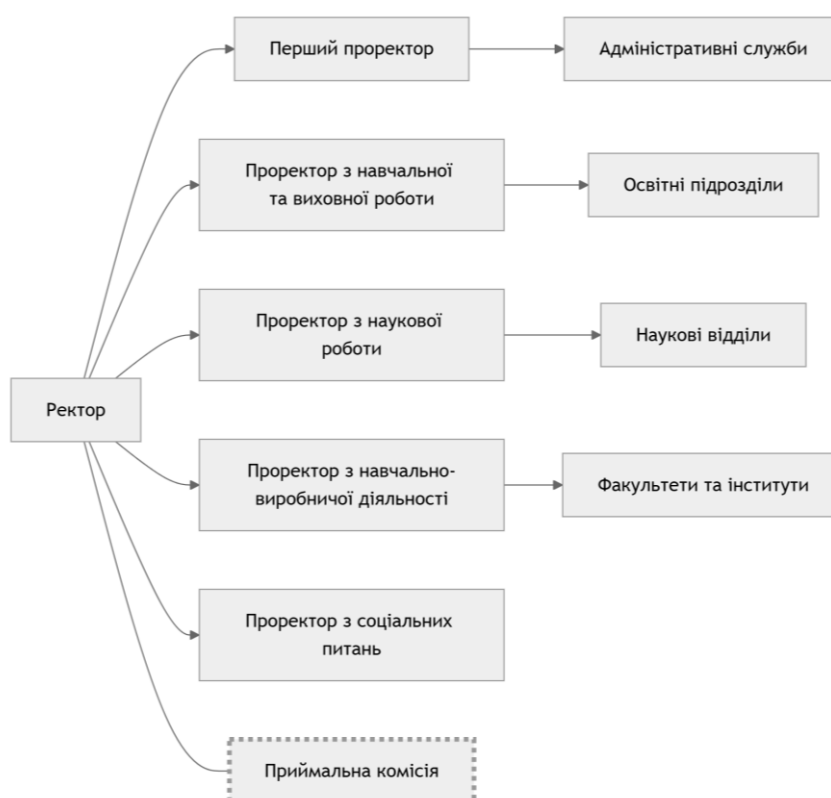


Рисунок 1.1 – Схема організаційної структури університету

Управління Національним університетом харчових технологій здійснюється на засадах автономії та академічного самоврядування. Найвищим колегіальним органом університету є Вчена рада, яка ухвалює рішення з ключових питань освітнього процесу. Безпосереднє керівництво діяльністю університету здійснює ректор, який обирається шляхом таємного голосування строком на п'ять років. Ректор очолює ректорат, до складу якого входять низка проректорів: з науково-педагогічної та виховної роботи, з наукової роботи, з навчально-виробничої діяльності та розвитку, з соціально-побутових питань тощо.

Центральний апарат університету включає також відповідні навчальні підрозділи (інститути та факультети зі своїми директорами і деканами) і решта

адміністративних служб. Наприклад до них належить Навчально-методичне управління (з підпорядкованими навчально-методичним та навчальним відділами), відділ міжнародних зв'язків, відділ докторантури і аспірантури, відділ наукової роботи студентів, а також загально-адміністративні відділи – юридичний, відділ кадрів, відділ документального забезпечення, архів, бухгалтерія, планово-фінансовий відділ, відділ матеріально-технічного постачання тощо (Рис 3.1).

При розробці системи буде частково автоматизована діяльність підрозділу університету - приймальної комісії.

1.2.2. Структура приймальної комісії

Приймальна комісія - дуже важливий підрозділ університету, який відповідає за організацію процесу вступу, прийом документів, оформлення контрактів та супровід вступників до моменту їх зарахування. Її схема зображена на рисунку 1.2.



Рисунок 1.2 – Схема структури приймальної комісії

Основні функції приймальної комісії:

- прийом та перевірка вступних документів;
- формування та видача контрактів на навчання;
- консультування вступників щодо умов вступу та оформлення документів;

- ведення обліку вступників та контрактів.

Посадові обов'язки членів приймальної комісії:

- **Ректор** – керує загальним процесом;
- **Перший проректор** – здійснює контроль за діяльністю комісії;
- **Голова приймальної комісії** – відповідає за організацію та проведення вступної кампанії;
- **Заступник голови приймальної комісії** – координує діяльність комісії;
- **Відповідальний секретар факультету** – забезпечує адміністрування процесів на рівні факультетів;
- **Члени приймальної комісії** – безпосередньо працюють із документами вступників.

Взаємодія з іншими структурними підрозділами показана на таблиці 1.1

Таблиця 1.1 Взаємодія системи з іншими структурними підрозділами

Підрозділ	Отримує	Надає
Приймальна комісія	Документи абітурієнтів	Списки допущених , рекомендованих , зарахованих.
Юридичний відділ	Проекти контрактів	Погоджені контракти
Бухгалтерія	Підписані контракти	Інформацію про оплату
Навчальний відділ	Інформацію про зарахованих студентів	Накази про зарахування
Деканати	Особові справи зарахованих студентів	Інформацію про групи

1.3. Аналіз нинішнього стану комп'ютеризації університету

На даний момент в НУХТ працюють багато інформаційних систем, що забезпечують автоматизацію деяких аспектів діяльності університету:

- система управління навчальним процесом (пакет програм “Деканат”):
 - модуль електронного журналу;

- модуль електронного розкладу.
- електронна бібліотека;
- система дистанційного навчання.

Однак, для приймальної комісії немає окремої інформаційної системи для формування та ведення договорів. Наразі процес формування контрактів здійснюється з використанням стандартних офісних програм (Microsoft Word, Excel) та включає багато ручних операцій:

1. Оператор вручну вводить дані вступника у Excel файл.
2. Здійснюється генерація документів за допомогою вбудованої функції програмного забезпечення Excel - VBA (Visual Basic for Applications)
3. Документ зберігається, і якщо потрібно - відправляється іншим учасникам приймальної комісії.

Такий підхід має ряд суттєвих недоліків:

- високий шанс виникнення помилок при ручному введенні даних;
- значні витрати часу на формування кожного контракту через повільну роботу скриптів VBA;
- складність внесення змін до шаблонів документів, адже потрібно частково переписувати наявні VBA скрипти;
- відсутність централізованого зберігання даних;
- складність редагування сформованих документів;
- відсутні можливості пошуку та фільтрації інформації, адже вона не зберігається;
- відсутність контролю за правильністю заповнення документів.

Нинішній стан цифровізації процесу формування контрактів є незадовільним, оскільки не забезпечує ефективного та зручного виконання задач приймальної комісії, особливо в період активної вступної кампанії. Ручне заповнення контрактів та договорів на навчання, а також їх повільна генерація призводить до затримок, помилок та сповільнення роботи підрозділу.

1.4. Функціональне моделювання та аналіз існуючих бізнес-процесів

1.4.1. Функціональна модель процесу формування контрактів для вступників

Основний бізнес-процес формування контрактів для вступників (Рис. 1.3) включає наступні етапи.

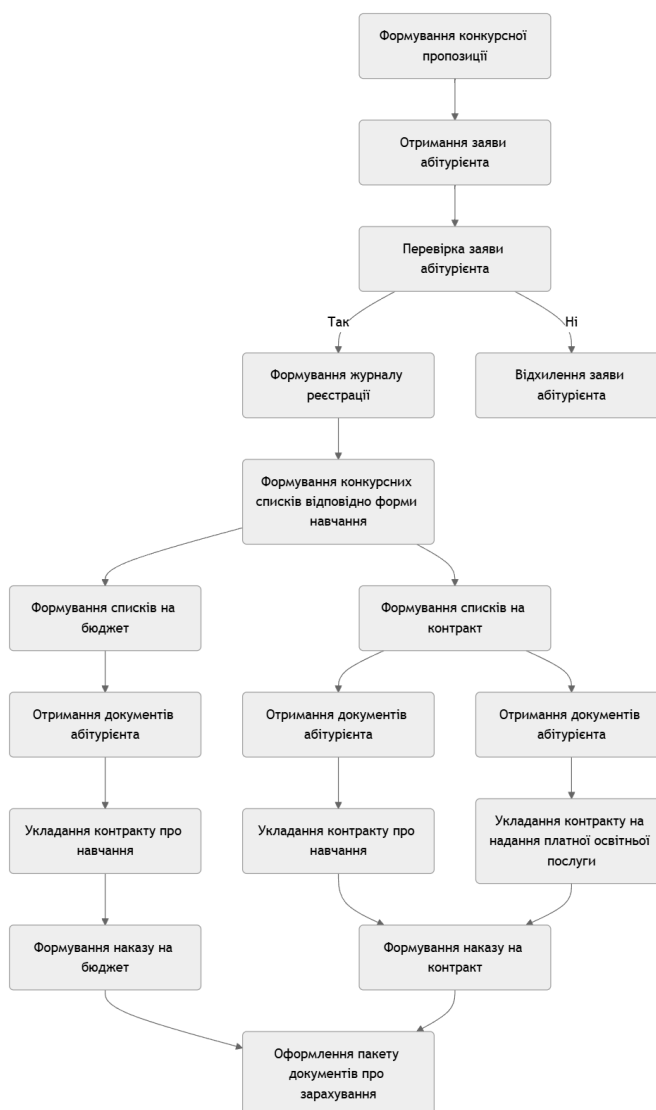


Рисунок 1.3 - Бізнес-процес по оформленню пакету документів про зарахування

1. Формування конкурсної пропозиції

- а. Вхідні дані: навчальні програми, спеціальності, форми навчання
- б. Виконавець: відбіркова комісія
- с. Результат: опубліковані конкурсні пропозиції

2. Отримання заяви абітурієнта

- a. Вхідні дані: заява, документи вступника
- b. Виконавець: оператор приймальної комісії
- c. Результат: прийнята до розгляду заява

3. Перевірка заяви абітурієнта

- a. Вхідні дані: заява, особисті документи
- b. Виконавець: оператор приймальної комісії
- c. Результат: підтвердження або відхилення заяви

4. Формування журналу реєстрації

- a. Вхідні дані: прийнята заява
- b. Виконавець: оператор приймальної комісії
- c. Результат: зареєстрована заява у журналі

5. Формування конкурсних списків

- a. Вхідні дані: реєстр заяв, форма навчання
- b. Виконавець: приймальна комісія
- c. Результат: списки на бюджет та контракт

6. Отримання документів абітурієнта

- a. Вхідні дані: особисті документи, оригінали
- b. Виконавець: оператор приймальної комісії
- c. Результат: повний пакет документів вступника

7. Укладання контракту про навчання / надання платної освітньої послуги

- a. Вхідні дані: документи вступника, форма навчання
- b. Виконавець: юрист або відповідальний оператор
- c. Результат: укладений контракт

8. Формування наказу

- a. Вхідні дані: підписані контракти
- b. Виконавець: навчальний відділ
- c. Результат: наказ на зарахування (бюджет/контракт)

9. Оформлення пакету документів про зарахування

- a. Вхідні дані: наказ про зарахування, документи вступника

- b. Виконавець: оператор навчального відділу
- c. Результат: завершене зарахування вступника

1.4.2. Виявлені проблеми

На основі аналізу існуючих бізнес-процесів виявлено наступні проблеми:

1. Низька ефективність процесу

- Значні витрати часу на заповнення та формування документів
- Дублювання введення даних на різних етапах
- Відсутність автоматизованої перевірки правильності заповнення

2. Висока ймовірність помилок

- Ручне введення даних збільшує ризик помилок
- Відсутність перевірки наявності даних при введенні
- Складність внесення змін у вже створені документи

3. Неефективне зберігання та обробка даних

- Відсутність централізованої бази даних
- Складність пошуку інформації про сформовані документи

1.4.3. Пропозиції щодо усунення наявних проблем

На основі аналізу існуючих бізнес-процесів та виявлених проблем пропонується розробка спеціалізованої інформаційної системи формування контрактів для вступників НУХТ [29]. На рисунку 1.4 представлена функціональна модель бізнес-процесів ТО ВЕ.

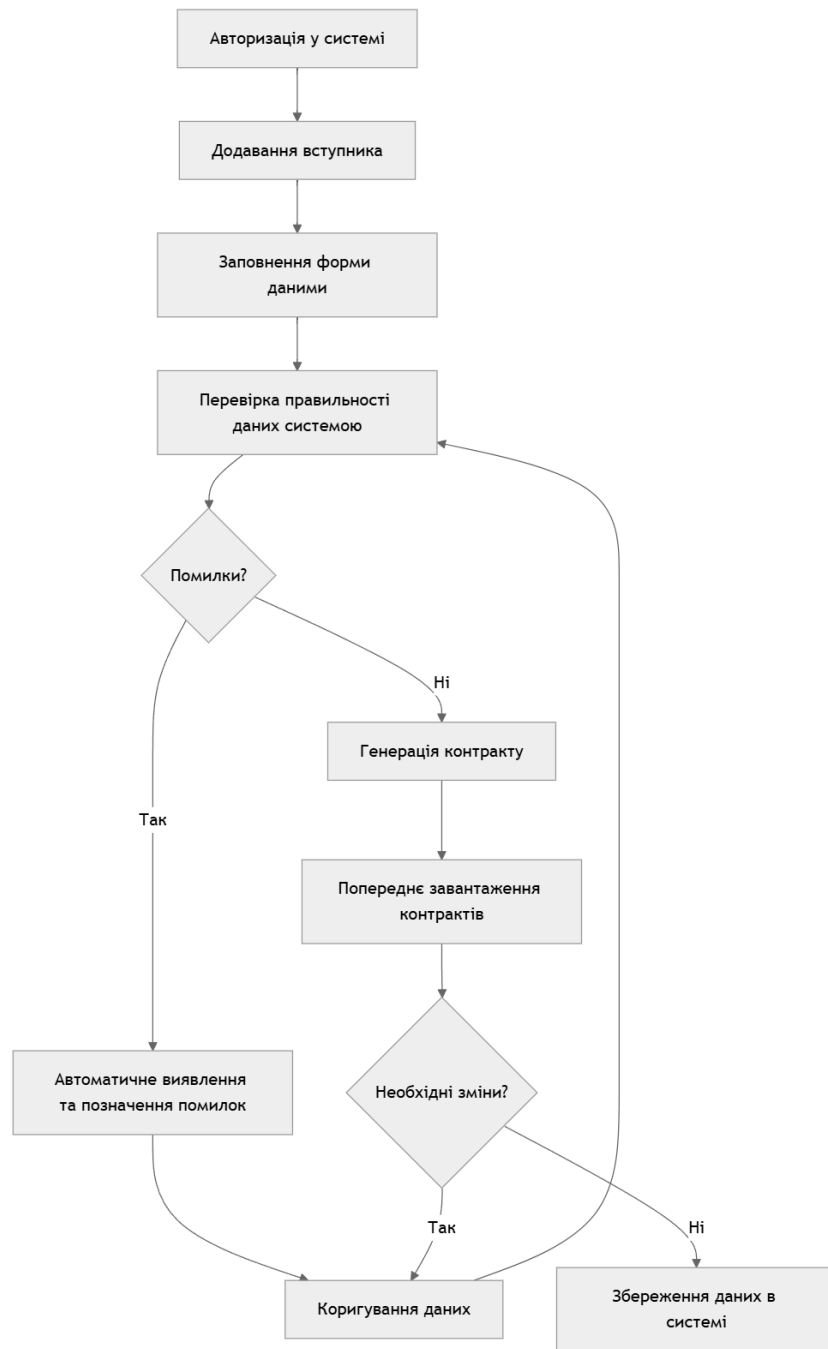


Рисунок 1.4 – Функціональна модель TO BE

Порівняння цих моделей AS IS та TO BE можна побачити у таблиці 1.2.

Таблиця 1.2 Порівняння моделей AS IS та TO BE

Параметр	AS IS	TO BE
Введення даних	Ручне введення з паперових документів	Ручне введення в форму застосунку
Перевірка даних	Візуальна перевірка оператором	Автоматична валідація системою (лише на обов'язкові поля)
Формування контракту	Заповнення шаблону у Excel	Автоматичне формування з перевіреними даними
Виправлення помилок	Повторне заповнення шаблону, або повторне заповнення полів в Excel файлі	Редагування проблемних полів
Зберігання даних	Локальні файли	Централізована база даних
Пошук інформації	Перегляд паперових документів або локальних файлів	Повнотекстовий пошук у базі даних
Формування звітності	Відсутнє повністю	Генерація загального звіту усіх збережених даних
Доступ до інформації	Локальний доступ	Веб-доступ з авторизацією

Ключові переваги запропонованої моделі:

1. Підвищення ефективності процесу

- скорочення часу на формування одного контракту з 15-20 хвилин до 5-10 хвилин;

2. Мінімізація помилок

- автоматична валідація даних при введенні;
- легкість внесення змін у контракти;

3. Ефективне зберігання та обробка даних

- Швидкий пошук та фільтрація інформації

Перспективи використання розробки:

1. Розширення функціональності системи

- Автоматичне формування наказів про зарахування
- Інтеграція з системою електронного документообігу університету

2. Масштабування системи

- Адаптація системи для використання іншими структурними підрозділами університету
- Використання системи у філіях та коледжах НУХТ

3. Додаткові можливості

- Реалізація функції електронного підпису контрактів

1.5. Огляд існуючих рішень для розв'язання виявлених проблем

Для розв'язання виявлених проблем було проведено дослідження вітчизняного ринку інформаційних систем, які могли б задовольнити потреби приймальної комісії НУХТ. Розглянемо основні системи-аналоги.

1.5.1. Система «АБІТУРІЄНТ» від компанії «Політек-СОФТ»

Система «АБІТУРІЄНТ» призначена для автоматизації роботи приймальної комісії закладу освіти. Даний програмний комплекс дозволяє організувати автоматизоване робоче місце усіх співробітників приймальної комісії.

Основні функціональні можливості системи:

- створення та редагування анкет абітурієнтів;
- формування необхідних документів (заяви, екзаменаційні листи, контракти);
- формування статистичних звітів;
- імпорт даних з ЄДЕБО;
- архівування даних.

Переваги:

- наявність інтеграції з ЄДЕБО;
- широкий спектр готових форм звітності та документів;
- наявність української локалізації.

Недоліки:

- висока вартість ліцензії та щорічної підтримки;
- обмежені можливості для модифікації шаблонів документів.

1.5.2. Автоматизована система «Приймальна комісія» від розробника «НДІ ПІТ»

Система «Приймальна комісія» є модулем комплексної системи управління навчальним закладом АСУ «ВНЗ». Вона призначена для автоматизації всіх етапів роботи приймальної комісії.

Основні функціональні можливості системи:

- ведення реєстру абітурієнтів;
- формування рейтингових списків;
- генерація та друк документів (заяви, договори);
- зберігання даних про результати вступних випробувань;
- формування наказів про зарахування.

Переваги:

- інтеграція з іншими модулями системи АСУ «ВНЗ»;
- регулярні оновлення відповідно до змін у законодавстві.

Недоліки:

- висока вартість;
- складний інтерфейс, що потребує тривалого навчання персоналу;
- необхідність придбання всієї системи АСУ «ВНЗ»;
- обмежені можливості інтеграції з зовнішніми системами.

1.5.3. Система «1С:Університет» від компанії «1С»

Система «1С:Університет» є спеціалізованим рішенням на платформі «1С» для автоматизації діяльності закладів вищої освіти. Наразі система «1С» заборонена для використання на території України.

Основні функціональні можливості системи:

- реєстрація абітурієнтів та їх заяв;
- автоматичне формування рейтингових списків;
- формування та друк документів (заяви, договори);
- автоматизація наказів;
- формування статистичних звітів.

Переваги:

- широкі можливості налаштування;
- інтеграція з іншими продуктами на платформі «1С».

Недоліки:

- висока сукупна вартість впровадження;
- складність розгортання та налаштування;
- необхідність наявності спеціальних навичок для адміністрування;
- неможливість легального використання у зв'язку з санкціями щодо російського ПЗ.

Порівняння цих систем наведено у таблиці 1.3.

Таблиця 1.3 Порівняння систем-аналогів

Критерій	АБІТУРІЄН Т	Приймальн а комісія	1С:Університет	Власна розробка
Вартість придбання	25 000 грн	35 000 грн	40 000 грн	0 грн
Вартість щорічної підтримки	5 000 грн	7 000 грн	8 000 грн	Виконується власними силами
Інтеграція з ЄДЕБО	+	+/-	-	-
Веб-інтерфейс	-	-	-	+
Можливість доопрацювання	+/-	-	+/-	+
Швидкість розгортання	Середня	Середня	Низька	Висока
Необхідність навчання персоналу	Висока	Висока	Висока	Низька
Відповідність специфіці університету	Часткова	Часткова	Часткова	Повна
Інтерфейс українською мовою	+	+	+	+
Режим автономної роботи	+	-	+	+

1.6. Розрахунок техніко-економічного обґрунтування впровадження створюваного програмного забезпечення

Для оцінки економічної доцільності впровадження власної інформаційної системи формування контрактів для вступників НУХТ проведемо розрахунок основних економічних показників.

1.6.1. Витрати на розробку та впровадження

Прямі витрати на розробку зображені на Таблиці 1.4 , та становлять 167 500 грн. Ця сума складається з заробітної плати працівників , залучених до розробки системи.

Таблиця 1.4 Прямі витрати на розробку

Витрати	Кількість	Вартість за од., грн	Сума, грн
Заробітна плата розробника	3 місяці	30 000	90 000
Заробітна плата проектного менеджера	1,5 місяці	35 000	52 500
Заробітна плата тестувальника	1 місяць	25 000	25 000
Всього			167 500

Непрямі витрати зображені на Таблиці 1.5 та становлять 18 000 грн. В цю суму входять комунальні послуги , амортизація обладнання та решта витрат.

Таблиця 1.5 Прямі витрати на розробку

Витрати	Сума, грн
Комунальні послуги	5 000
Амортизація обладнання	10 000
Інші витрати	3 000
Всього	18 000

До витрат на впровадження входить навчання персоналу та налаштування системи , що становить в сумі 0 гривень , адже для навчання персоналу , в системі передбачена детальна документація по користуванню застосунком.

Загальні витрати на розробку та впровадження: $167\,500 + 18\,000 = 185\,500$ грн

1.6.2. Економічний ефект від впровадження

Економія коштів за рахунок автоматизації процесів описана в Таблиці 1.6.

Таблиця 1.6 Економія коштів за рахунок автоматизації

Процес	Економія часу на 1 операцію, хв	Кількість операцій за рік	Загальна економія часу, годин	Вартість години, грн	Сума економії, грн
Формування контракту	15	2 000	500	150	75 000
Пошук інформації	20	500	167	150	25 050
Формування звітів	120	50	100	200	20 000
Всього					120 050

Загальний економічний ефект за перший рік: 120 050 грн

1.6.3. Розрахунок терміну окупності

Термін окупності = Витрати на розробку та впровадження / Річний економічний ефект

Термін окупності = 209 500 / 120 050 = 1,7 року

Таким чином, витрати на розробку та впровадження інформаційної системи формування контрактів для вступників НУХТ окупляться за **20,4 місяці** (1,7 року).

Варто зазначити, що розробка даної інформаційної системи, хоча і не забезпечує прямої прибутковості, має значний соціальний ефект для працівників приймальної комісії НУХТ. Система спрямована на оптимізацію та прискорення робочих процесів, що дозволить зменшити навантаження на членів приймальної комісії під час пікових періодів вступної кампанії, та покращити процес формування договорів.

Отже, незважаючи на відсутність прямої прибутковості, впровадження системи є обґрунтованим з точки зору підвищення ефективності роботи приймальної комісії та загального покращення якості надання освітніх послуг НУХТ.

1.7. Обґрунтування доцільності проєктування й розроблення інформаційної системи формування контрактів для вступників НУХТ

На основі проведеного аналізу предметної області, бізнес-процесів та техніко-економічного обґрунтування можна зробити наступні висновки:

1. **Недоліки існуючого процесу формування контрактів:** Поточний процес формування контрактів для вступників НУХТ має декілька суттєвих недоліків, такі як висока ймовірність помилок при ручному введенні даних, значні витрати часу, відсутність централізованого зберігання даних, складність редагування згенерованих документів, відсутність автоматизованого контролю за заповненням документів.

2. **Аналіз існуючих рішень:** Дослідження ринку показало наявність кількох систем-аналогів, однак жодна з них не відповідає повністю потребам приймальної комісії НУХТ. Основними недоліками існуючих рішень є: висока вартість придбання та підтримки, складність налаштування та адаптації до специфіки університету, обмежені можливості інтеграції з іншими системами.

3. **Економічна доцільність:** Розрахунок техніко-економічного обґрунтування показав, що розробка власної інформаційної системи формування контрактів для вступників НУХТ є економічно доцільною. Термін окупності проєкту складає 20,4 місяці, що є прийнятним для такого типу проєктів.

4. **Перспективи розвитку:** Власна розробка дозволить у майбутньому розширювати функціональність системи відповідно до потреб університету, інтегрувати її з іншими інформаційними системами, що використовуються в НУХТ, а також адаптувати до можливих змін у процесах роботи приймальної комісії.

Таким чином, розробка інформаційної системи формування контрактів для вступників НУХТ на базі сучасних веб-технологій є доцільною та обґрунтованою з точки зору підвищення ефективності процесів, економічної вигоди та перспектив подальшого розвитку.

Впровадження даної системи дозволить:

- більше автоматизувати процес формування контрактів для вступників;
- зменшити ймовірність помилок при формуванні документів;
- скоротити час на створення одного набору контрактів з 15-20 хвилин до 5-10 хвилин;
- забезпечити централізоване зберігання та швидкий доступ до інформації усіма членами приймальної комісії;
- підвищити ефективність роботи приймальної комісії в цілому;

Згідно з проведеним дослідженням, розробка власної інформаційної системи є доцільним рішенням для усунення виявлених проблем та підвищення ефективності процесу формування контрактів для вступників.

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

2.1. Загальні положення

2.1.1 Найменування системи: «Інформаційна система формування контрактів для вступників НУХТ»

2.1.2 Результати робіт зі створення системи оформлюються згідно з вимогами ДСТУ [1] на відповідні етапи розробки. Порядок оформлення і передачі результатів [2] у даному випадку визначається змістом і календарним планом виконання розробки.

2.1.3 У випадку необхідності на наступних стадіях робіт по створенню системи окремі положення можуть уточнюватися і розвиватися.

2.2. Призначення і цілі створення системи

2.2.1. Призначення системи.

Система призначена для автоматизації процесу формування договорів для вступників Національного університету харчових технологій. Система автоматизує генерацію та управління різними типами договорів для абітурієнтів (договір про навчання, договір з фізичною особою, договір з юридичною особою)

2.2.2. Цілі створення системи.

Основною метою створення системи є забезпечення зручного та ефективного процесу формування договорів для вступників, що включає:

- спрощення процесу створення та зберігання договорів;
- стандартизацію формату договорів;
- автоматизацію процесу формування договорів на основі шаблонів;
- зберігання та пошук даних вступників та їхніх договорів.

2.3. Характеристика об'єкта автоматизації

2.3.1. Короткі відомості про об'єкт автоматизації.

Об'єктом автоматизації є діяльність приймальної комісії НУХТ з формування контрактів для вступників. Базовий об'єкт впровадження — НУХТ.

2.4. Вимоги до системи

2.4.1. Вимоги до структури і функціонування системи

Система повинна мати клієнт-серверну архітектуру на основі веб-технологій (Next.js), що використовує єдину базу даних (MySQL) та розгортається за допомогою Docker.

Система забезпечує доступ для таких категорій користувачів:

- адміністратор системи;
- користувачі системи (співробітники університету)

Взаємозв'язок між компонентами системи має здійснюватися на інформаційному рівні через загальну базу даних із використанням RESTful API для обміну даними.

Розвиток і модернізація системи повинні проводитися шляхом уточнення, нарощування чи заміни виконуваних функцій, модернізації технічних і програмних засобів. Структура і технологія програмного забезпечення системи повинні забезпечити простоту їх модернізації та розвитку, з можливістю масштабування, збільшення розмірності задач і масивів інформації.

Програмно-технічні засоби функціонування системи повинні забезпечувати засоби табличного, текстового, графічного відображення даних. Програмна та інформаційна сумісність має забезпечуватися використанням сучасних веб-технологій та API для міжсистемної взаємодії.

Функціонування системи має забезпечувати веб-орієнтовану та розподілену обробку даних з використанням контейнеризації (Docker).

2.4.2. Вимоги до чисельності і кваліфікації персоналу

Персонал, який використовує автоматизовану систему, повинен дотримуватися наступних вимог:

- пройти навчання і отримати навички роботи з системою;
- дотримуватись технологічних інструкцій при роботі з системою;
- дотримуватись умов експлуатації системи у відповідності з інструкціями;

- дотримуватись правил зберігання інформації і організації резервних копій бази даних;
- дотримуватись правил інформаційної безпеки при роботі з системою.

Користувачами системи можуть усі члени приймальної комісії. Авторизація в системі повинна здійснюватися через логін та пароль, які відображають рівень користувача і відповідні права доступу. Залежно від рівня, користувач отримує повний чи обмежений доступ до системи та її функцій.

2.4.3. Показники призначення

Відповідно до п. 2.1, показники призначення повинні характеризувати ступінь та якість автоматизації процесу формування контрактів вступників для оптимального функціонування приймальної комісії. Перелік і допустимі значення показників, при яких зберігається цільове призначення системи, повинні бути визначені на стадії проектування.

Система повинна мати можливість налаштування на параметри об'єкта управління при її модернізації та розвитку, а також зміні процесів та методів організаційного управління.

2.4.4. Вимоги до надійності

Система є багатофункціональною і призначена для використання протягом робочого дня. Всі функції системи виконуються дискретно. У відповідності з ДСТУ 2226-93 оцінка надійності проводиться по кожній функції окремо. Враховуючи особливості функціонування системи, показники її надійності є показниками надійності СУБД, на якій вона реалізована, та технічних засобів, на яких вона експлуатується. Основними показниками надійності є:

- L_i — ймовірність безвідмовного виконання задачі в заданий термін (ймовірність того, що i -тий запит буде виконаний);
- K_r — коефіцієнт готовності програмно-технічного комплексу;
- T_v — середній час відновлення програмно-технічного комплексу;
- T_e — мінімальний час між двома відмовами за календарний місяць.

Комплекс технічних засобів повинен передбачати:

- можливість забезпечення роботи з різних пристроїв і браузерів;
- можливість автоматичного масштабування системи при зростанні навантаження;
- мінімізацію часу відгуку системи.

Для забезпечення надійності програмного та інформаційного забезпечення необхідно передбачити використання:

- модульного, структурного і об'єктно-орієнтованого програмування;
- програмних засобів контролю вхідної інформації з видачею користувачу повідомлень про виявлені помилки;
- засобів захисту від збоїв, несанкціонованого доступу, помилкових дій персоналу тощо;
- системи контейнеризації для швидкого відновлення.

2.4.5. Вимоги до безпеки

Для забезпечення безпеки при експлуатації, налагодженні, монтажі, обслуговуванні і ремонті технічних засобів системи потрібно дотримуватись вимог ДСТУ: ДСТУ 2293-99, ДСТУ ISO 6309:2007, ДСТУ 12.0.230:2008, ДСТУ 7237:2011, ДСТУ 7238:2011, ДСТУ 7239:2011; по доступним рівням освітленості, вібраційних і шумових навантажень слід дотримуватися вимог відповідно ДСТУ Б А.3.2-15:2011, ДСТУ EN 14253:2018, ДСТУ 2867-94 [3, 4, 5].

2.4.6. Вимоги з ергономіки та технічної естетики

Загальні ергономічні і естетичні вимоги до системи повинні відповідати держстандартам ДСТУ 8604:2015, ДСТУ 7298:2013 [8]. Освітленість робочого місця повинна відповідати ДСТУ EN 12464-1:2016 [9], ДБН В.2.5-28-2006 [10].

Веб-інтерфейс повинен бути розроблений з урахуванням сучасних принципів UI/UX дизайну [28] та відповідати таким вимогам:

- адаптивність до різних розмірів екранів;
- інтуїтивно зрозумілий інтерфейс;
- оптимізація для швидкого завантаження;

2.4.7. Вимоги по експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Види обслуговування системи визначаються у відповідності з ДСТУ EN 13306:2019. Загальні вимоги по експлуатації, технічному обслуговуванню і ремонту повинні відповідати ДСТУ 3576-97 [6].

Для розміщення технічних засобів системи необхідні площі, визначені в ДБН В.2.2-9-2009 [11]. Напруга живлення технічних засобів системи 230 В змінного струму, частотою (50 ± 1) Гц. Допустиме відхилення напруги від +10 до -15%, тривалість перерв у живленні не повинна перевищувати 0,001 с.

Кількість, кваліфікація і режими роботи обслуговуючого персоналу повинні відповідати рекомендаціям, зазначеним в технічних умовах і інструкціях з експлуатації.

Регламент обслуговування повинен відповідати їх рівню і умовам роботи, щоб у випадку відмови системи забезпечити роботу в аварійному режимі.

2.4.8. Вимоги до захисту інформації від несанкціонованого доступу

Для надійності збереження і доступу до інформації необхідно використовувати засоби захисту:

1. на рівні серверної інфраструктури:
 - захист від несанкціонованого доступу , дозволивши доступ до системи лише на рівні локальної мережі приймальної комісії
2. на рівні веб-застосунку (Next.js):
 - захист від основних веб-вразливостей (OWASP Top 10) [19];
 - JWT-автентифікація та авторизація;
 - захист від XSS, CSRF та SQL-ін'єкцій.

Кожен сеанс роботи системи має розпочинатися з автентифікації користувача. Система має підтримувати різні рівні доступу відповідно до ролей користувачів. Для надійного захисту від несанкціонованого доступу кожен із працівників повинен мати персональний логін та пароль [7].

2.4.9. Вимоги щодо збереження інформації при аваріях

Контейнеризація за допомогою Docker повинна забезпечувати швидке перезавантаження системи на сервері у випадку виходу з ладу контейнера.

Усі дані повинні зберігатись в базі даних, яку можливо відновити або використовувати далі після виходу з ладу серверу.

2.4.10. Вимоги по захисту від впливу зовнішніх діянь

Електрична складова електромагнітного поля завад в приміщеннях не повинна перевищувати $0,3 \text{ В/м}^2$ в діапазоні частот від 0,15 до 300 МГц. Для захисту від впливу електромагнітних полів та індустриальних завад слід передбачити різноманітні екрани та фільтри.

Засоби, які виключають вплив шкідливих факторів на функціонування комплексу технічних засобів, повинні бути запроектовані згідно з ДБН В.2.2-9-2009. Обчислювальні засоби по стійкості до зовнішніх впливів повинні відповідати ДСТУ 2506-94.

2.4.11. Вимоги до патентної чистоти

При створенні даної системи патентні дослідження не проводяться. Система розробляється з використанням відкритих технологій та бібліотек з відповідними ліцензіями, що дозволяють їх використання у даному проєкті.

2.4.12. Вимоги по стандартизації і уніфікації

У системі кодування інформації необхідно проводити за світовим класифікатором і стандартом. При розробці веб-застосунку необхідно дотримуватись сучасних стандартів веб-розробки (HTML5 [13], CSS3 [14], ECMAScript [12], WCAG 2.1 [15]).

2.4.2. Вимоги до функцій

Перелік функцій із зазначенням вхідної та вихідної інформації наведено в Таблиці 2.1.

Функції мають забезпечити раціональну організацію роботи користувача на основі безперервної технології: заповнення бази даних, довідників, формування звітів і виконання інших функцій, визначених чинним документом.

Таблиця 2.1 Перелік функцій

№ п/п	Найменування функції	Вхідна інформація	Вихідна інформація
1	Реєстрація та автентифікація користувачів	Дані користувача (ім'я, email, пароль)	Статус автентифікації, сесія користувача
2	Формування та виведення списку вступників	Дані з таблиць БД	Таблиця зі списком вступників та їх даними
3	Управління даними вступників	Особисті дані вступника (ПІБ, паспортні дані, адреса, ПІН, телефон, та решта потрібних даних)	Форма з даними вступника та можливістю їх редагування
4	Формування договорів	Дані вступника, представника, освітньої програми	Згенеровані договори у форматі DOCX
5	Збереження даних вступників	Всі введені дані про вступника та обрану програму	Збережений запис у базі даних
6	Завантаження договорів	Дані вступника, тип договору	Файл договору або архів з договорами
7	Пошук та фільтрація даних	Параметри пошуку та фільтрації	Відфільтрований список вступників
8	Адміністрування інформаційної системи	Дані адміністратора	Панель адміністратора
9	Редагування внутрішніх таблиць	Дані адміністратора, нові дані для таблиць	Форма для створення нових записів в таблицях бази даних

2.4.3. Вимоги до видів забезпечення

Вимоги до математичного забезпечення (МЗ)

Система не вимагає спеціального математичного забезпечення для реалізації покладених на неї функцій. Достатньо можливостей обраної СУБД та фреймворку Next.js.

Вимоги до інформаційного забезпечення (ІЗ)

Інформаційне забезпечення системи повинно містити дані, достатні для виконання всіх покладених на систему функцій. ІЗ повинно гарантувати раціональну організацію зберігання інформації та доступу до неї.

Заповнення БД інформацією покладається на замовника за методиками і формами, створеними розробниками системи.

Склад, структура і спосіб організації інформації представляються у логічній моделі БД і можуть бути уточнені на етапі технічного проектування.

Слід передбачити захист даних від руйнування при аваріях і порушеннях у енергоживленні системи — використання контейнеризації за допомогою Docker.

Вимоги до лінгвістичного забезпечення (ЛЗ)

Для розробки програмних засобів, які реалізують виконання функцій і забезпечують сервіс користувачів повинні використовуватися сучасні мови програмування та фреймворки:

- TypeScript для фронтенд та бекенд розробки;
- Next.js як основний фреймворк для розробки веб-застосунку;
- SQL для роботи з базою даних MySQL.

Організація інтерфейсу користувача має будуватися на сучасних принципах веб-розробки з використанням інтуїтивно зрозумілих елементів управління, навігації та форм вводу даних. Інтерфейс має бути багатомовним з підтримкою щонайменше української та англійської мов.

Вимоги до програмного забезпечення (ПЗ)

Загальносистемне ПЗ має забезпечувати надійне і якісне виконання функціональних завдань системи. До загальносистемного ПЗ належить:

- операційна система серверної частини — Linux (Ubuntu/Debian) / Windows;
- система управління базами даних — MySQL, PHPMyAdmin;
- контейнеризація — Docker, Docker Compose / Docker Desktop;

Загальні вимоги до системного ПЗ можна сформулювати так:

- мінімальні вимоги до ресурсів технічних засобів;
- максимальна швидкодія;
- повне задоволення потреб функціональних завдань системи.

Вимоги до технологічного стеку:

- мінімальне використання ресурсів сервера для власних потреб;
- максимальна швидкодія при обробці запитів;

- масштабованість і можливість горизонтального розширення;
- безпечність та надійність.

Вимоги до СУБД:

- максимальне задоволення потреб функціональних задач;
- надійність;
- ефективне управління потрібного обсягу і структури;
- швидкість виконання запитів користувачів;
- підтримка транзакцій та забезпечення цілісності даних;
- мінімальні вимоги до технічних засобів.

Програмні засоби введення та виведення даних і ведення діалогу повинні забезпечувати:

- виведення необхідних даних на екран у вигляді зручних для користувача інтерфейсів;
- супровід введення даних контролем і сигналізацією користувачу про наявність помилок з можливістю їх виправлення під час введення даних;
- валідацію введених даних на стороні клієнта та сервера;
- виведення даних у відповідному вигляді (форма документа, Word) за запитом користувача;
- експорт даних у поширені формати (Word, Excel).

При розробленні спеціального ПЗ слід виконати наступні вимоги:

- використані технології мають бути сумісні між собою та із загальносистемним ПЗ;
- ПЗ має розроблятися з використанням сучасних підходів до розробки (компонентний підхід, функціональне програмування);
- забезпечити відповідність інтерфейсу користувача сучасним веб-стандартам;
- використовувати модульну структуру програм;
- передбачити можливість розширення складу задач у відповідності з новими функціональними потребами;
- забезпечити крос-браузерність.

Вимоги до технічного забезпечення

Технічні засоби системи (Таблиця 2.2) повинні забезпечувати виконання функцій, перерахованих в Таблиці 2.1.

Засоби обчислювальної техніки повинні забезпечувати обмін інформації в об'ємах, приведених в п. 4.3.

Таблиця 2.2 Вимоги до технічного забезпечення системи

№ п/п	Основні характеристики комп'ютера
Технічне забезпечення для сервера	
1	CPU: 2+ ядра, 2.5 ГГц+
2	RAM: 16+ ГБ
3	SSD/HDD: 50+ ГБ
4	Мережа: 100 Мбіт/с
Технічне забезпечення для клієнта	
1	Сучасний веб-браузер (Chrome, Firefox, Safari, Edge)
2	Підключення до локальної мережі: 10+ Мбіт/с

Вимоги до метрологічного забезпечення

Система не має вимірювальних каналів, вимірювального обладнання і приладів, тому вимоги до даного виду забезпечення не висуваються.

Вимоги до організаційного забезпечення

Організаційне забезпечення системи розробляється в відповідності з вимогами державного стандарту по АСУП.

При впровадженні системи не передбачається збільшення штатної чисельності університету. Територіальне розміщення робочих місць, на яких буде використовуватись система, визначається університетом.

До функціонування системи висуваються наступні вимоги:

- наказом ректора визначається список співробітників, які мають доступ до системи;
- контроль і прийняття рішень при аварійних ситуаціях при експлуатації системи здійснює відповідальний за систему.

2.5. Склад і зміст робіт по створенню системи

Календарний план розробки системи наведений в Таблиці 2.3.

Таблиця 2.3. Календарний план розробки системи

№	Робота	Початок, дні	Тривалість, дні
1	Аналіз вимог, збір інформації	1	5
2	Розробка форми введення даних	6	4
3	Реалізація функції генерації контрактів	6	7
4	Впровадження механізму вибору шаблонів	13	5
5	Створення адміністративного інтерфейсу	18	4
6	Тестування функціоналу	22	4
7	Демонстрація готової програми	26	2
8	Розгортання програми	28	2

Діаграма Ганта зображена на рисунку 2.1.

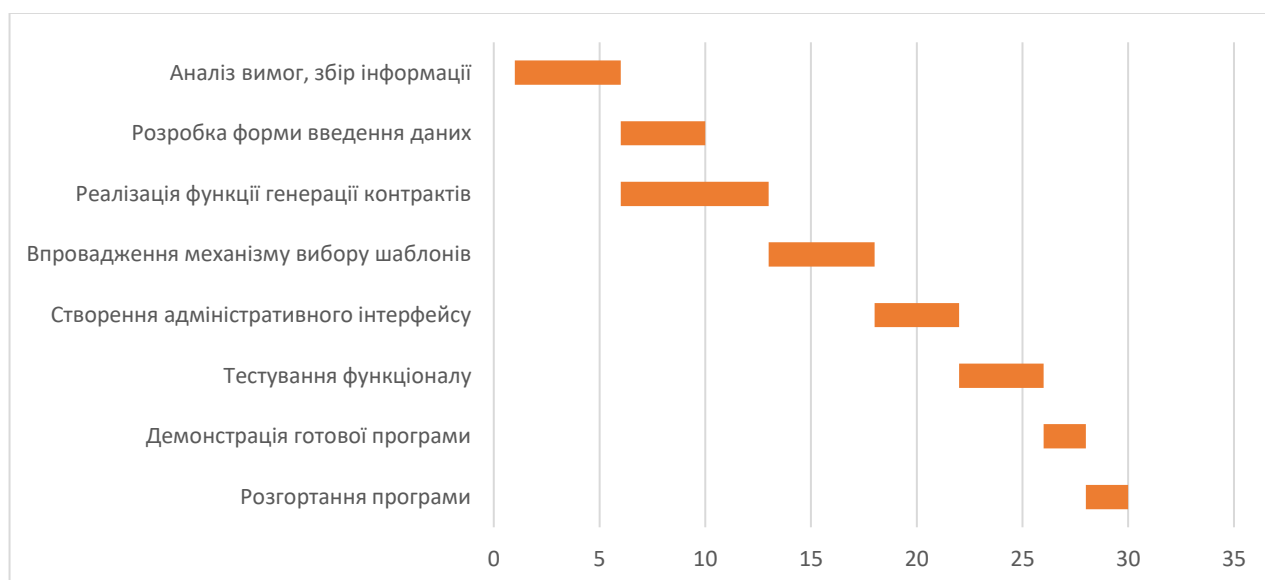


Рисунок 2.1 - Діаграма Ганта

2.6. Порядок контролю і приймання системи

2.6.1. Випробування системи

Система вводиться на діючій інфраструктурі НУХТ. При введенні в дію система повинна пройти приймальні випробування згідно з ДСТУ 3974-2000.

2.6.2. Програма випробувань

Випробування для визначення працездатності і рішення про можливість приймання системи в дослідну експлуатацію проводять розробники разом із замовником. Програму випробувань складає розробник і затверджує замовник. Програма випробувань повинна включати:

- перевірку повноти і якості реалізації функцій системи;
- перевірку надійності і стійкості системи;
- тестування користувацького інтерфейсу;
- перевірку безпеки системи;
- тестування процесу формування контрактів;
- тестування інтеграції основних компонентів системи.

2.6.3. Дослідна експлуатація

Здача в дослідну експлуатацію здійснюється на основі технічного завдання та інструкції користувача. За результатами дослідної експлуатації формується перелік доробок і рекомендовані строки їх виконання. Тривалість дослідної експлуатації складає один місяць.

2.6.4. Приймання системи

Введення в дію системи оформлюється актом здачі-прийому. У процесі приймання проводиться оцінка відповідності системи вимогам технічного завдання. При виявленні невідповідностей вони документуються та визначаються терміни їх усунення.

2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію

Для введення в дію замовник виконує ряд робіт із підготовки об'єкта:

- проводить укомплектування технічних засобів відповідно до вимог, зазначених у розділі 2.4.3;
- забезпечує необхідну мережеву інфраструктуру для розгортання системи;
- надає доступ до серверних потужностей для встановлення Docker-контейнерів;
- організовує навчання користувачів системи роботі з веб-інтерфейсом і вивчення інструкції з її експлуатації;
- забезпечує наявність технічного персоналу для підтримки функціонування системи;
- проводить дослідну експлуатацію і вводять систему в дію згідно з встановленим графіком;
- забезпечує збір та аналіз зауважень користувачів під час дослідної експлуатації.

2.8. Вимоги до документації

2.8.1. Склад документації

На систему розробляється комплекс документації у складі:

- технічне завдання;
- технічний проект;
- програма і методика випробувань;
- документація для користувача;
- документація для адміністратора;
- документація на розгортання системи за допомогою Docker;
- опис програмного інтерфейсу (API) системи.

2.8.2. Стандарти документації

Документація на систему розробляється у відповідності з вимогами Державних стандартів серії 19 «Єдина система програмної документації» та серії 24 «Єдина система стандартів автоматизованих систем управління».

2.8.3. Вимоги до оформлення документації

Документація повинна бути виконана на українській мові. Технічна документація може містити посилання на англomовні джерела та стандарти, якщо це необхідно для опису технології або особливостей розробки.

Документація повинна бути доступна в електронному вигляді (формати MDX та HTML).

Дане технічне завдання складено згідно з методичними рекомендаціями кафедри [30].

РОЗДІЛ 3. СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Опис та обґрунтування вибору програмно-технічних засобів розроблення програмного продукту

Для розробки інформаційної системи формування контрактів для вступників НУХТ було використано сучасний стек веб-технологій, який дозволяє створити гнучку, швидку стабільну та зручну у використанні систему [27].

Основним фреймворком для розробки проєкту було обрано - Next.js від компанії Vercel, через його потужні можливості для створення швидких та оптимізованих веб-додатків. Цей фреймворк базується на React і надає зручні інструменти для серверного (SSR , Server side rendering) та клієнтського рендерингу, маршрутизації та API розробки.

Переваги використання Next.js у проєкті [16]:

- вбудована маршрутизація, що спрощує навігацію між сторінками застосунку;
- API маршрути для створення ендпоінтів для генерації контрактів та взаємодією з базою даних;
- швидкий час завантаження сторінок завдяки усім оптимізаціям цього фреймворку , як кешування , пре-рендеринг а також стрімінг компонентів в статичні сторінки.

React було обрано за його простоту у розробці , створення компонентів які можуть бути повторно використані у різних частинах системи , а також за його популярність та високу підтримку спільноти, що дозволяє швидко знаходити готові рішення та бібліотеки для реалізації різних типів задач [21].

Також для розробки візуальної частини проєкту використовується TailwindCSS — сучасний CSS-фреймворк, що дозволяє швидко створювати адаптивні та естетично привабливі інтерфейси. TailwindCSS використовує підхід utility-first, який дозволяє застосовувати стилі безпосередньо у HTML та JSX

розмітці, що збільшує швидкість розробки та водночас зменшує кінцевий файл стилів.

Усе це разом дає змогу використовувати ShadCN – набір стилізованих компонентів, код яких додається прямо в проект. Тобто, в порівнянні зі звичайними UI бібліотеками, як от Bootstrap або Material UI, ShadCN копіює всі потрібні компоненти прямо в папку застосунку, що дозволяє швидко додавати потрібні зміни, а також розширювати функціонал цих компонентів, на відміну від традиційних бібліотек, де весь код уже скомпільований і не може бути легко змінений. Також додатковим плюсом ShadCN є те, що він оснований на headless-UI (без жодних стилів) бібліотеці Radix UI, яка надає компоненти, які працюють на всіх пристроях, адаптовані для доступності і надійності [24].

Для збереження даних використовується реляційна база даних MySQL. Цей вибір обумовлений надійністю, продуктивністю та широкою підтримкою MySQL. База даних зберігає інформацію про навчальні підрозділи, освітні програми, користувачів системи та збережені контракти [17].

Для генерації документів контрактів у форматі DOCX використовується бібліотека Docxtemplater, яка дозволяє заповнювати шаблони документів даними з системи.

Також весь проект використовує мову Typescript, що на відміну від Javascript [20] має підтримку типізації, що дозволяє уникнути багатьох помилок під час розробки.

Загалом весь стек описано в таблиці 3.1

Таблиця 3.1 Стек технологій системи

Компонент	Технологія	Мета
Фронтенд	Next.js	Серверний рендеринг, маршрутизація та API маршрути.
Мова програмування	TypeScript	Розробка безпечного та типізованого коду
Стилізування	Tailwind CSS	Utility-first CSS фреймворк
Інтерфейс	Shadcn UI & Radix UI	Головна UI система застосунку
База даних	MySQL	Збереження даних
Авторизація	NextAuth.js	Авторизація користувачів та керування сесіями
Генерація документів	docxtemplater, exceljs	Генерація Excel та Word документів
Управління даними	SWR	Кешування та отримання даних
Теми	next-themes	Підтримка різних тем інтерфейсу
Сповіщення	sonner	Система сповіщень в застосунку

Архітектура застосунку

Застосунок розроблено за принципами сучасної веб-розробки з використанням мікросервісної архітектури (Microservices Architecture [26]):

- клієнтська частина (frontend) написана на React з використанням Next.js;
- серверна частина (backend) реалізована на Node.js з використанням API-маршрутів Next.js;
- база даних MySQL для збереження даних;
- контейнеризація за допомогою Docker для спрощення розгортання.

Схему високорівневої архітектури застосунку можна побачити на рисунку 3.1.

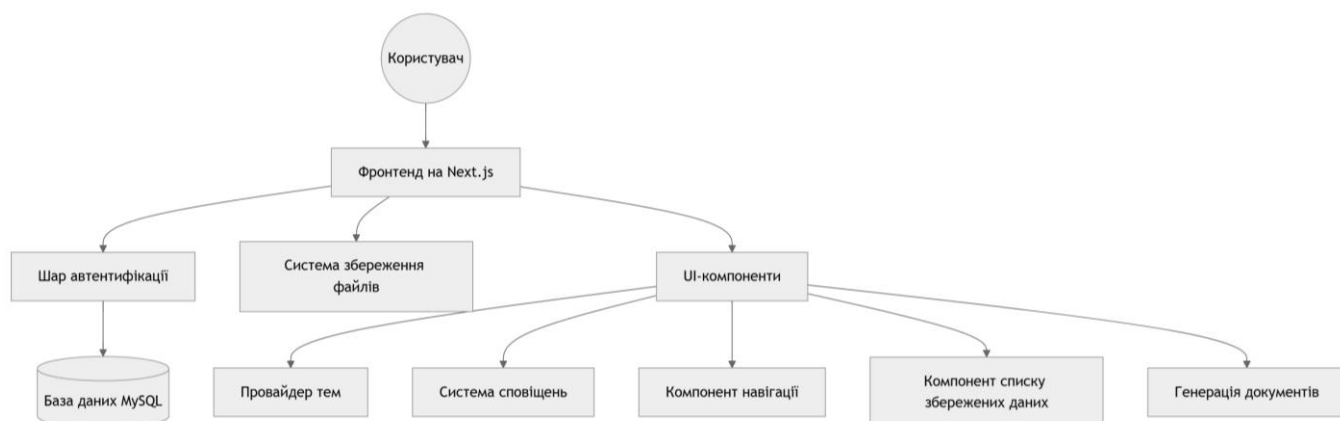


Рисунок 3.1 - Високорівнева схема архітектури застосунку

3.2. Проектування та створення бази даних

База даних проєкту складається з кількох ключових таблиць, що зберігають та організовують дані, необхідні для роботи системи (Рис. 3.2).

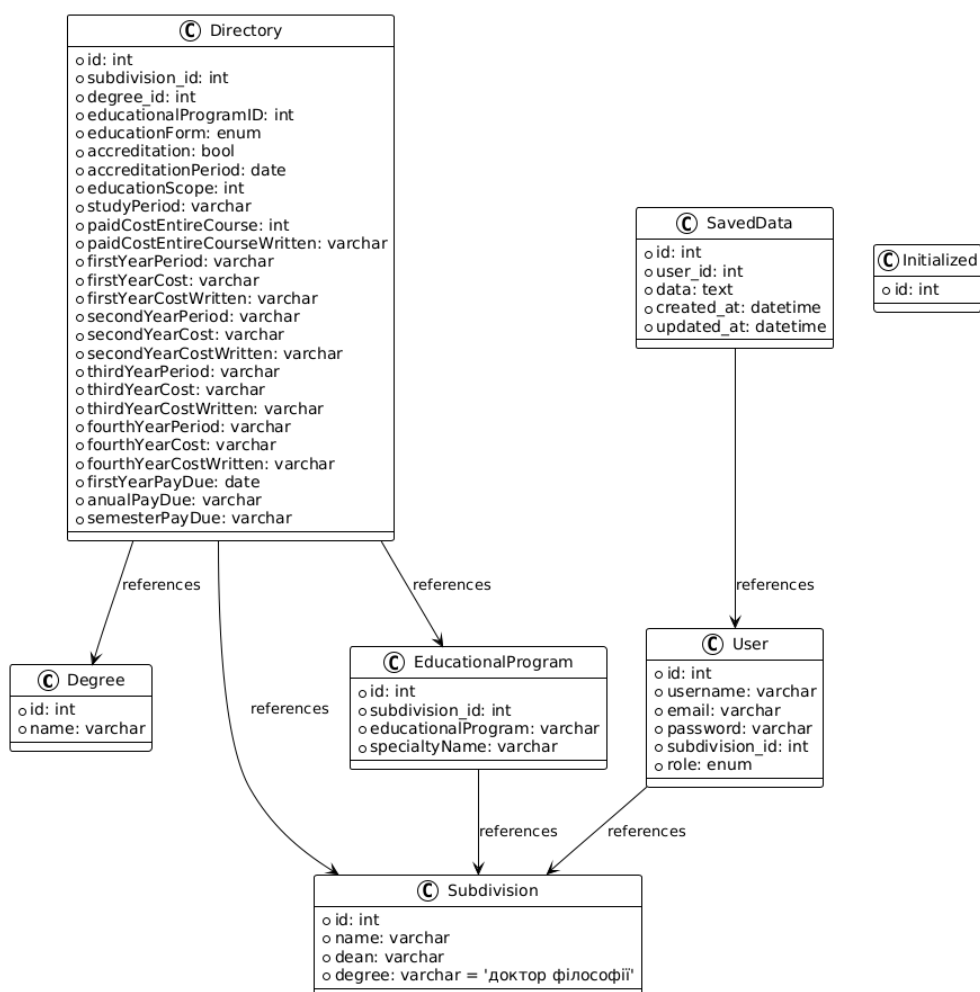


Рисунок 3.2 - Схема бази даних

Таблиця користувачів (users)

Зберігає інформацію про користувачів системи, включаючи адміністраторів та звичайних користувачів. Поля таблиці:

- id - унікальний ідентифікатор користувача
- username - ім'я користувача
- email - електронна пошта
- password - хешований пароль
- subdivision_id - ідентифікатор підрозділу
- role - роль користувача (admin або user)

Таблиця підрозділів (subdivisions)

Містить інформацію про навчальні підрозділи НУХТ:

- id - унікальний ідентифікатор підрозділу
- name - назва підрозділу
- dean - декан підрозділу
- degree - науковий ступінь

Таблиця освітніх програм (educational_program)

Зберігає дані про доступні освітні програми:

- id - унікальний ідентифікатор програми
- subdivision_id - ідентифікатор підрозділу, до якого належить програма
- educationalProgram - назва освітньої програми
- specialtyName - назва та код спеціальності

Таблиця наукових ступенів (degree)

Містить перелік наукових ступенів:

- id - унікальний ідентифікатор ступеня
- name - назва наукового ступеня

Таблиця довідника (directory)

Основна таблиця, що містить інформацію про освітні програми, їхню вартість, терміни навчання та інші деталі для формування контрактів:

- Інформація про підрозділ, форму навчання, освітню програму
- Дані про акредитацію програми

- Терміни навчання та оплати
- Вартість навчання, включаючи розбивку по роках

Таблиця збережених даних (saved_data)

Зберігає дані заповнених контрактів для можливості їх подальшого редагування:

- id - унікальний ідентифікатор запису
- user_id - ідентифікатор користувача, який створив запис
- data - серіалізовані дані контракту (дані контракту у форматі JSON)
- created_at - дата створення запису
- updated_at - дата оновлення запису

Зв'язки між таблицями

База даних має наступні ключові зв'язки між таблицями:

- Users мають Subdivision (зовнішній ключ subdivision_id)
- Directory містить Subdivisions (зовнішній ключ subdivision_id)
- Directory містить Degree (зовнішній ключ degree_id)
- Directory містить Educational_program (зовнішній ключ educationalProgramID)
- Educational_program має Subdivisions (зовнішній ключ subdivision_id)
- Saved_data створений User (зовнішній ключ user_id) і

Також для оптимізації роботи з базою даних було створено представлення (view) directory_with_program, яке об'єднує дані з таблиць directory та educational_program

3.3 Реалізація функцій системи

Сторінка авторизації – забезпечує контрольований доступ до системи для членів приймальної комісії та адміністраторів (Рис. 3.3).

Вітаю

Введіть ваші дані для входу

Email

m@example.com

Пароль

Password

Увійти

Рисунок 3.3 - Форма авторизації застосунку

Сторінка управління контрактами – містить таблицю збережених контрактів із можливістю фільтрації за навчальним підрозділом авторизованого користувача, за фінансуванням та за спеціальністю , пошуку, редагування, видалення та завантаження файлів для друку (Рис. 3.4).

Привіт, Administrator

ДокументаціяАдміністрування

Пошук по імені вступника

Скинути фільтриФільтри

ПІБ вступника	ПІБ представника	Підрозділ	Спеціальність	Фінансування	Дата створення	Ступінь	Дата оновлення	
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Факультет біотехнології та екологічного контролю	162 Біотехнології та біоінженерія	Контракт фіз. особа	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Навчально-науковий інженерно-технічний інститут імені акад. І.С.Гулога	133 Галузева машинобудування	Контракт юр. особа	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Факультет біотехнології та екологічного контролю	162 Біотехнології та біоінженерія	Контракт фіз. особа	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Факультет біотехнології та екологічного контролю	162 Біотехнології та біоінженерія	Бюджет	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Факультет біотехнології та екологічного контролю	162 Біотехнології та біоінженерія	Контракт фіз. особа	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Навчально-науковий інститут економіки і управління	075 Маркетинг	Бюджет	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Навчально-науковий інженерно-технічний інститут імені акад. І.С.Гулога	141 Електроенергетика, електротехніка та електромеханіка	Контракт фіз. особа	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Факультет біотехнології та екологічного контролю	162 Біотехнології та біоінженерія	Контракт фіз. особа	12.05.2025	доктор філософії	12.05.2025	...

Рядків на сторінці: 8

Сторінка 1 з 2

Рисунок 3.4 – Сторінка управління контрактами

Сторінка створення контракту - надає зручну форму для введення даних вступника, де поля заповнюються або обираються з випадаючих списків, сформованих на основі даних із бази (Рис. 3.5).

← Назад

Крок 1. Оберіть навчальний підрозділ та фінансування

Підрозділ не обрано
Оберіть підрозділ зі списку нижче

Оберіть підрозділ...

Фінансування навчання

Оберіть фінансування

Далі

Рисунок 3.5 - Сторінка створення контракту

Адміністративна панель — дозволяє адміністратору додавати нових користувачів із прив'язкою до навчального підрозділу редагувати довідкові дані через інтерфейс, а також виконувати додаткові налаштування, такі як ініціалізація та очищення бази даних, експорт , редагування таблиць бази даних без використання додаткових інструментів, а також редагування вбудованих шаблонів для генерації контрактів (Рис. 3.6).

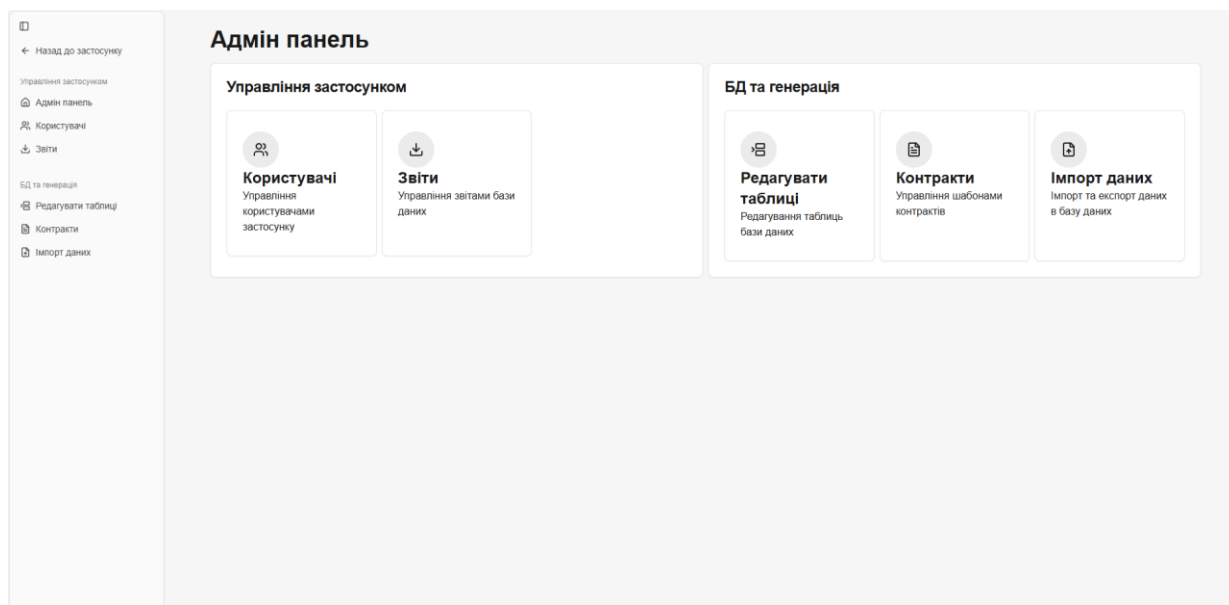


Рисунок 3.6 – Адміністративна панель застосунку

Імпорт даних - передбачено можливість завантаження Excel-файлів з усіма даними в базі даних для зручного редагування. А також є можливість імпорту з Excel-файлу назад в базу даних (Рис. 3.7).

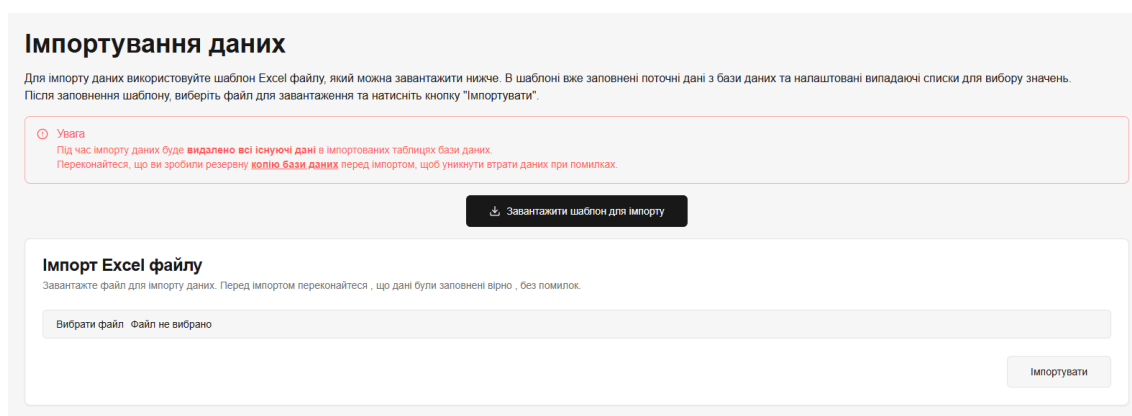


Рисунок 3.7 - Форма імпортування даних

Документація – містить набір сторінок, де описаний принцип роботи з системою для звичайних користувачів та адміністраторів (Рис. 3.8).

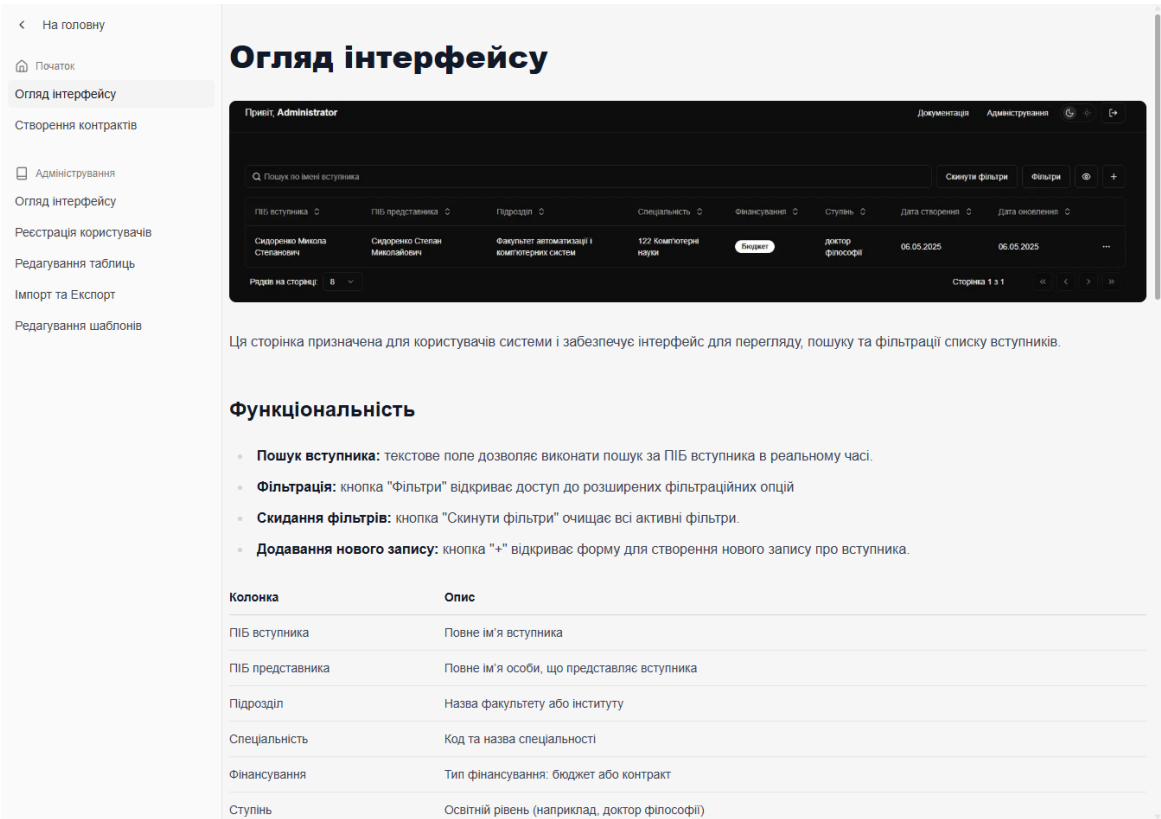


Рисунок 3.8 - Сторінка в документації застосунку

Основна функціональність системи реалізована через багатокрокову форму для створення контрактів та API для генерації документів. Розглянемо ключові функції системи детальніше.

Основні потоки даних у застосунку працюють за схожими принципами, як наведено на рисунку 3.9.

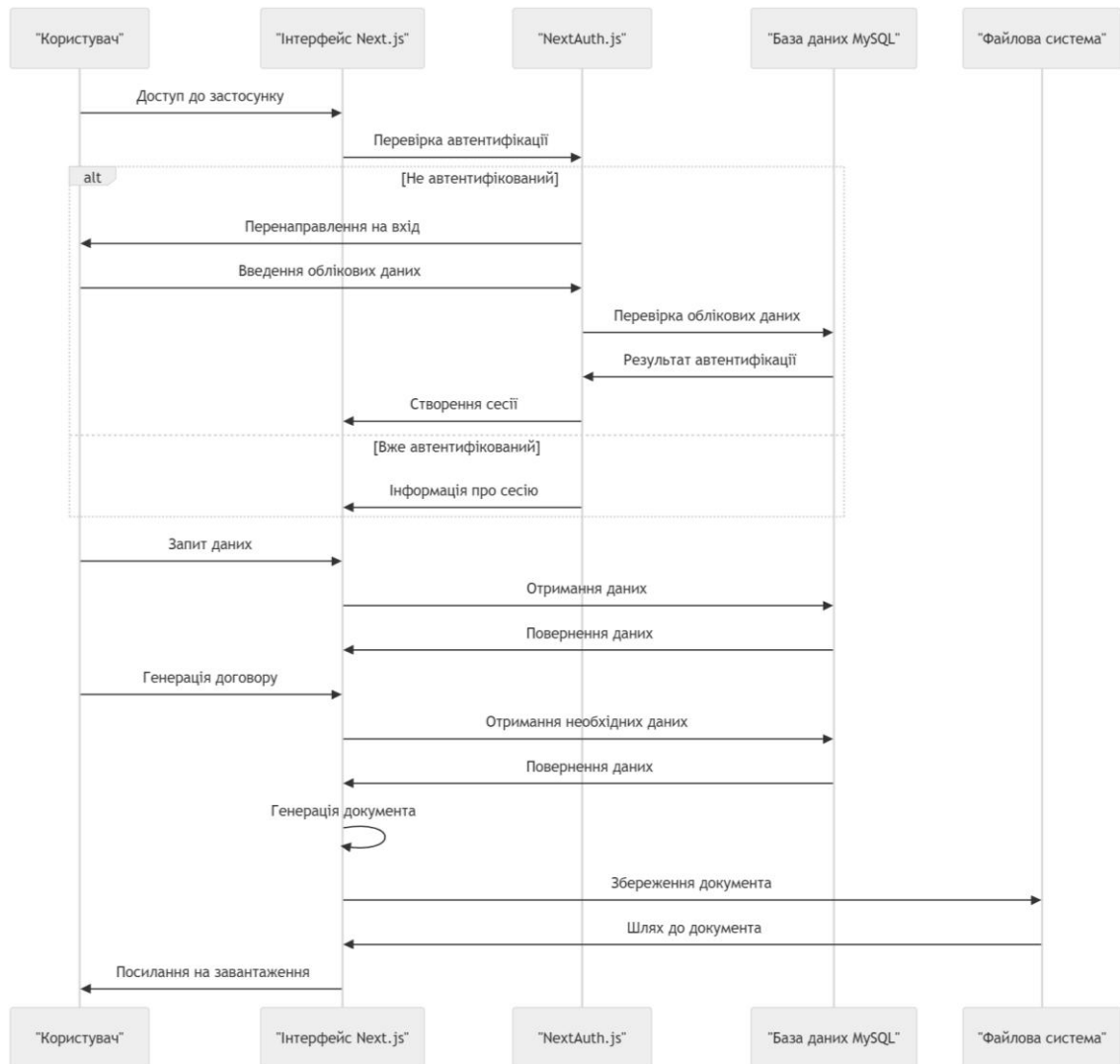


Рисунок 3.9 – Потік даних при завантаженні документів

Код , відповідальний за обробку запиту на завантаження документів відображено на рисунку 3.10.

```

export const POST = auth( async (req) => {
  try{
    if(!req.auth) return NextResponse.json({ error: "Unauthorized" }, { status: 401 });
    const initialData = await req.json();
    const requiredFields = ['directory_id', 'subdivision', 'directory_id']; // Перевірка на обов'язкові поля
    if (!requiredFields.every(field => field in initialData && initialData[field] !== '')) {
      return NextResponse.json({ error: "Обов'язкові поля відсутні" }, { status: 400 });
    }
    const data = await getVariables(initialData); // Мабинг змінних у файлах в об'єкт
    const fileNames = [EDU_FILE_NAME]; // Список файлів для генерації
    const endFileNames = [`_Договір_навчання_${new Date().getFullYear()}.docx`];
    const endPrefix = initialData?.pib_vstup.replace(' ', '_');
    if(initialData?.financing === 'phys'){
      fileNames.push(PHYS_FILE_NAME);
      endFileNames.push(`_Договір_фіз_особа_${new Date().getFullYear()}.docx`);
    }
    if(initialData?.financing === 'jur'){
      fileNames.push(JUR_FILE_NAME);
      endFileNames.push(`_Договір_юр_особа_${new Date().getFullYear()}.docx`);
    }
    const stream = new ReadableStream({ // Використовуємо ReadableStream для потокової передачі архіву
      async start(controller) {
        const archive = archiver('zip', { zlib: { level: 9 } });
        archive.on('data', (chunk) => controller.enqueue(chunk));
        archive.on('end', () => controller.close());
        archive.on('error', (err) => controller.error(err));
        // Додаємо файли до архіву
        for (const fileName of fileNames) {
          const buffer = await generateFile(data, fileName);
          archive.append(buffer, { name: endPrefix +
            endFileNames[fileNames.indexOf(fileName)]
          });
        }
        archive.finalize();
      }
    });
    return new NextResponse(stream, {
      status: 200,
      headers: {
        'Content-Type': 'application/zip',
        'Content-Disposition': `attachment; filename="contracts.zip"`,
      },
    });
  } catch (err) {
    return NextResponse.json({ error: (err as Error).message || (err as { code: string })?.code }, { status: 500 });
  }
});

```

Рисунок 3.10 - Обробник POST запиту на завантаження документів

Форма створення/редагування контракту

Форма складається з 5 кроків, кожен з яких відповідає за заповнення певної частини даних:

1. Вибір навчального підрозділу та фінансування (Рис. 3.11):

- Вибір навчального підрозділу
- Вибір типу фінансування (бюджет, контракт фізична особа, контракт юридична особа)
- Вибір строків здійснення оплати для контрактної форми

← Назад

Крок 1. Оберіть навчальний підрозділ та фінансування

Факультет біотехнології та екологічного контролю
Наталія ГРЕГІРЧАК (доктор філософії)

Факультет біотехнології та екологічного контролю

Фінансування навчання
Контракт фіз. особа

Строки здійснення оплати
кожного семестру

Далі

Рисунок 3.11 - Перший крок форми створення контракту

1.5 Дані юридичної особи , яка відображається тільки при виборі фінансування «Контракт юридична особа» (Рис. 3.12)

- Форма для введення даних про юридичну особу
- Поля для назви організації, представника, підстави діяльності та реквізитів

← Назад

Крок 1.5 Дані юридичної особи

Організаційно-правова форма та назва юридичної особи, яка замовляє освітню послугу
Введіть назву юридичної особи

Посада і представник замовника
Введіть посаду і представника замовника

Замовник діє на підставі
Введіть підставу, на якій діє замовник

ІП представника
Олена ГОЛОВАЧ

Реквізити замовника
Введіть реквізити замовника

Назад Далі

Рисунок 3.12 – Введення даних про юридичну особу на формі створення контракту

2. Загальні дані вступника (Рис. 3.13)

- Форма для введення ПІБ, паспортних даних, місця проживання
- Поля для ПН та контактних телефонів

← Назад

Крок 2. Загальні дані вступника

ПІБ вступника
Сидоренко Микола Степанович

Серія та номер, ким виданий паспорт вступника
123456789, 4321, 01.01.2000

Місце проживання (реєстрації) вступника
Київська область, Бучанський район, с. Шпитьки, вул. Шевченка, буд. 57

Реєстраційний номер облікової картки платника податків вступника
1234567890

Номери телефонів
0931234567; 0978765432

Назад Далі

Рисунок 3.13 - Другий крок форми створення контракту

3. Загальні дані законного представника (Рис. 3.14)

- Необов'язкова форма для введення даних про законного представника
- Аналогічні поля як для вступника

← Назад

Крок 3. Загальні дані законного представника

ПІБ законного представника
Сидоренко Микола Степанович

Серія та номер, ким виданий паспорт законного представника
123456789, 4321, 01.01.2000

Місце проживання (реєстрації) законного представника
Київська область, Бучанський район, с. Шпитьки, вул. Шевченка, буд. 57

Реєстраційний номер облікової картки платника податків законного представника
1234567890

Номери телефонів законного представника
0931234567; 0978765432

Назад Далі

Рисунок 3.14 - Третій крок форми створення контракту

4. Вибір освітньої програми (Рис. 3.15)

- Вибір ступеня вищої освіти, форми навчання, освітньої програми
- Відображення детальної інформації про обрану програму

Крок 4. Освітня програма

Ступінь вищої освіти
доктор філософії

Форма навчання
денна

Назва освітньої програми
Біотехнологія

Назад Далі

Попередній перегляд
Обрана освітня програма

Назва освітньої програми
Біотехнологія

Назва та код спеціальності
162 Біотехнології та біоінженерія

Форма навчання
денна

Загальний термін навчання
2024-2028

Вартість навчання
205 200 грн.

Акредитація
акредитована до 01.07.2027

Навчальний підрозділ
Факультет біотехнології та екологічного контролю

Рисунок 3.15 - Четвертий крок форми створення контракту

5. Збереження та завантаження документів (Рис. 3.16)

- Можливість завантажити архів з усіма документами
- Можливість завантажити окремі документи
- Збереження введених даних в системі

Крок 5. Збереження та завантаження документів

Усі дані заповнені, ви можете прямо зараз завантажити архів з усіма документами, або окремо кожен документ.

Скачати архів ▾ Назад Зберегти

Договір про навчання
Договір юр. особа

Рисунок 3.16 – Фінальний крок форми створення контракту

Генерація документів контрактів

Система підтримує генерацію трьох типів контрактів:

1. Договір про навчання (обов'язковий для всіх)
2. Договір з фізичною особою (при виборі відповідного типу фінансування)
3. Договір з юридичною особою (при виборі відповідного типу фінансування)

Процес генерації реалізовано у API-ендпоінтах `/api/generate`, та схему їх роботи можна побачити на рисунку 3.17.

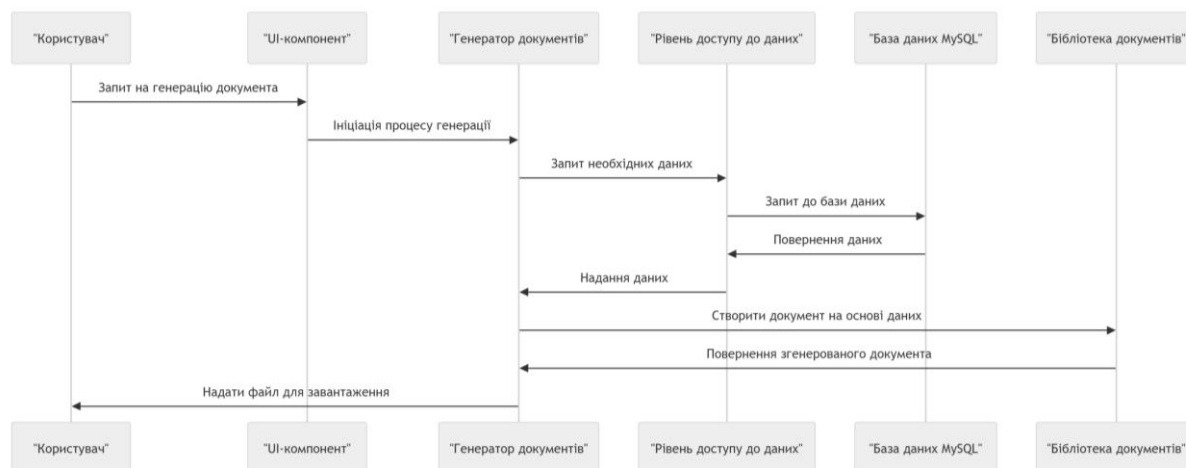


Рисунок 3.17 - Процес генерації документів.

Алгоритм роботи:

1. Отримання даних з форми
2. Перевірка обов'язкових полів
3. Підготовка змінних для заміни у шаблонах документів
4. Визначення необхідних типів документів залежно від типу фінансування
5. Створення архіву з документами
6. Повернення архіву/файлів користувачу для завантаження

Збереження та редагування даних

Система дозволяє зберігати заповнені форми у базі даних та редагувати їх пізніше:

Авторизація та розмежування доступу

Для захисту системи використовується авторизація за допомогою NextAuth, її реалізацію можна побачити на Рисунку 3.18.

Система має два типи користувачів:

- Адміністратори — можуть керувати шаблонами документів та користувачами
- Звичайні користувачі — можуть створювати та редагувати вступників.

```

providers: [
  Credentials({
    credentials: {
      email: {},
      password: {},
    },
    authorize: async (credentials) => {
      const { email, password } = credentials;
      if(typeof email !== 'string' || typeof password !== 'string'){
        throw new InvalidEmailOrPassword();
      }

      const [users] = await pool.query('SELECT * FROM users WHERE email = ?', [email]);
      const typedUsers = users as User[];
      if(typedUsers.length === 0){
        throw new InvalidLoginError();
      }
      const loginUser = typedUsers[0];

      //compare password
      const valid = await compare(password, loginUser.password);
      if(!valid){
        throw new InvalidLoginError();
      }
      return {
        id: loginUser.id.toString(),
        username: loginUser.username,
        email: loginUser.email,
        role: loginUser.role,
        subdivision_id: loginUser.subdivision_id
      }
    },
  )],
],

```

Рисунок 3.18 - Код відповідальний за авторизацію

3.4. Інструкція користувача

3.4.1. Інструкція для звичайних користувачів системи

Вхід у систему

1. Відкрийте головну сторінку системи у веб-браузері
2. На сторінці входу введіть свої облікові дані:
 - Логін (email на який був зареєстрований аккаунт адміністратором)
 - Пароль
3. Натисніть кнопку «Увійти»

Після успішного входу ви потрапите на головну сторінку системи, яка містить:

- Верхню навігаційну панель з можливістю змінити тему застосунку , перейти до документації а також перейти до адміністрування, якщо вхід був здійснений адміністратором
- Таблицю з усіма збереженими контрактами , а також різні частини управління нею , такі як пагінація , фільтрування , налаштовування колонок таблиці а також пошук по ПІБ вступника.

Фільтрація можлива за трьома критеріями , такі як Навчальний підрозділ , спеціальність та тип фінансування. Фільтр навчального підрозділу ставиться автоматично , якщо обліковий запис аккаунта користувача містить в собі поле підрозділа.

Для того , щоб відкрити меню фільтрації потрібно натиснути кнопку «Фільтри», що покаже усі випадаючі списки для фільтрування (Рис. 3.19).

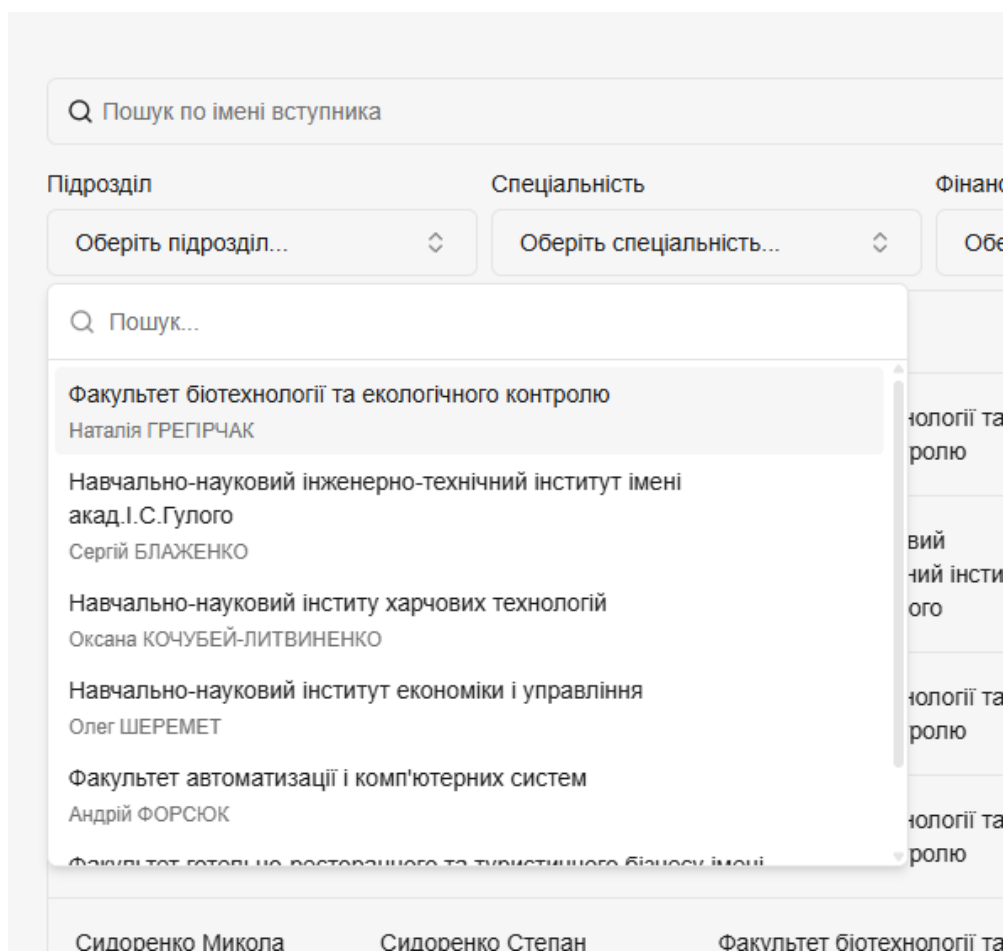


Рисунок 3.19 - Випадаючий список фільтрування за підрозділом

Також є управління налаштуваннями таблиці , а саме можливість приховати непотрібні стовпці , або перемістити їх як зручно користувачу. Ці налаштування зберігаються в Local Storage браузера , що дозволяє не налаштовувати таблицю при кожному завантаженні застосунку.

Для того щоб налаштувати видимість таблиць , потрібно натиснути кнопку з іконкою людського ока , що відкриє випадаючий список , де можна приховати або показати конкретні колонки таблиці (Рис. 3.20).

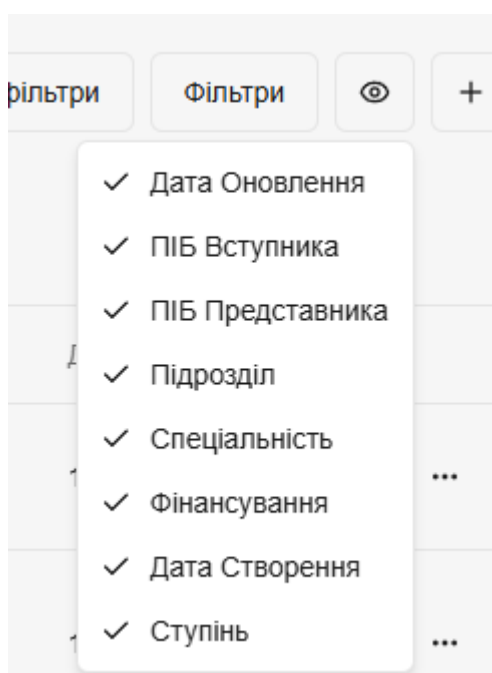


Рисунок 3.20 - Випадаючий список видимості колонок

Також для приховування, переміщення або сортуванню по конкретній колонці, потрібно натиснути на її заголовок в таблиці , що відкриє меню з налаштуванням сортування , видимості та двома кнопками зі стрілками для переміщення вліво та вправо (Рис. 3.21). Якщо переміщення в потрібну сторону недоступне , тобто колонка знаходиться в самому лівому або правому краю таблиці, то відповідна кнопка стає неактивною.

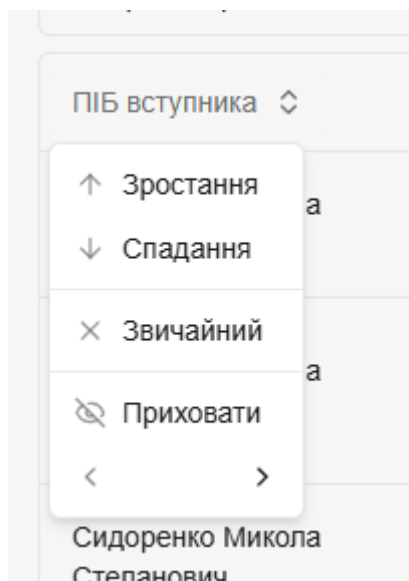


Рисунок 3.21 - Меню управління колонкою таблиці

Для завантаження, видалення або редагування , в таблиці біля кожного запису є кнопка додатковий дій , з відповідною іконкою – три крапки. При її натисканні , відкривається меню з трьома опціями – завантажити контракти (які будуть збережені в архіві), редагувати та видалити , що можна побачити на рисунку 3.22. При натисканні на редагування , відкривається форма створення контракту , але з заповненими даними, готовими до редагування.

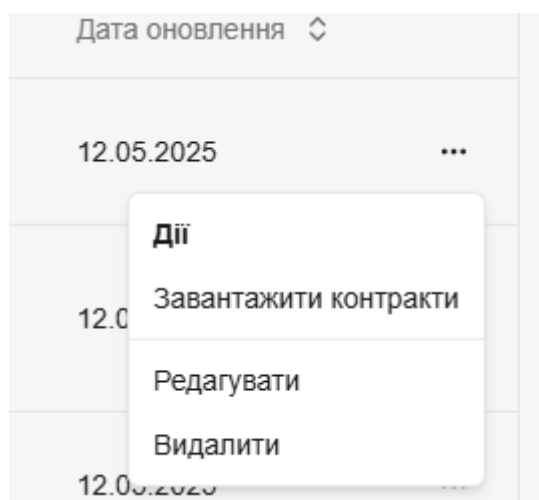


Рисунок 3.22 - Меню дій над збереженими контрактами

Створення нового контракту

Для створення нового контракту , потрібно на головній сторінці натиснути кнопку зі знаком «+» , що відкриє сторінку з формою створення контрактів , що розбита на п'ять кроків для зручного заповнення:

Крок 1: Вибір навчального підрозділу та фінансування

- Оберіть навчальний підрозділ зі списку запропонованих варіантів
- Оберіть тип фінансування:
 - Бюджет
 - Контракт фіз. особа
 - Контракт юр. особа
- Якщо обрано контрактну форму, виберіть строки здійснення оплати:
 - щороку
 - кожного семестру
- Натисніть кнопку «Далі» для переходу до наступного кроку

Крок 1.5: Дані юридичної особи (якщо обрано фінансування "Контракт юр. особа")

- Заповніть наступні поля:
 - Організаційно-правова форма та назва юридичної особи
 - Посада і представник замовника
 - Інформація про підставу, на якій діє замовник
 - ІП представника
 - Реквізити замовника
- Натисніть кнопку «Далі» для переходу до наступного кроку

Крок 2: Загальні дані вступника

- Заповніть ПІБ вступника
- Введіть паспортні дані (серію, номер, ким і коли виданий)
- Вкажіть місце проживання (реєстрації) вступника
- Введіть реєстраційний номер облікової картки платника податків (ПІН)
- Вкажіть номери телефонів (можна вказати кілька через крапку з комою)
- Натисніть кнопку «Далі» для переходу до наступного кроку

Крок 3: Загальні дані законного представника

- Якщо необхідно, заповніть наступні поля даними законного представника:
 - ПІБ законного представника
 - Серія та номер паспорта, ким виданий
 - Місце проживання (реєстрації) законного представника
 - Реєстраційний номер облікової картки платника податків
 - Номери телефонів законного представника
- Натисніть кнопку «Далі» для переходу до наступного кроку

Крок 4: Освітня програма

- Оберіть ступінь вищої освіти
- Виберіть форму навчання (денна/заочна)
- Оберіть назву освітньої програми зі списку доступних для обраного підрозділу та форми навчання
- Є можливість переглянути детальну інформацію про обрану програму у картці попереднього перегляду, яка знаходиться справа від основної форми та містить такі поля:
 - Назву освітньої програми
 - Назву та код спеціальності
 - Форму навчання
 - Загальний термін навчання
 - Вартість навчання
 - Інформацію про акредитацію
 - Навчальний підрозділ
- Натисніть кнопку «Далі» для завершення заповнення

Крок 5: Збереження та завантаження документів

- Застосунок пропонує два варіанти дій:
 - Завантажити архів з усіма документами, натиснувши кнопку «Скачати архів»

- Завантажити окремі документи через випадаюче меню (кнопка зі стрілкою вниз):
 - Договір про навчання
 - Договір фіз. особа (якщо обрано контракт фіз. особа)
 - Договір юр. особа (якщо обрано контракт юр. особа)
- Для збереження даних в системі натисніть кнопку «Зберегти»

3.4.2 Інструкція для адміністраторів

Для користувачів з роллю адміністратора доступна спеціальна адміністративна панель, в яку можна потрапити натиснувши кнопку Адміністрування в навігаційному меню головної сторінки застосунку. Відкриється головне меню адміністрування, з бічним меню для навігації по різним частинам панелі адміністратора, а також кнопками , які дублюють позиції в бічному меню для швидкої навігації.

Управління користувачами

Для створення нових користувачів , або часткового редагування існуючих користувачів , потрібно натиснути відповідну позицію в бічному меню навігації, що відкриє сторінку управління користувачами , яка показана на рисунку 3.23.

Управління користувачами

☐	Ім'я	Email	Роль	Підрозділ	
☐	Administrator	admin@example.com	Адміністратор	Оберіть підрозділ...	Видалити
☐	TestUser	test	Користувач	Факультет біотехнології та екологічного контролю	Видалити

0 з 2 рядків обрано.

Рядків на сторінці: 8

Сторінка 1 з 1

Додати нового користувача

Ім'я:

Email:

Пароль:

Роль:

Підрозділ:

Рисунок 3.23 - Сторінка управління користувачами

На цій сторінці є таблиця зі списком усіх користувачів, а також можливість зміни їх ролі і підрозділу через випадаючий список, видалення користувача. Обраний підрозділ буде використовуватись як стандартний фільтр по підрозділу на головній сторінці застосунку, що забезпечить зручніше використання застосунку в межах одного навчального підрозділа.

Також на цій сторінці є форма створення нового користувача з такими полями:

- Ім'я користувача
- Email користувача, за яким буде здійснюватись авторизація
- Пароль користувача
- Роль користувача
- Підрозділ користувача, який вказати за потреби.

Звіти

Щоб сформувати загальний звіт, потрібно натиснути відповідний пункт навігації, що відкриє сторінку зі звітом (Рис. 3.24).

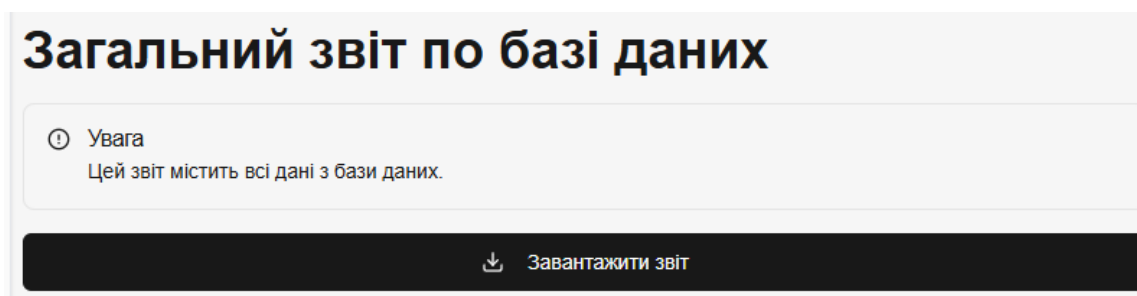


Рисунок 3.24 - Сторінка звіту по базі даних

На даний момент цей звіт експортує абсолютно усі збережені поля в форматі Excel, що дозволяє вручну налаштувати його під потреби, прибрати лишні поля або додати поля для обчислення в програмі Excel. Приклад звіту можна побачити на рисунку 3.25.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	saved_data	created_a	updated_a	data	user_id	username	email	role	subdivisic	subdivisic	subdivisic	subdivisic	form_sub	form_deg	form_dir	pub_vstup	passport_vr	address_vstup	ipn_vstup	phone_vs	pb_legal	passport	address_l	ipn_legal	phone_le	educati
2	9	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		4	Навчальн	Олег ШЕР	доктор фі 4	1	10	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	денна
3	10	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		2	Навчальн	Сергій БІЛ	доктор фі 2	1	1	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	денна
4	7	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		1	Факульте	Наталія ГІ	доктор фі 1	1	19	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	денна
5	5	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		2	Навчальн	Сергій БІЛ	доктор фі 2	1	5	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	заочна
6	4	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		1	Факульте	Наталія ГІ	доктор фі 1	1	19	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	денна
7	6	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		1	Факульте	Наталія ГІ	доктор фі 1	1	19	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	денна
8	8	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		1	Факульте	Наталія ГІ	доктор фі 1	1	19	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	денна
9	11	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		1	Факульте	Наталія ГІ	доктор фі 1	1	19	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	денна
10	12	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		1	Факульте	Наталія ГІ	доктор фі 1	1	19	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	денна
11	13	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		1	Факульте	Наталія ГІ	доктор фі 1	1	19	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	денна
12	14	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		1	Факульте	Наталія ГІ	доктор фі 1	1	19	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	денна
13	15	12.05.2021	12.05.2021	(*subdivis	1	Administr	admin@e admin		1	Факульте	Наталія ГІ	доктор фі 1	1	19	Сидоренко	Микола Степанович	паспорт	Київ	123456	093124141	Сидоренко	паспорт	київ	123456685	093423415	денна

Рисунок 3.25 - Загальний звіт по базі даних

Редагування таблиць бази даних

Для редагування даних в базі даних, потрібно перейти на сторінку Редагування таблиць , натиснувши відповідний пункт в бічному меню. Ця сторінка має чотири блоки , які відповідають кожній таблиці з бази даних , такі як Навчальні підрозділи , Ступені вищої освіти , Довідка та Спеціальності та освітні програми (Рис. 3.26)

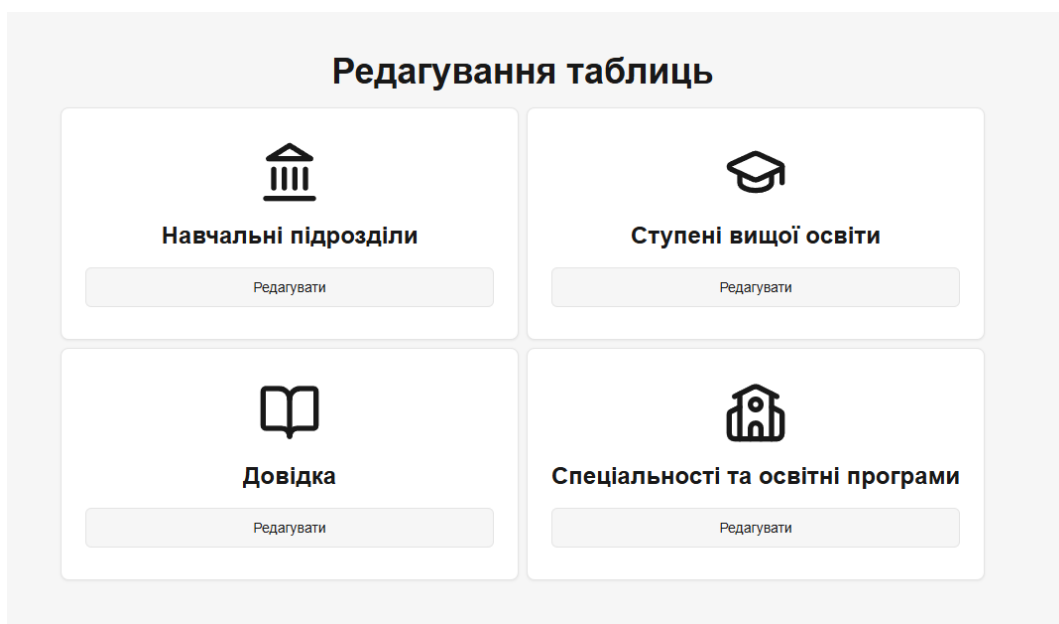


Рисунок 3.26 - Сторінка редагування таблиць

При відкритті будь-якої з цих сторінок , відкриється відповідна сторінка для редагування з таблицею , яка містить усі поля таблиці і їх можна редагувати прямо в таблиці, а також кнопками для додавання нових записів в ці таблиці. Для редагування текстових записів потрібно два рази натиснути на потрібну клітинку таблиці , що перетворить її на поле для введення. Усі поля , які посилаються на інші таблиці, тобто зовнішні ключі , мають свої випадваючі списки , для вибору значень.

Сторінку управління навчальними підрозділами можна побачити на рисунку 3.27.

Таблиця підрозділів			
+ Додати підрозділ			
☐	Назва	Директор / Декан	Ступінь
☐	Факультет біотехнології та екологічного контролю	Наталія ГРЕГРЧАК	доктор філософії Видалити
☐	Навчально-науковий інженерно-технічний інститут імені акад. І.С. Гулого	Сергій БЛАЖЕНКО	доктор філософії Видалити
☐	Навчально-науковий інститут харчових технологій	Оксана КОЧУБЕЙ-ЛИТВИНЕНКО	доктор філософії Видалити
☐	Навчально-науковий інститут економіки і управління	Олег ШЕРЕМЕТ	доктор філософії Видалити
☐	Факультет автоматизації і комп'ютерних систем	Андрій ФОРСЮК	доктор філософії Видалити
☐	Факультет готельно-ресторанного та туристичного бізнесу імені проф. В.Ф. Доценка	Віта ЦИРУЛЬНІКОВА	доктор філософії Видалити
0 з 6 рядків обрано. Видалити			
Рядків на сторінці: 8			
Сторінка 1 з 1			
Увага Після видалення записів, пов'язані рядки в інших таблицях стануть пустими, що може призвести до помилок.			

Рисунок 3.27 - Управління навчальними підрозділами

Сторінку управління ступенями вищої освіти можна побачити на рисунку 3.28.

Таблиця ступеней вищої освіти	
+ Додати ступінь	
☐	Назва
☐	доктор філософії Видалити
☐	бакалавр Видалити
☐	магістр Видалити
0 з 3 рядків обрано. Видалити	
Рядків на сторінці: 8	
Сторінка 1 з 1	
Увага Після видалення записів, пов'язані рядки в інших таблицях стануть пустими, що може призвести до помилок.	

Рисунок 3.28 - Управління ступенями освіти

Сторінку управління спеціальностями можна побачити на рисунку 3.29.

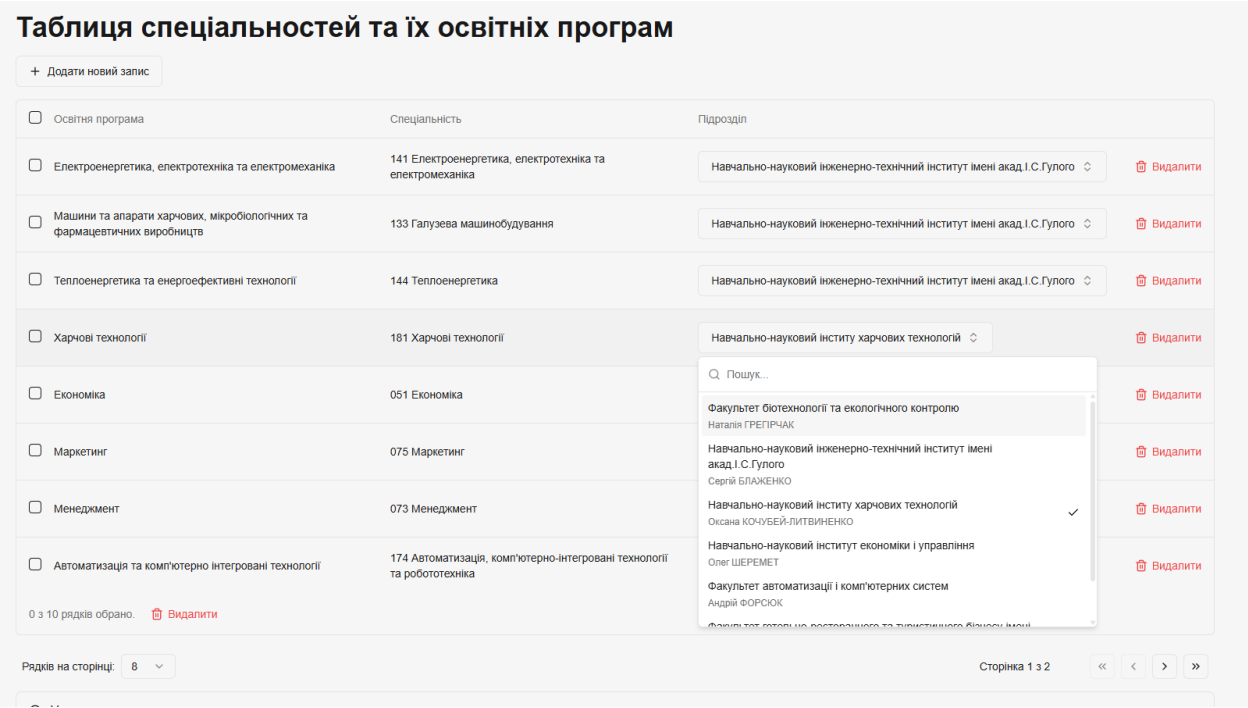


Рисунок 3.29 - Управління спеціальностями

Сторінку управління довідковою інформацією можна побачити на рисунку 3.30.

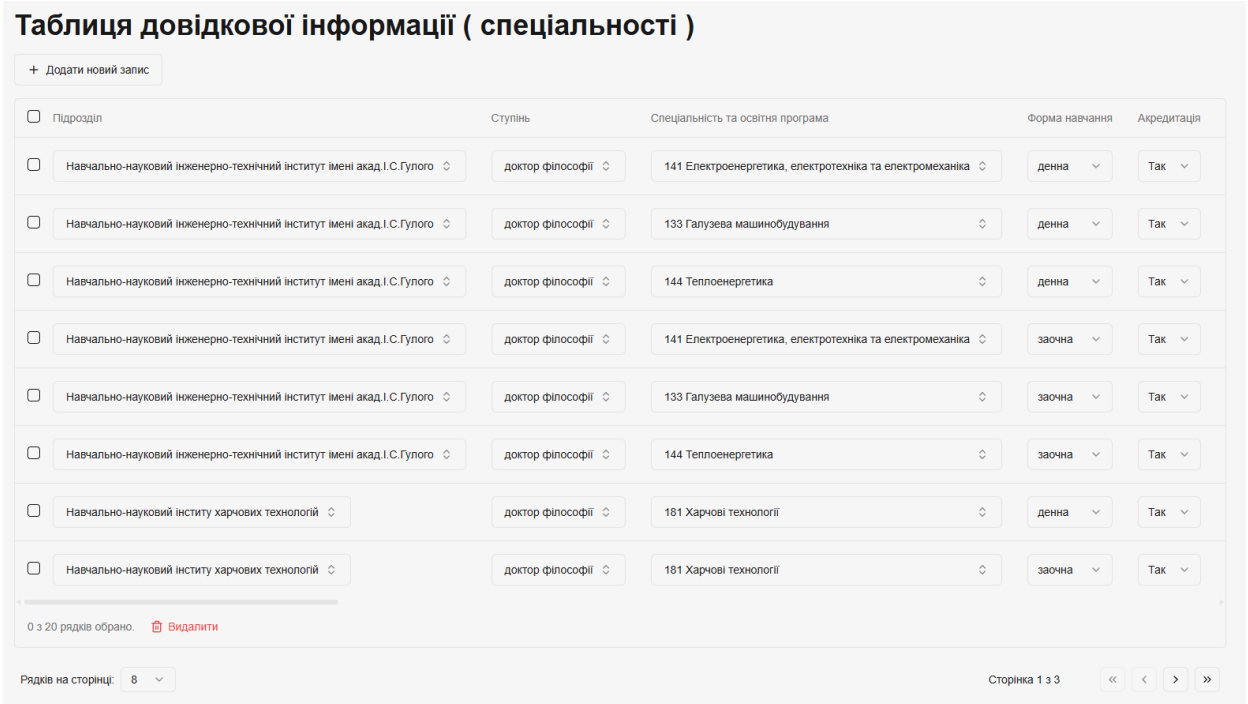


Рисунок 3.30 - Управління довідковою інформацією

Особливістю редагування цієї таблиці є те, що різні типи колонок мають різні способи їх редагування, наприклад для полів з логічними значеннями, такі як акредитованість спеціальності, показується випадаючий список з вибором Так або Ні. Для полів з зовнішніми ключами є свої випадаючі списки з пошуком. Для полів з датою, є кнопка для відкриття календаря, де можна обрати потрібну дату (Рис. 3.31).

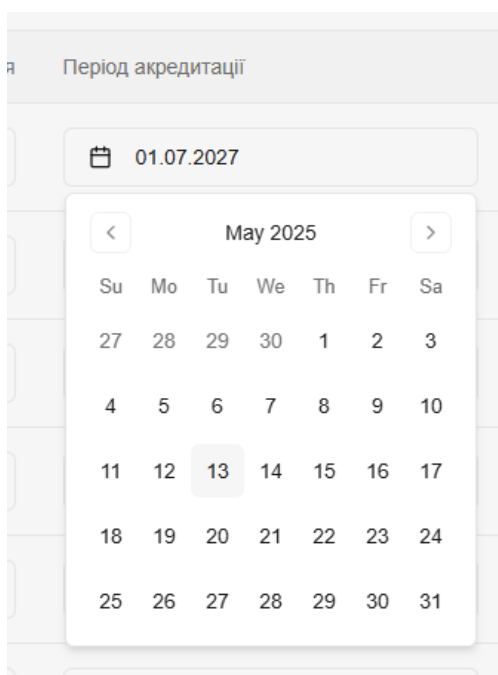


Рисунок 3.31 - Поле з типом дати

Також особливою є форма додавання нового запису в довідку, адже ця таблиця має набагато більше полів, ніж інші таблиці, тому був створений набір кнопок для зміни кількості колонок з полями для введення та випадаючих списків, що наведено на рисунку 3.32.

← Назад

Новий запис в довідці

Введіть усі потрібні довідкові дані для створення нового запису довідки

Підрозділ Оберіть підрозділ...	Ступінь Оберіть ступінь...	Форма навчання Оберіть форму навчання	Спеціальність та освітня програма Оберіть спеціальність...
Акредитація Ні	Обсяг освітньої програми 0	Період навчання	Вартість всього курсу 0
Вартість всього курсу (прописом)	Перший рік (період)	Вартість першого року 0	Вартість першого року (прописом)
Другий рік (період)	Вартість другого року 0	Вартість другого року (прописом)	Третій рік (період)
Вартість третього року 0	Вартість третього року (прописом)	Четвертий рік (період)	Вартість четвертого року 0
Вартість четвертого року (прописом)	Термін оплати першого року Оберіть дату	Щорічний термін оплати	Семестровий термін оплати

Створити новий запис

Рисунок 3.32 - Форма створення нового запису в довідці з 4 колонками

Управління шаблонами контрактів

Також адміністратор може керувати шаблонами контрактів, які використовуються при генерації документів (при їх завантаженні користувачем). Для цього потрібно відкрити відповідний пункт в навігації адміністраторської панелі, а саме пункт Контракти, що відкриє сторінку редагування шаблонів (Рис. 3.33).

Управління контрактами

Для зміни заготовок для контрактів використовуйте Word файли з потрібним форматуванням. Завантажте поточний шаблон, внесіть необхідні зміни, а потім завантажте оновлений файл назад.

Увага
При завантаженні нового шаблону буде **замінено існуючий шаблон**.
Переконайтеся, що ви зробили резервну копію поточного шаблону перед завантаженням нового.

Шаблон навчального контракту	Шаблон контракту юридичної особи	Шаблон контракту фізичної особи
Завантажте новий шаблон навчального контракту або отримайте існуючий	Завантажте новий шаблон контракту для юридичної особи або отримайте існуючий	Завантажте новий шаблон контракту для фізичної особи або отримайте існуючий
Вибірть файл або перетягніть сюди Формат DOCX (.docx) Завантажити поточний Замінити шаблон	Вибірть файл або перетягніть сюди Формат DOCX (.docx) Завантажити поточний Замінити шаблон	Вибірть файл або перетягніть сюди Формат DOCX (.docx) Завантажити поточний Замінити шаблон

Рисунок 3.33 - Сторінка управління шаблонами

Усі шаблони – це файли Word , з розширенням docx, де потрібні поля заміняються справжніми даними на момент генерації файлу. На цій сторінці можна завантажити поточний шаблон , або вивантажити новий , обравши його в меню вибору файлу , або перетягнувши в відповідну область застосунку, після чого потрібно натиснути кнопку “Замінити шаблон”.

Усі поля в шаблоні підставляються , використовуючи спеціальний синтаксис , а саме назва змінної в фігурних дужках , наприклад {year} , що при формуванні контракту буде замінено на поточний рік. Приклад шаблону контракту можна побачити на рисунку 3.34.

ДОГОВІР №

про навчання у Національному університеті харчових технологій

м. Київ « _____ » {year} р.

Національний університет харчових технологій державної форми власності в особі ректора Шевченка Олександра Юхимовича, що діє на підставі Статуту (далі – заклад), з однієї сторони, та вступник _____ {pib vstup} (прізвище, ім'я та по батькові) (далі – вступник) та/або законний представник* _____ {pib legal} (прізвище, ім'я та по батькові) (далі – законний представник), з іншої сторони (далі – сторони), уклали договір про нижченаведене

Загальні питання

- Предметом договору є навчання у закладі вищої освіти.
- Заклад бере на себе зобов'язання:
 - зарахувати вступника на навчання на _____ 1 _____ курс після виконання вимог умов прийому на навчання для здобуття вищої освіти, правил прийому та укладення договору;
 - здійснити підготовку вступника за {education form} формою здобуття освіти за освітньою програмою {education program} _____ (назва освітньої програми)

за спеціальністю {education specialty} _____ (код та назва спеціальності)

для здобуття ступеня вищої освіти {degree} _____ (ступінь вищої освіти)

- забезпечити створення належних умов для дотримання правил і норм охорони праці, техніки безпеки і виробничої санітарії відповідно до законодавства.

- Освітня програма є {accreditation} _____ (акредитованою або неакредитованою) строком до {accreditation term} _____ (строк акредитації)
- Обсяг освітньої програми – {credits} _____ кредитів Європейської кредитної трансферно-накопичувальної системи.

Рисунок 3.34 - Приклад шаблону контракту

Усі змінні, доступні для використання в шаблоні можна побачити в таблиці 3.2, а також вони описані в документації застосунку.

Таблиця 3.2 Опис змінних які використовуються в шаблонах документів

Змінна	Призначення
<i>year</i>	Поточний рік.
<i>pib_vstup</i>	Повне ПІБ вступника.
<i>passport_vstup</i>	Паспортні дані вступника.
<i>address_vstup</i>	Адреса проживання вступника.
<i>ipn_vstup</i>	ПІН (ідентифікаційний код) вступника.
<i>phone_vstup</i>	Номер телефону вступника.
<i>pib_legal</i>	Повне ПІБ законного представника (якщо є).
<i>passport_legal</i>	Паспортні дані законного представника (якщо є).
<i>address_legal</i>	Адреса проживання законного представника (якщо є).
<i>ipn_legal</i>	ПІН (ідентифікаційний код) законного представника (якщо є).
<i>phone_legal</i>	Номер телефону законного представника (якщо є).
<i>education_form</i>	Форма навчання в давальному відмінку ("денною", "заочною").
<i>education_program</i>	Назва освітньої програми.
<i>education_specialty</i>	Назва спеціальності.
<i>degree</i>	Назва ступеня вищої освіти.
<i>accreditation</i>	Статус акредитації освітньої програми ("акредитована" або "не акредитована").
<i>accreditation_term</i>	Термін дії акредитації освітньої програми (відформатована дата).
<i>credits</i>	Кількість кредитів ЄКТС освітньої програми.
<i>financing</i>	Тип фінансування навчання ("державного бюджету" або "коштів фізичних та/або юридичних осіб").
<i>pib_vstup_short</i>	Скорочене ПІБ вступника у форматі "Ім'я ПРІЗВИЩЕ КАПСОМ".
<i>pib_legal_short</i>	Скорочене ПІБ законного представника (якщо є) у форматі "Ім'я ПРІЗВИЩЕ КАПСОМ".
<i>paidEntire</i>	Загальна вартість навчання за весь строк цифрами (відформатована для української локалізації).
<i>paidEntireW</i>	Загальна вартість навчання за весь строк прописом.
<i>y1Paid</i>	Вартість навчання за перший рік цифрами (відформатована для української локалізації).
<i>y1PaidW</i>	Вартість навчання за перший рік прописом.
<i>yy1</i>	Період (навчальний рік) для першого року оплати.

Змінна	Призначення
<i>y2Paid</i>	Вартість навчання за другий рік цифрами (відформатована для української локалізації).
<i>y2PaidW</i>	Вартість навчання за другий рік прописом.
<i>yy2</i>	Період (навчальний рік) для другого року оплати.
<i>y3Paid</i>	Вартість навчання за третій рік цифрами (відформатована для української локалізації). Може бути порожньою.
<i>y3PaidW</i>	Вартість навчання за третій рік прописом. Може бути порожньою.
<i>yy3</i>	Період (навчальний рік) для третього року оплати. Може бути порожнім.
<i>y4Paid</i>	Вартість навчання за четвертий рік цифрами (відформатована для української локалізації). Може бути порожньою.
<i>y4PaidW</i>	Вартість навчання за четвертий рік прописом. Може бути порожньою.
<i>yy4</i>	Період (навчальний рік) для четвертого року оплати. Може бути порожнім.
<i>y1DuePay</i>	Дата терміну оплати за перший рік у відформатованому вигляді (наприклад, "1 вересня 2024 р.").
<i>anualPayDue</i>	Термін оплати при щорічній формі оплати. Використовується, якщо <code>payment_term</code> = "щороку".
<i>semesterPayDue</i>	Термін оплати при посеместровій формі оплати. Використовується, якщо <code>payment_term</code> = "кожного семестру".
<i>dean</i>	ПІБ декана факультету / директора навчально-наукового інституту.
<i>jur_name</i>	Назва юридичної особи (якщо замовник - юридична особа).
<i>jur_pib</i>	ПІБ представника юридичної особи (якщо замовник - юридична особа).
<i>jur_status</i>	Посада представника юридичної особи (якщо замовник - юридична особа).
<i>jur_pib_short</i>	Скорочене ПІБ представника юридичної особи (якщо замовник - юридична особа) у форматі "Ім'я ПРІЗВИЩЕ КАПСОМ".
<i>jur_req</i>	Юридичні реквізити юридичної особи (якщо замовник - юридична особа).

Імпорт та експорт даних

Також адміністратор має можливість експортувати дані з таблиць БД в спеціальний шаблон Excel, який можна також імпортувати в базу даних. Для цього

потрібно перейти до сторінки “Імпорт даних” в навігації, що відкриє сторінку імпортування даних (Рис. 3.35).

Імпортування даних

Для імпорту даних використовуйте шаблон Excel файлу, який можна завантажити нижче. В шаблоні вже заповнені поточні дані з бази даних та налаштовані випадаючі списки для вибору значень. Після заповнення шаблону, виберіть файл для завантаження та натисніть кнопку "Імпортувати".

Увага
Під час імпорту даних буде **видалено всі існуючі дані** в імпортованих таблицях бази даних. Переконайтеся, що ви зробили резервну копію бази даних перед імпортом, щоб уникнути втрати даних при помилках.

Завантажити шаблон для імпорту

Імпорт Excel файлу
Завантажте файл для імпорту даних. Перед імпортом переконайтеся, що дані були заповнені вірно, без помилок.

Вибрати файл Файл не вибрано

Імпортувати

Рисунок 3.35 - Сторінка імпортування даних

Для завантаження експортованих даних, потрібно натиснути кнопку “Завантажити шаблон для імпорту”. Це завантажить експортовані дані в спеціально створеному шаблоні для імпорту, що допоможе зручно робити масове редагування бази даних. Також в цьому шаблоні відразу налаштовані колонки, які мають зв’язок з іншими таблицями, щоб їх можна було обирати в випадаючих списках, вбудованих в програмне забезпечення Excel. Вигляд одного з листів цього шаблону можна побачити на рисунку 3.36.

	A	B	C	D	E	F	G	H	I
1	ID	Навчальний Підрозділ	id підрозділа	Ступінь	id ступеня	Спеціальність	id спеціальності	Форма навчання	Акредит
2	1	Факультет біотехнології та біоінженерії	1	доктор філософії	1	162 Біотехнології та біоінженерія	10	денна	Так
3	2	Факультет біотехнології та біоінженерії	1	доктор філософії	1	162 Біотехнології та біоінженерія	10	заочна	Так
4	3	Навчально-науковий інженерний факультет	2	доктор філософії	1	141 Електроенергетика, електротехніка та електромеханіка	1	денна	Так
5	4	Навчально-науковий інженерний факультет	2	доктор філософії	1	133 Галузева машинобудування	2	денна	Так
6	5	Навчально-науковий інженерний факультет	2	доктор філософії	1	144 Теплоенергетика	3	денна	Так
7	6	Навчально-науковий інженерний факультет	2	доктор філософії	1	181 Харчові технології	1	заочна	Так
8	7	Навчально-науковий інженерний факультет	2	доктор філософії	1	051 Економіка	2	заочна	Так
9	8	Навчально-науковий інженерний факультет	2	доктор філософії	1	075 Маркетинг	3	заочна	Так
10	9	Навчально-науковий інженерний факультет	3	доктор філософії	1	174 Автоматизація, комп'ютерно-інтегровані технології та інформатика	4	денна	Так
11	10	Навчально-науковий інженерний факультет	3	доктор філософії	1	181 Харчові технології	4	заочна	Так
12	11	Навчально-науковий інженерний факультет	4	доктор філософії	1	051 Економіка	5	денна	Ні
13	12	Навчально-науковий інженерний факультет	4	доктор філософії	1	075 Маркетинг	6	денна	Ні
14	13	Навчально-науковий інженерний факультет	4	доктор філософії	1	073 Менеджмент	7	денна	Ні
15	14	Навчально-науковий інженерний факультет	4	доктор філософії	1	051 Економіка	5	заочна	Ні
16	15	Навчально-науковий інженерний факультет	4	доктор філософії	1	075 Маркетинг	6	заочна	Ні
17	16	Навчально-науковий інженерний факультет	4	доктор філософії	1	073 Менеджмент	7	заочна	Ні
18	17	Факультет автоматизації та інформатики	5	доктор філософії	1	174 Автоматизація, комп'ютерно-інтегровані технології та інформатика	8	денна	Так
19	18	Факультет автоматизації та інформатики	5	доктор філософії	1	122 Комп'ютерні науки	9	денна	Ні
20	19	Факультет автоматизації та інформатики	5	доктор філософії	1	174 Автоматизація, комп'ютерно-інтегровані технології та інформатика	8	заочна	Так
21	20	Факультет автоматизації та інформатики	5	доктор філософії	1	122 Комп'ютерні науки	9	заочна	Ні
22									

Рисунок 3.36 - Лист редагування довідкової інформації

Для імпорту даних достатньо завантажити цей файл у відповідне поле для вибору файлу , та натиснути кнопку “Імпортувати”, що очистить таблиці бази даних, та впише в них нові значення з цього файлу.

3.4.3 Інструкція з розгортання системи

Вимоги до системи

Для розгортання системи потрібно:

- Docker і Docker Compose
- Доступ до інтернету для завантаження контейнерів
- Мінімум 2 ГБ оперативної пам'яті
- Мінімум 10 ГБ вільного місця на диску

Розгортання за допомогою Docker

Для швидкого і надійного розгортання системи за допомогою контейнеризації, потрібно виконати такі кроки:

1. Завантажте файли проекту на сервер
2. Відкрийте термінал і перейдіть до каталогу з файлом docker-compose.yml (Рис 3.37)
3. Запустіть систему за допомогою команди:
docker-compose up -d

4. Дочекайтеся завершення розгортання. Система буде доступна за адресою localhost:3000.

Цей файл запускає відразу три контейнера , один для застосунку , один для бази даних mysql, і ще один для системи управління базою даних – phpMyAdmin.

Але якщо потрібно , цю систему можна запускати модульно , адже вона підтримує зовнішню базу даних , тобто можна запустити лише один контейнер з застосунком , вказавши правильні налаштування змінних середовища [18].

```
version: '3.8'
services:
  app:
    build:
      context: .
      dockerfile: Dockerfile
    container_name: app
    environment:
      - MYSQL_HOST=mysql # Хост MySQL (потрібно використовувати 172.17.0.1,
                          # якщо використовується локальний MySQL)
      - MYSQL_PORT=3306 # Порт MySQL
      - MYSQL_DATABASE=db # Назва бази даних
      - MYSQL_USER=admin # Користувач MySQL
      - MYSQL_PASSWORD=admin # Пароль MySQL
      - AUTH_SECRET=OqW/n3/i/e65k0xpc6juz73UsqidP0hh5q0eV/kQptQ # Секрет для JWT
    volumes:
      - templates_data:/app/src/files # Зберігання шаблонів на локальному диску,
                                     # щоб не втратити їх при перезапуску контейнера
  mysql:
    image: mysql:latest
    container_name: mysql
    environment:
      MYSQL_ROOT_PASSWORD: admin # Пароль root
      MYSQL_DATABASE: db # Назва бази даних
      MYSQL_USER: admin # Користувач MySQL
      MYSQL_PASSWORD: admin # Пароль MySQL
    volumes:
      - mysql_data:/var/lib/mysql # Зберігання даних MySQL на локальному диску
    ports:
      - "3306:3306"
  phpMyAdmin:
    image: phpmyadmin/phpmyadmin
    container_name: phpmyadmin
    environment:
      PMA_HOST: mysql # Хост MySQL
      PMA_PORT: 3306 # Порт MySQL
    ports:
      - "8080:80"
    depends_on:
      - mysql
volumes:
  mysql_data:
    driver: local
  templates_data:
    driver: local
```

Рисунок 3.37 - Вміст конфігурації docker-compose.yml

Налаштування системи

Для налаштування системи потрібно редагувати змінні середовища в файлі `docker-compose.yml`:

- **MYSQL_HOST**: адреса хоста MySQL (за замовчуванням `mysql`)
- **MYSQL_PORT**: порт MySQL (за замовчуванням `3306`)
- **MYSQL_DATABASE**: назва бази даних (за замовчуванням `db`)
- **MYSQL_USER**: ім'я користувача MySQL (за замовчуванням `admin`)
- **MYSQL_PASSWORD**: пароль користувача MySQL (за замовчуванням `admin`)
- **AUTH_SECRET**: секретний ключ для JWT автентифікації

Загалом система при запуску її розгортання, збирає образ для майбутнього контейнера застосунку, що дозволяє використовувати його пізніше без оригінальних файлів та коду застосунку. Також, так як докер-контейнери по своїй природі імутабельні, тобто не змінні, то кожен його перезапуск починається приводить застосунок до його початкового стану на момент збирання образу. Використовуючи це, можна налаштувати автоматичний перезапуск контейнера, у разі виникнення неочікуваних помилок, що призведе до того, що застосунок запуситься і буде працювати стабільно надалі. Також, через це потрібно налаштовувати спеціальні змінні для контейнера, а саме підключати `volume` (Томи). Наприклад, це потрібно для директорії з файлами шаблонів, щоб при перезапуску контейнера, редаговані файли не змінювались, адже при перезапуску контейнера він приводиться в початковий стан, в тому числі файли. Але томи дозволяють зберігати деяку частину застосунку в локальній файловій системі.

Використання сучасних технологій (Next.js, React, MySQL) забезпечує швидку та стабільну роботу системи під високим навантаженням.

Розроблений програмний продукт повністю відповідає поставленим вимогам і готовий до впровадження у реальному середовищі НУХТ.

3.5 Тестування програмного продукту

3.5.1 Тест-план

Для перевірки працездатності застосунку було створено тест-план, що включає базові рівні тестування. Метою тестування є:

1. Перевірка відповідності функціонування системи згідно заданим вимогам
2. Виявлення помилок у програмному коді
3. Перевірка роботи програми в різних браузерах, з різними рушіями
4. Перевірка безпеки системи (зокрема, авторизації та доступів до різних функцій застосунку) [23]

Рівні тестування включають:

1. Модульне тестування - перевірка окремих компонентів, API функцій [22], фронтенд частини застосунку
2. Інтеграційне тестування - перевірка взаємодії між компонентами, збереження стану застосунку
3. Системне тестування - перевірка всього застосунку
4. Крос-браузерне тестування - перевірка в різних браузерах

3.5.2 Модульне тестування

На етапі модульного тестування було перевірено кожний основний компонент програми окремо:

1. Компоненти авторизації:
 - Вхід з правильним та неправильним логіном/паролем
 - Доступ до захищених сторінок без авторизації
 - Перевірка форм введення даних
2. Адміністративна панель:
 - Створення та видалення користувачів
 - Експорт та імпорт даних
 - Редагування таблиць та створення нових записів

3. Робота з документами:

- Завантаження документів
- Збереження , редагування записів вступників.

Для кожного тесту було визначено пріоритет (високий, середній, низький) та очікуваний результат, що дозволило систематизувати процес тестування.

3.5.3 Інтеграційне тестування

Після перевірки окремих компонентів було проведено тестування їх взаємодії:

1. Взаємодія клієнт-сервер:

- Передача даних через API
- Обробка помилок від сервера

2. Робота з базою даних:

- Коректність запитів до бази даних
- Збереження та оновлення інформації

3.5.4 Системне тестування

На етапі системного тестування було перевірено функціонування всієї системи в цілому , а саме:

1. Час завантаження сторінок [25]
2. Робота системи при одночасному доступі кількох користувачів
3. Перевірка захисту від несанкціонованого доступу
4. Захист від простих SQL-ін'єкцій

3.5.5 Крос-браузерне тестування

Було проведено тестування в різних браузерах, які базуються на різних рушіях, результат якого можна побачити в таблиці 3.3.

Таблиця 3.3 Результат тестування кросбраузерності застосунку

Браузер	Версії	Результат
Google Chrome	115-121	Passed
Mozilla Firefox	110-117	Passed
Safari	14-16	Passed

3.5.6 Детальні функціональні тести

В таблиці 3.4 наведено приклади конкретних тестів, що було проведено для перевірки ключових функцій системи.

Таблиця 3.4 Результати тестування окремих функцій застосунку

№	Пріоритет	Перевірка	Очікуваний результат	Google Chrome	Mozilla Firefox	Safari
1	високий	Авторизація користувача	Успішний вхід до системи	Passed	Passed	Passed
2	високий	Створення нового документа в відповідній формі	Документ створюється і зберігається	Passed	Passed	Passed
3	середній	Експорт даних в Excel	Коректне збереження файлу	Passed	Passed	Passed
4	Середній	Імпорт даних з файлу Excel	Коректне імпортування усіх даних	Passed	Passed	Passed
5	середній	Редагування профілю користувача	Зміни зберігаються в системі	Passed	Passed	Passed

№	Пріоритет	Перевірка	Очікуваний результат	Google Chrome	Mozilla Firefox	Safari
6	низький	Відображення даних з порожніми полями	Система коректно обробляє пусті значення	Passed	Passed	Passed
7	середній	Перевірка пагінації головної таблиці контрактів при великій кількості записів	Коректне розбиття на сторінки	Passed	Passed	Passed

3.5.7 Висновки за результатами тестування

На підставі проведеного тестування можна зробити такі висновки:

1. Програмний продукт стабільно функціонує в усіх основних браузерях.
2. Основні функції (авторизація, робота з документами, управління користувачами) відповідають заданим вимогам.
3. Система забезпечує задовільний рівень безпеки для захисту збережених даних та обмеження несанкціонованого доступу.

Під час тестування було виявлено кілька незначних помилок, які було усунуто в процесі доопрацювання. В цілому, розроблена система відповідає початковим вимогам і може бути впровадженою для використання.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розроблено інформаційну систему формування контрактів для вступників НУХТ, яка повністю відповідає поставленим завданням і вимогам.

В процесі розробки системи було вирішено усі покладені завдання. Було проаналізовано існуючий процес формування контрактів у НУХТ та виявлено основні проблеми, зокрема відсутність автоматизації, значні часові витрати на створення документів та висока ймовірність помилок через ручне заповнення, відсутність централізованого сховища документів, спроектовано архітектуру системи, яка базується на сучасному стеку веб-технологій (Next.js, React, TailwindCSS, MySQL), що забезпечує швидкодію, надійність та зручність використання. Також було розроблено усі потрібні функції системи- авторизація користувачів з розмежуванням прав доступу, зручна багатокрокова форма для створення контрактів, завантаження договорів у форматі DOCX, адміністративна панель застосунку з широкими можливостями управління системою, детальна документація прямо в застосунку, а також розроблений механізм швидкого та надійного розгортання за допомогою контейнеризації усіх компонентів системи.

Розроблена система дозволяє значно спростити та прискорити процес формування контрактів для вступників НУХТ шляхом автоматизації рутинних операцій. Основні переваги системи є скорочення часу на створення контрактів у 2 рази, зменшення ймовірності помилок та пропущених даних завдяки валідації форми створення договорів, можливість зберігання, редагування та повторного використання заповнених даних, наявність адміністративного інтерфейсу для легкого управління даними та шаблонами документів.

Впровадження розробленої системи дозволить підвищити ефективність роботи приймальної комісії НУХТ. Зокрема, система дозволить обробляти більшу кількість запитів на формування контрактів за той самий час, що особливо важливо в період активної вступної кампанії.

Перспективами розвитку цієї системи можуть стати:

- інтеграція з існуючими інформаційними системами НУХТ;
- додавання функцій електронного підпису документів;
- розширення можливостей аналітики та звітності;
- імпортування даних вступників з ЄДЕБО;

Розроблена система готова до впровадження у реальне середовище НУХТ і може бути адаптована для використання в інших навчальних закладах з подібними потребами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008:2015. ЗВІТИ У СФЕРІ НАУКИ І ТЕХНІКИ. На заміну ДСТУ 3008-95 ; чинний від 22.06.2015. Київ : ДП «УкрНДНЦ», 2016. 26 с. URL: https://science.kname.edu.ua/images/dok/derzhstandart_3008_2015.pdf (дата звернення: 14.05.2025).
2. ДСТУ 3973-2000. СИСТЕМА РОЗРОБЛЕННЯ ТА ПОСТАВЛЕННЯ ПРОДУКЦІЇ НА ВИРОБНИЦТВО. Чинний від 27.11.2000. Київ : ДЕРЖСТАНДАРТ УКРАЇНИ, 2001. 43 с. URL: <https://www.pdau.edu.ua/sites/default/files/node/2805/dstu39732000.pdf> (дата звернення: 14.05.2025).
3. ДСТУ Б В.2.5-82:2016. ЕЛЕКТРОБЕЗПЕКА В БУДІВЛЯХ І СПОРУДАХ. Вимоги до захисних заходів від ураження електричним струмом. На заміну ДБН В.2.5-27-2006 ; чинний від 01.04.2017. Київ : ДП "УкрНДНЦ", 2016. 110 с. URL: <http://www.tsatu.edu.ua/ettp/wp-content/uploads/sites/25/dstu-b-v.2.5-82-2016-elektrobezpeka-v-budivljah-i-sporudah-1.pdf> (дата звернення: 14.05.2025).
4. ДСТУ 2293:2014. ОХОРОНА ПРАЦІ. Терміни та визначення основних понять. На заміну ДСТУ 2293-99 ; чинний від 01.05.2015. Київ : МІНЕКОНОМРОЗВИТКУ УКРАЇНИ, 2015. 18 с.
5. ДСТУ ISO 6309:2007. ПРОТИПОЖЕЖНИЙ ЗАХИСТ. Знаки безпеки Форма та колір. На заміну ГОСТ 12.4.026–76 ; чинний від 30.03.2007. Київ : ДЕРЖСПОЖИВСТАНДАРТ УКРАЇНИ, 2008. 12 с. URL: https://www.ksv.biz.ua/GOST/DSTY_ALL/DSTY1/dsty_iso_6309-2007.pdf (дата звернення: 14.05.2025).
6. ДСТУ EN 13306:2019. ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ. Терміни та визначення понять. Чинний від 01.10.2007. Київ : ДЕРЖСПОЖИВСТАНДАРТ УКРАЇНИ, 2008. 34 с.
7. ДСТУ ISO/IEC 27001:2015. МЕТОДИ ЗАХИСТУ СИСТЕМИ УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ. На заміну ДСТУ ISO/IEC 27001:2010 ; чинний від 01.01.2017. Київ : ДП «УкрНДНЦ», 2016. 28 с. URL:

https://www.assistem.kiev.ua/doc/dstu_ISO-IEC_27001_2015.pdf (дата звернення: 14.05.2025).

8. ДСТУ 8604:2015. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги. На заміну ГОСТ 12.2.032-78 ; чинний від 01.07.2017. Київ : ДП «УкрНДНЦ», 2017.

9. ДСТУ EN 12464-1:2016. Світло та освітлення. Освітлення робочих місць. Частина 1. Внутрішні робочі місця.. Чинний від 01.12.2017. Київ : ДП «УкрНДНЦ», 2018. 47 с.

10. ДБН В.2.5-28-2006. ПРИРОДНЕ І ШТУЧНЕ ОСВІТЛЕННЯ. На заміну СНиП II-4-79 ; чинний від 15.05.2006. Київ : Мінбуд України, 2006. 96 с.

11. ДБН В.2.2-9:2018. ГРОМАДСЬКІ БУДИНКИ ТА СПОРУДИ ОСНОВНІ ПОЛОЖЕННЯ. На заміну ДБН В.2.2-9-2009 ; чинний від 01.09.2019. Київ : Мінрегіон України, 2022. 49 с.

12. ECMAScript 2022 Language Specification. *ECMAScript*. 01.01.2022. URL: <https://262.ecma-international.org/13.0/> (дата звернення: 14.05.2025).

13. *HTML Living Standard*. 14.05.2025. URL: <https://html.spec.whatwg.org/> (дата звернення: 14.05.2025).

14. CSS Specifications Level 3. W3C. 22.12.2020. URL: <https://www.w3.org/TR/css-2020/> (дата звернення: 14.05.2025).

15. Web Content Accessibility Guidelines (WCAG). W3C. 06.05.2025. URL: <https://www.w3.org/TR/WCAG21/> (дата звернення: 14.05.2025).

16. Next.js Documentation. *Next.js*. URL: <https://nextjs.org/docs> (дата звернення: 14.05.2025).

17. MySQL 8.0 Reference Manual. *MySQL*. URL: <https://dev.mysql.com/doc/refman/8.0/en/> (дата звернення: 14.05.2025).

18. Docker Documentation. *Docker*. URL: <https://docs.docker.com/> (дата звернення: 14.05.2025).

19. OWASP Top 10 – 2021: The Ten Most Critical Web Application Security Risks.. *OWASP Foundation*. URL: <https://owasp.org/Top10/> (дата звернення: 14.05.2025).

20. JavaScript Guide. *MDN Web Docs*. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide> (дата звернення: 14.05.2025).
21. React Documentation. *React*. URL: <https://react.dev/> (дата звернення: 14.05.2025).
22. *RESTful API Design Guidelines*. URL: <https://restfulapi.net/> (дата звернення: 14.05.2025).
23. Auth.js documentation. *Auth.js*. URL: <https://authjs.dev/getting-started> (дата звернення: 14.05.2025).
24. ShadCN Documentation. *shadcn/ui*. URL: <https://ui.shadcn.com/docs> (дата звернення: 14.05.2025).
25. Lighthouse Performance Tool Documentation. *Google Developers*. URL: <https://developer.chrome.com/docs/lighthouse> (дата звернення: 14.05.2025).
26. Microservices: a definition of this new architectural term. *Martin Fowler*. URL: <https://martinfowler.com/articles/microservices.html> (дата звернення: 14.05.2025).
27. The Twelve-Factor App. *Jacob Kaplan-Moss*. URL: <https://12factor.net/> (дата звернення: 14.05.2025).
28. Don Norman, Jakob Nielsen. The Definition of User Experience (UX). *NN/g*. 08.08.1998. URL: <https://www.nngroup.com/articles/definition-user-experience/> (дата звернення: 14.05.2025).
29. Roy T. F. Architectural Styles and the Design of Network-based Software Architectures : DOCTOR OF PHILOSOPHY in Information and Computer Science / UNIVERSITY OF CALIFORNIA. IRVINE, 2000. 180 с. URL: <https://ics.uci.edu/~fielding/pubs/dissertation/top.htm> (дата звернення: 14.05.2025).
30. Грибков С. В., Ліманська Н. В., Костіков М. П. МЕТОДИЧНІ РЕКОМЕНДАЦІЇ : до виконання кваліфікаційної роботи на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «комп'ютерні науки», освітньо-професійної програми «комп'ютерні науки» денної та заочної форм здобуття освіти. КИЇВ : НУХТ, 2025. 43 с.

ДОДАТКИ

Додаток А. Схеми організаційної структури НУХТ

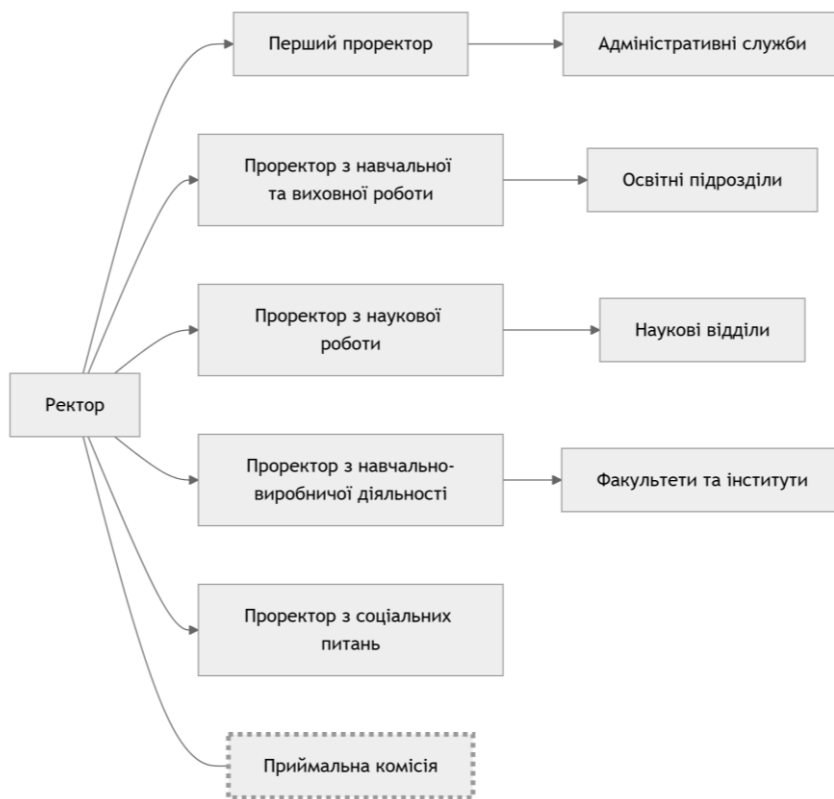


Рисунок А.1 – Схема організаційної структури університету



Рисунок А.2 – Схема структури приймальної комісії

Додаток Б. Функціональні моделі

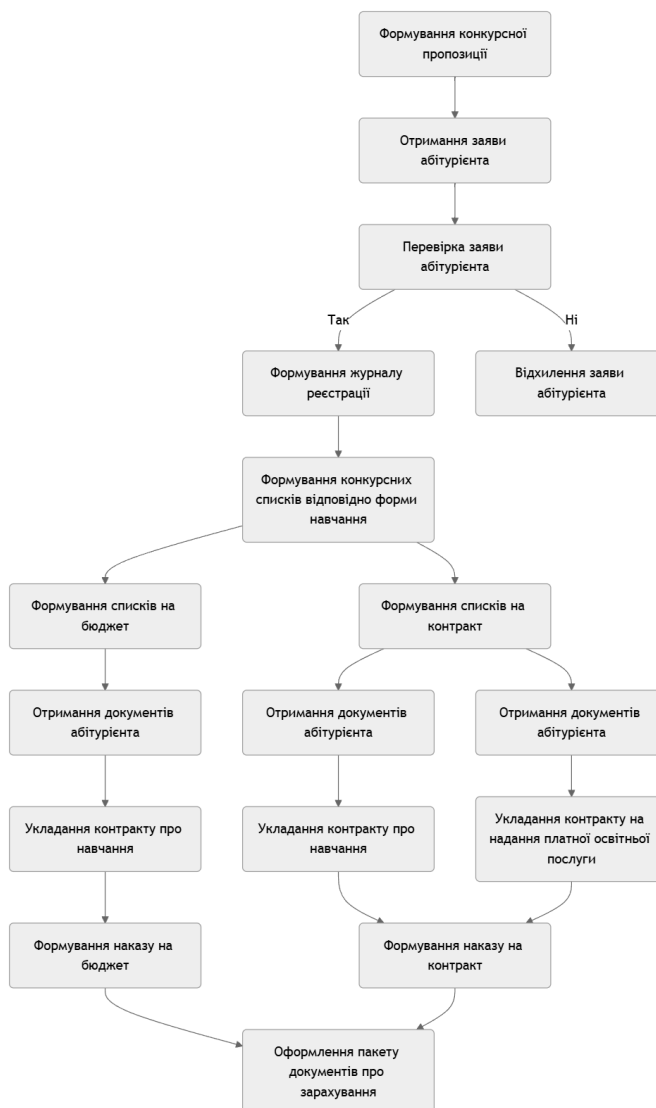


Рисунок Б.1 - Бізнес-процес по оформленню пакету документів про зарахування

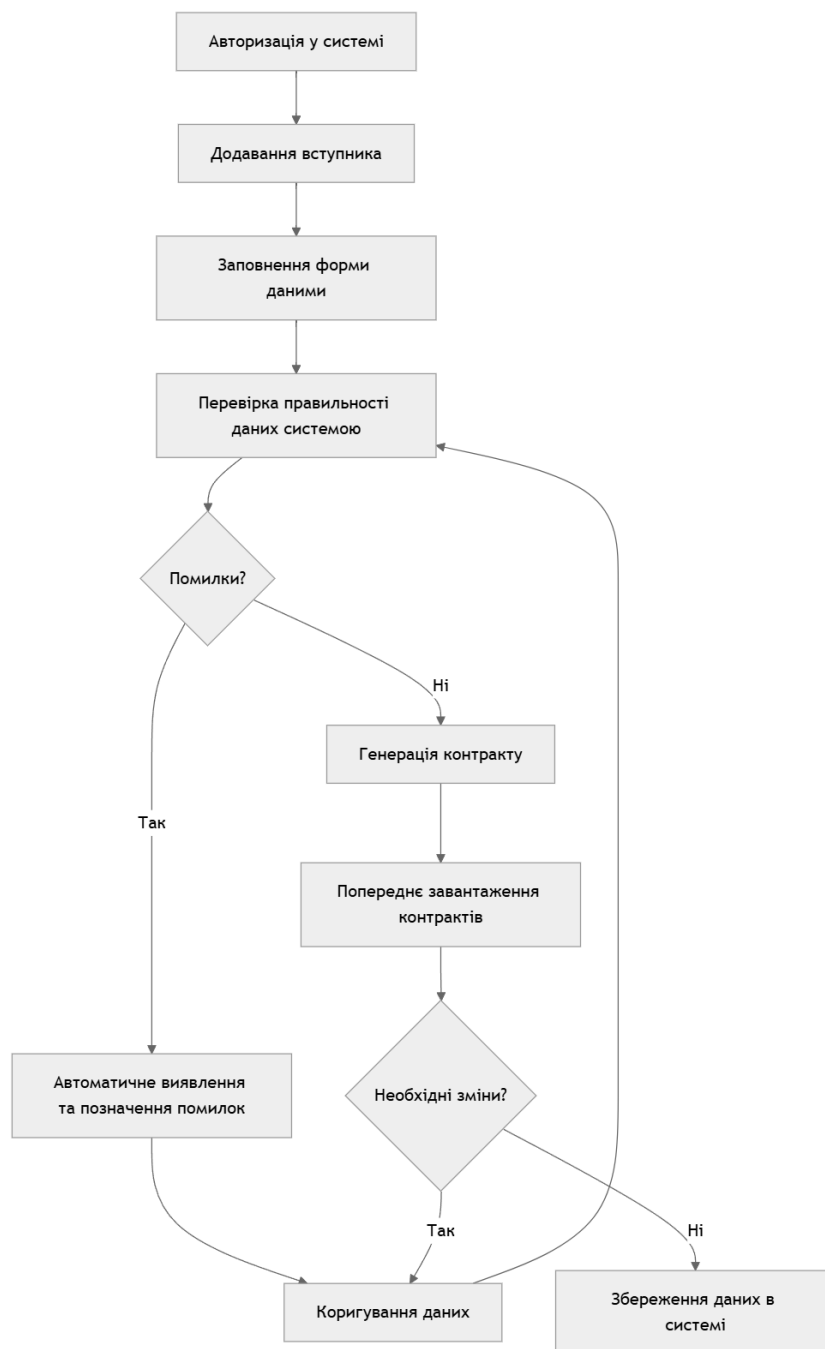


Рисунок Б.2 – Функціональна модель TO BE

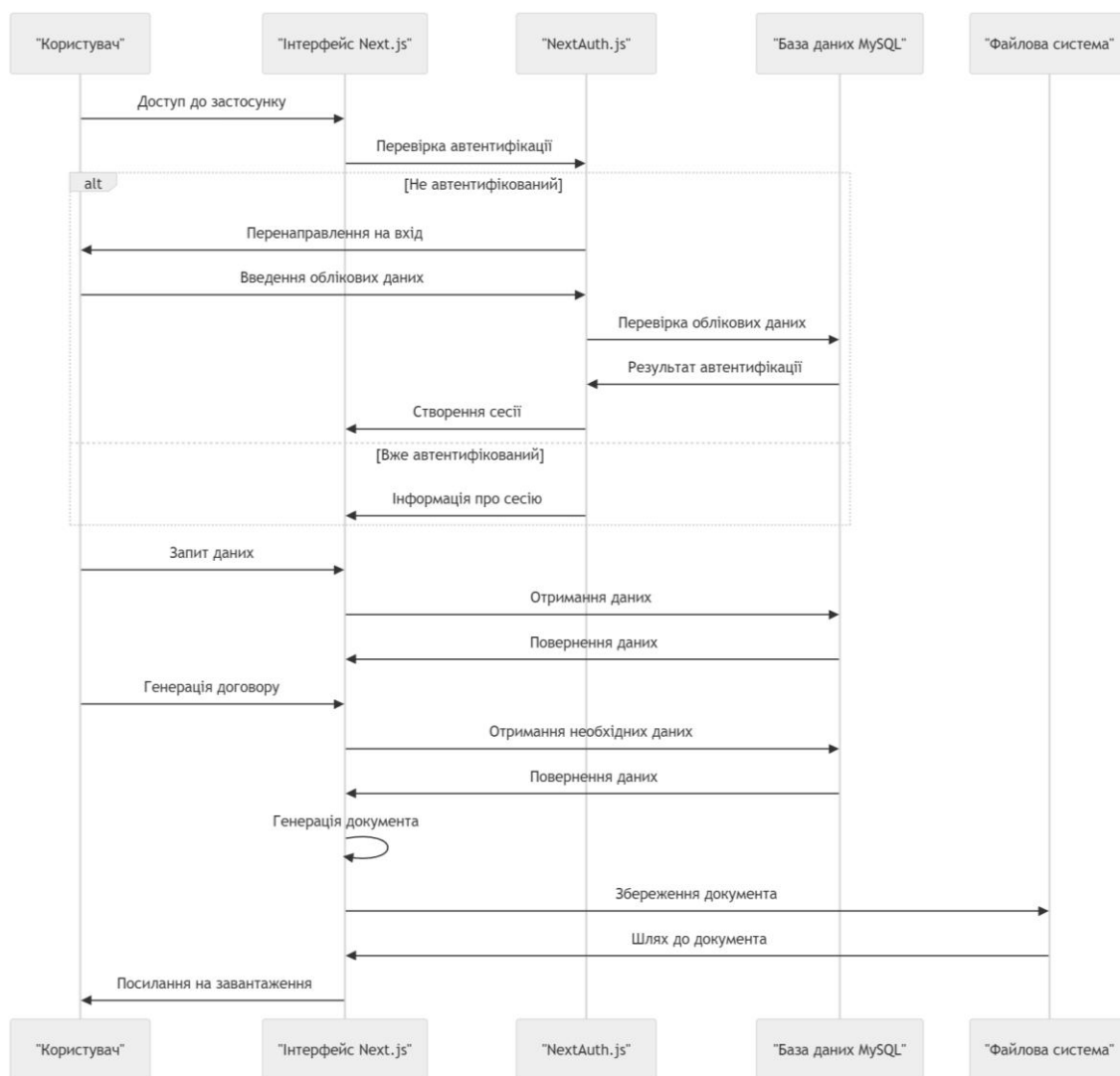


Рисунок Б.3 – Потік даних при завантаженні документів

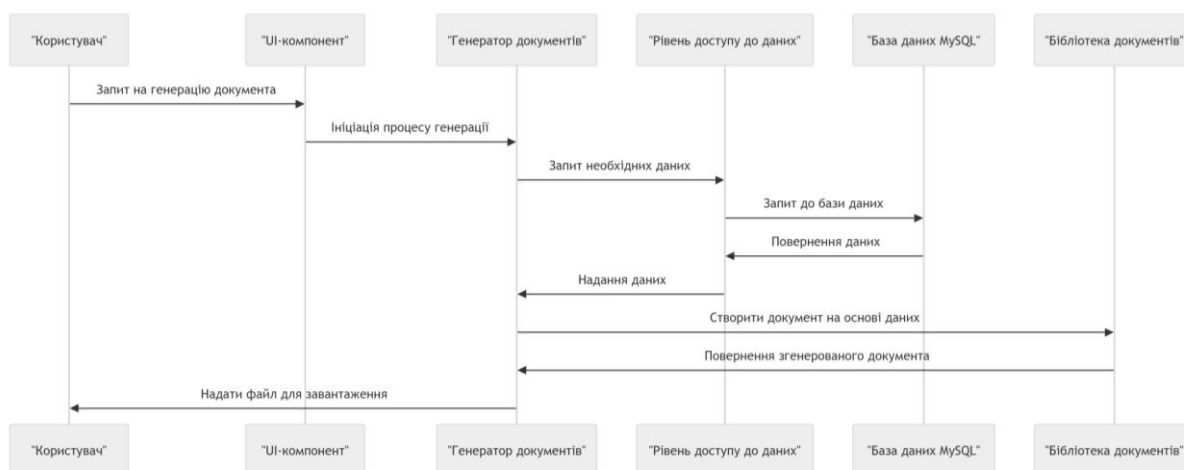


Рисунок Б.4 - Процес генерації документів.

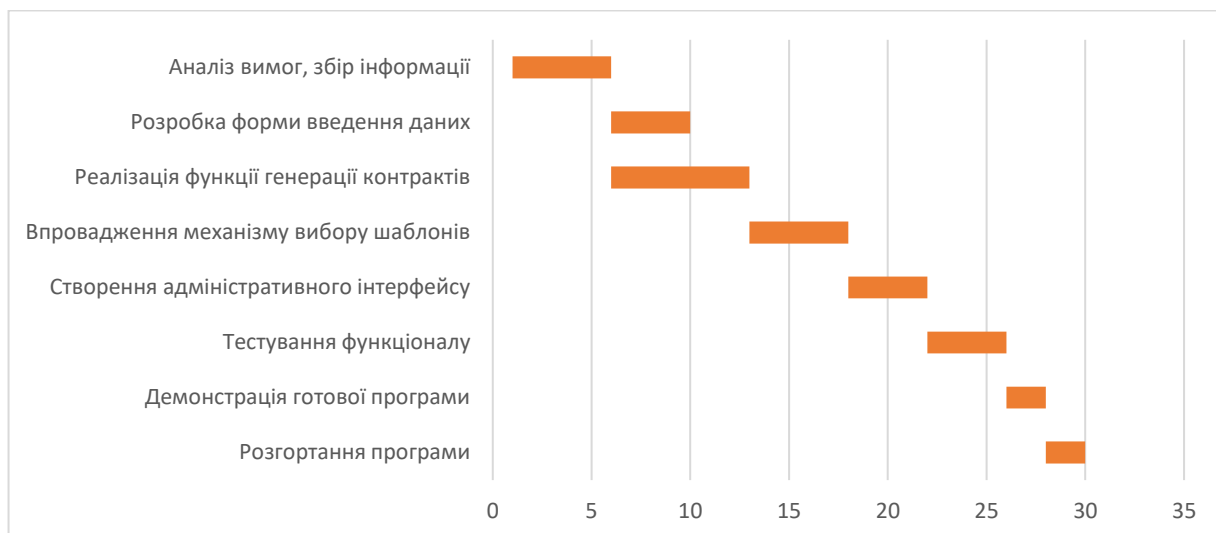


Рисунок Б.5 - Діаграма Ганта

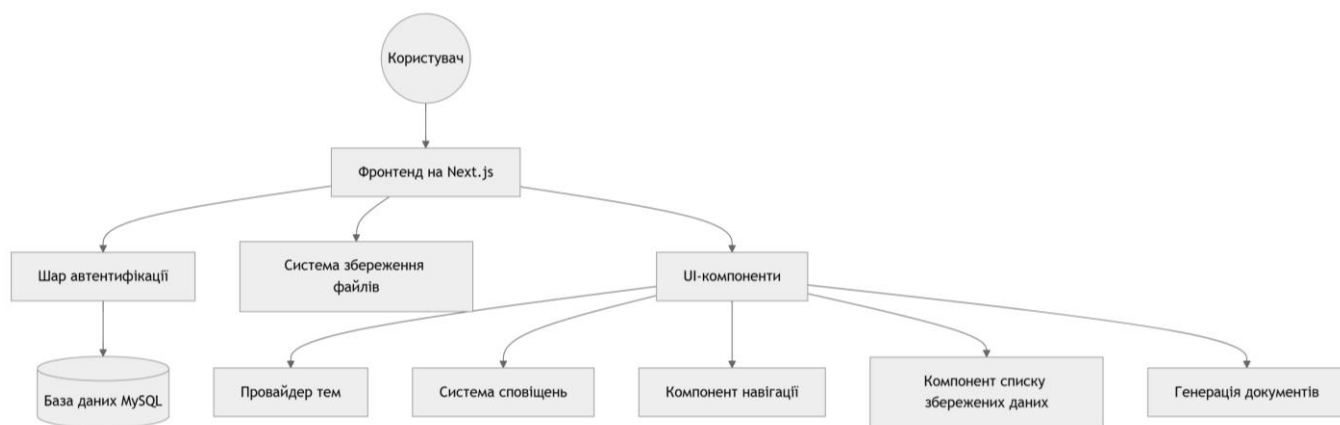


Рисунок Б.6 - Високорівнева схема архітектури застосунку

Додаток В. Моделі та схеми Бази Даних

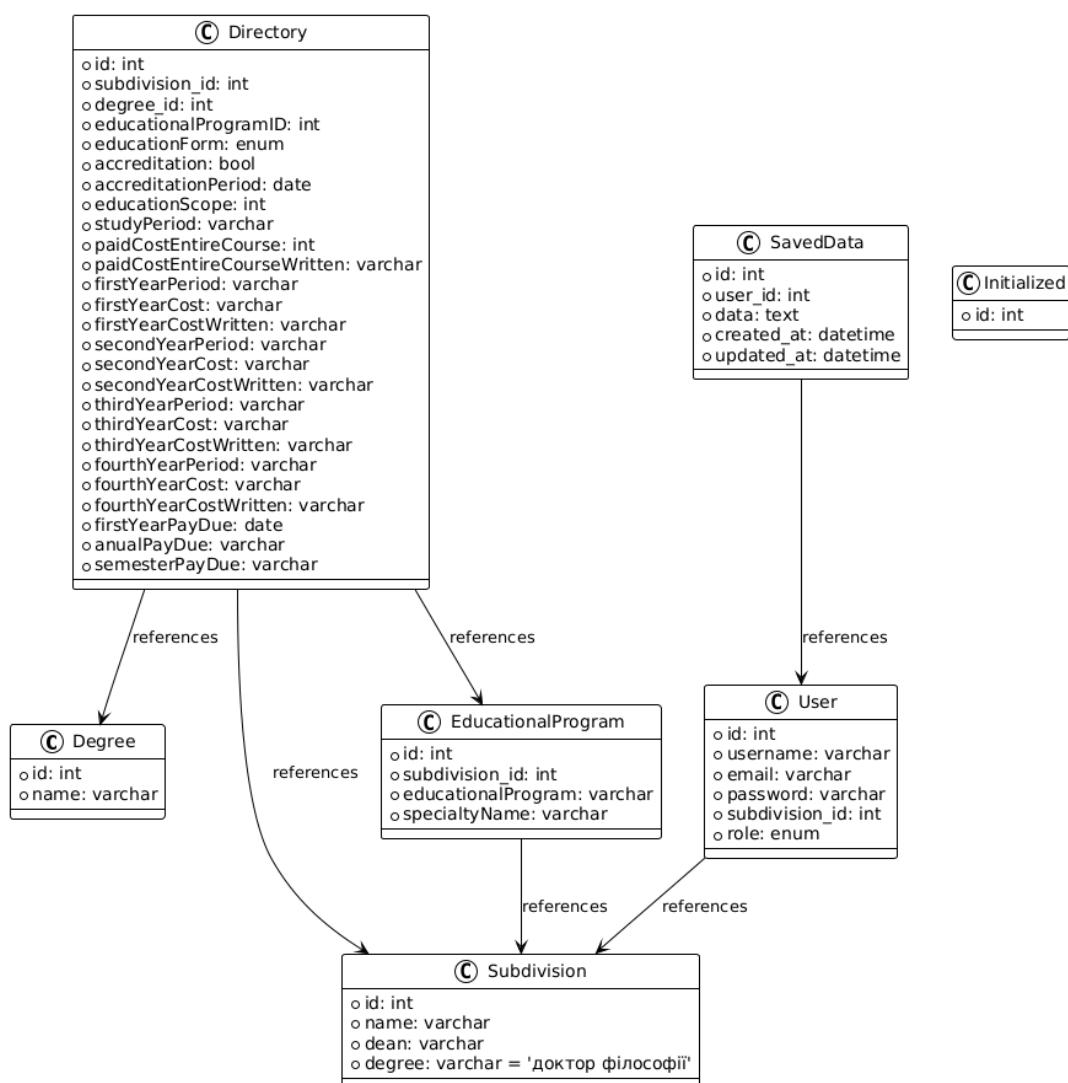


Рисунок В.1 - Схема бази даних

Додаток Г. Інтерфейс Інформаційної Системи

Вітаю

Введіть ваші дані для входу

Email

m@example.com

Пароль

Password

Увійти

Рисунок Г.1 - Форма авторизації застосунку

Привіт, Administrator

ДокументаціяАдміністрування

Пошук по імені вступника

Скинути фільтриФільтри

ПІБ вступника	ПІБ представника	Підрозділ	Спеціальність	Фінансування	Дата створення	Ступінь	Дата оновлення	
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Факультет біотехнології та екологічного контролю	162 Біотехнології та біоінженерія	Контракт фіз. особа	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Навчально-науковий інженерно-технічний інститут імені акад. І.С.Гулого	133 Галузева машинобудування	Контракт юр. особа	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Факультет біотехнології та екологічного контролю	162 Біотехнології та біоінженерія	Контракт фіз. особа	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Факультет біотехнології та екологічного контролю	162 Біотехнології та біоінженерія	Бюджет	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Факультет біотехнології та екологічного контролю	162 Біотехнології та біоінженерія	Контракт фіз. особа	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Навчально-науковий інститут економіки і управління	075 Маркетинг	Бюджет	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Навчально-науковий інженерно-технічний інститут імені акад. І.С.Гулого	141 Електроенергетика, електротехніка та електромеханіка	Контракт фіз. особа	12.05.2025	доктор філософії	12.05.2025	...
Сидоренко Микола Степанович	Сидоренко Степан Миколайович	Факультет біотехнології та екологічного контролю	162 Біотехнології та біоінженерія	Контракт фіз. особа	12.05.2025	доктор філософії	12.05.2025	...

Рядків на сторінці: 8

Сторінка 1 з 2

Рисунок Г.2 – Сторінка управління контрактами

← Назад

Крок 1. Оберіть навчальний підрозділ та фінансування

Підрозділ не обрано
Оберіть підрозділ зі списку нижче

Оберіть підрозділ...

Фінансування навчання

Оберіть фінансування

Далі

Рисунок Г.3 - Сторінка створення контракту

Адмін панель

Управління застосунком

Користувачі
Управління користувачами застосунку

Звіти
Управління звітами бази даних

БД та генерація

Редагувати таблиці
Редагування таблиць бази даних

Контракти
Управління шаблонами контрактів

Імпорт даних
Імпорт та експорт даних в базу даних

← Назад до застосунку

Управління застосунком

Адмін панель

Користувачі

Звіти

БД та генерація

Редагувати таблиці

Контракти

Імпорт даних

Рисунок Г.4 – Адміністративна панель застосунку

Імпортування даних

Для імпорту даних використовуйте шаблон Excel файлу, який можна завантажити нижче. В шаблоні вже заповнені поточні дані з бази даних та налаштовані випадаючі списки для вибору значень. Після заповнення шаблону, виберіть файл для завантаження та натисніть кнопку "Імпортувати".

Увага
Під час імпорту даних буде видалено всі існуючі дані в імпортованих таблицях бази даних.
Переконайтесь, що ви зробили резерву **копію бази даних** перед імпортом, щоб уникнути втрати даних при помилках.

Завантажити шаблон для імпорту

Імпорт Excel файлу
Завантажте файл для імпорту даних. Перед імпортом переконайтесь, що дані були заповнені вірно, без помилок.

Вибрати файл Файл не вибрано

Імпортувати

Рисунок Г.5 - Форма імпортування даних

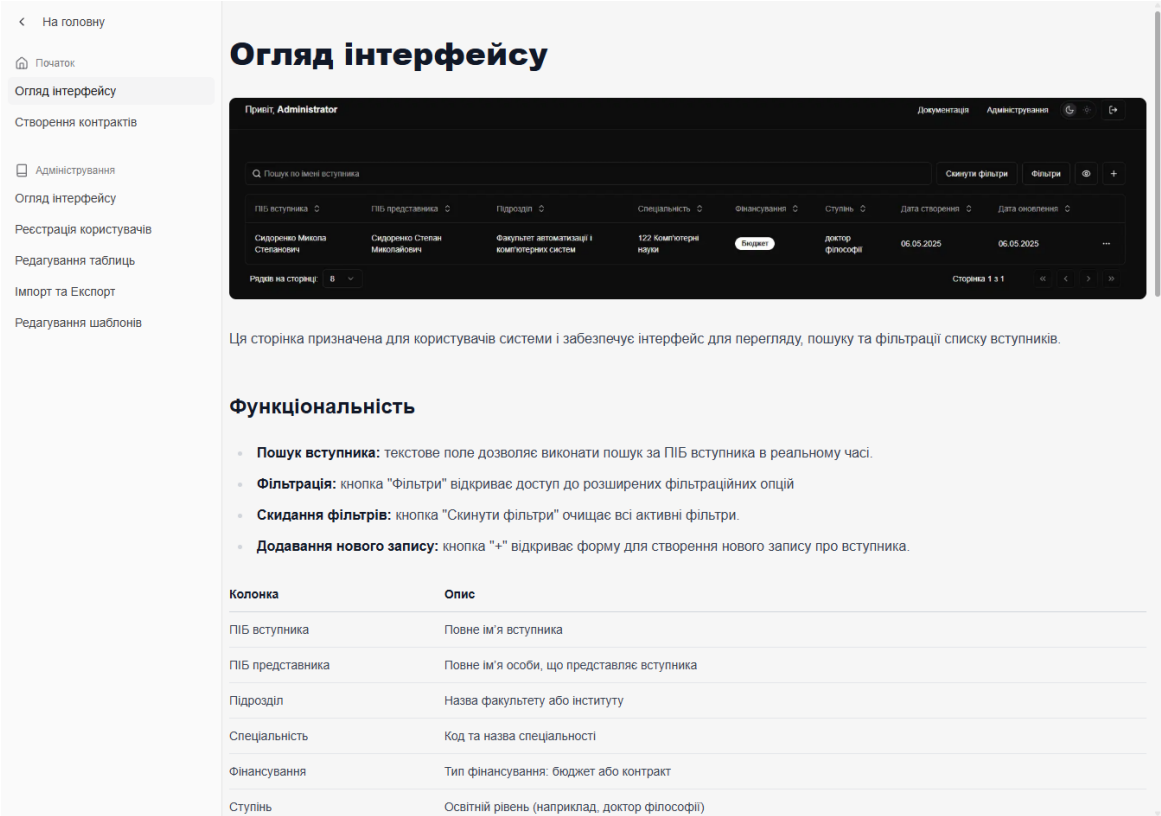


Рисунок Г.6 - Сторінка в документації застосунку

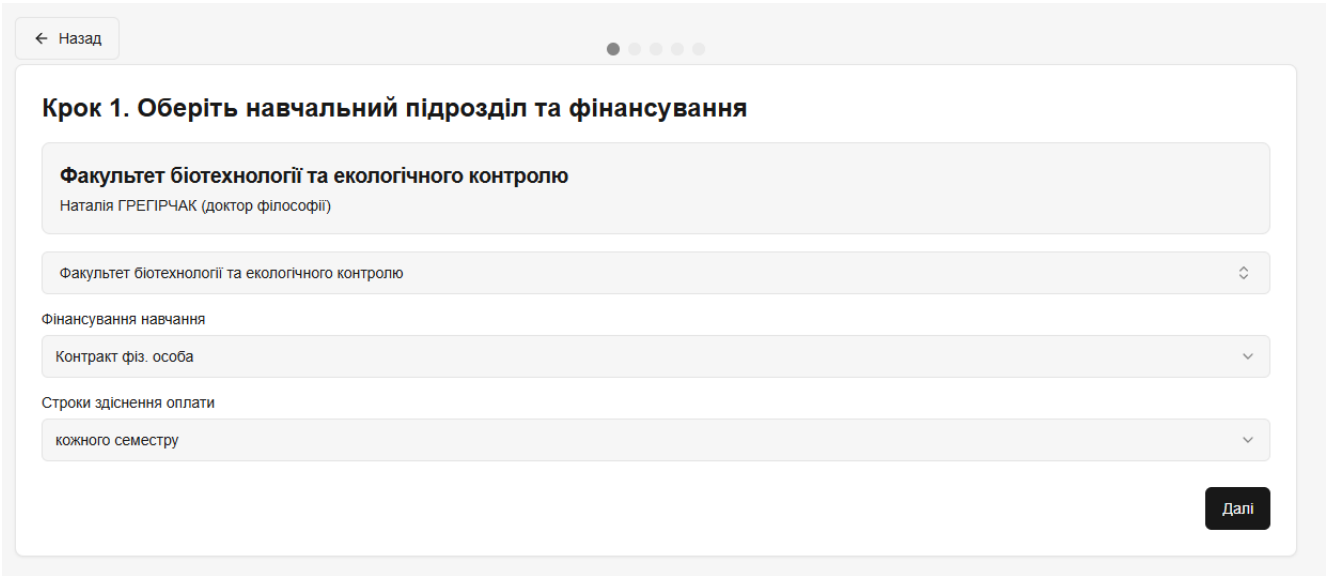


Рисунок Г.7 - Перший крок форми створення контракту

← Назад

Крок 1.5 Дані юридичної особи

Організаційно-правова форма та назва юридичної особи, яка замовляє освітню послугу

Введіть назву юридичної особи

Посада і представник замовника

Введіть посаду і представника замовника

Замовник діє на підставі

Введіть підставу, на якій діє замовник

ІП представника

Олена ГОЛОВАЧ

Реквізити замовника

Введіть реквізити замовника

Назад Далі

Рисунок Г.8 – Введення даних про юридичну особу на формі створення контракту

← Назад

Крок 2. Загальні дані вступника

ПІБ вступника

Сидоренко Микола Степанович

Серія та номер, ким виданий паспорт вступника

123456789, 4321, 01.01.2000

Місце проживання (реєстрації) вступника

Київська область, Бучанський район, с. Шпильки, вул. Шевченка, буд.57

Реєстраційний номер облікової картки платника податків вступника

1234567890

Номери телефонів

0931234567; 0978765432

Назад Далі

Рисунок Г.9 - Другий крок форми створення контракту

← Назад

Крок 3. Загальні дані законного представника

ПІБ законного представника
Сидоренко Микола Степанович

Серія та номер, ким виданий паспорт законного представника
123456789, 4321, 01.01.2000

Місце проживання (реєстрації) законного представника
Київська область, Бучанський район, с. Шпильки, вул. Шевченка, буд.57

Реєстраційний номер облікової картки платника податків законного представника
1234567890

Номери телефонів законного представника
0931234567; 0978765432

Назад Далі

Рисунок Г.10 - Третій крок форми створення
контракту

← Назад

Крок 4. Освітня програма

Ступінь вищої освіти
доктор філософії

Форма навчання
денна

Назва освітньої програми
Біотехнологія

Назад Далі

Попередній перегляд

Обрана освітня програма

Назва освітньої програми
Біотехнологія

Назва та код спеціальності
162 Біотехнології та біоінженерія

Форма навчання
денна

Загальний термін навчання
2024-2028

Вартість навчання
205 200 грн.

Акредитація
акредитована до 01.07.2027

Навчальний підрозділ
Факультет біотехнології та екологічного контролю

Рисунок Г.11 - Четвертий крок форми створення контракту

← Назад

Крок 5. Збереження та завантаження документів

Усі дані заповнені, ви можете прямо зараз завантажити архів з усіма документами, або окремо кожен документ.

Скачати архів

- Договір про навчання
- Договір юр. особа

Назад Зберегти

Рисунок Г.12 – Фінальний крок форми створення контракту

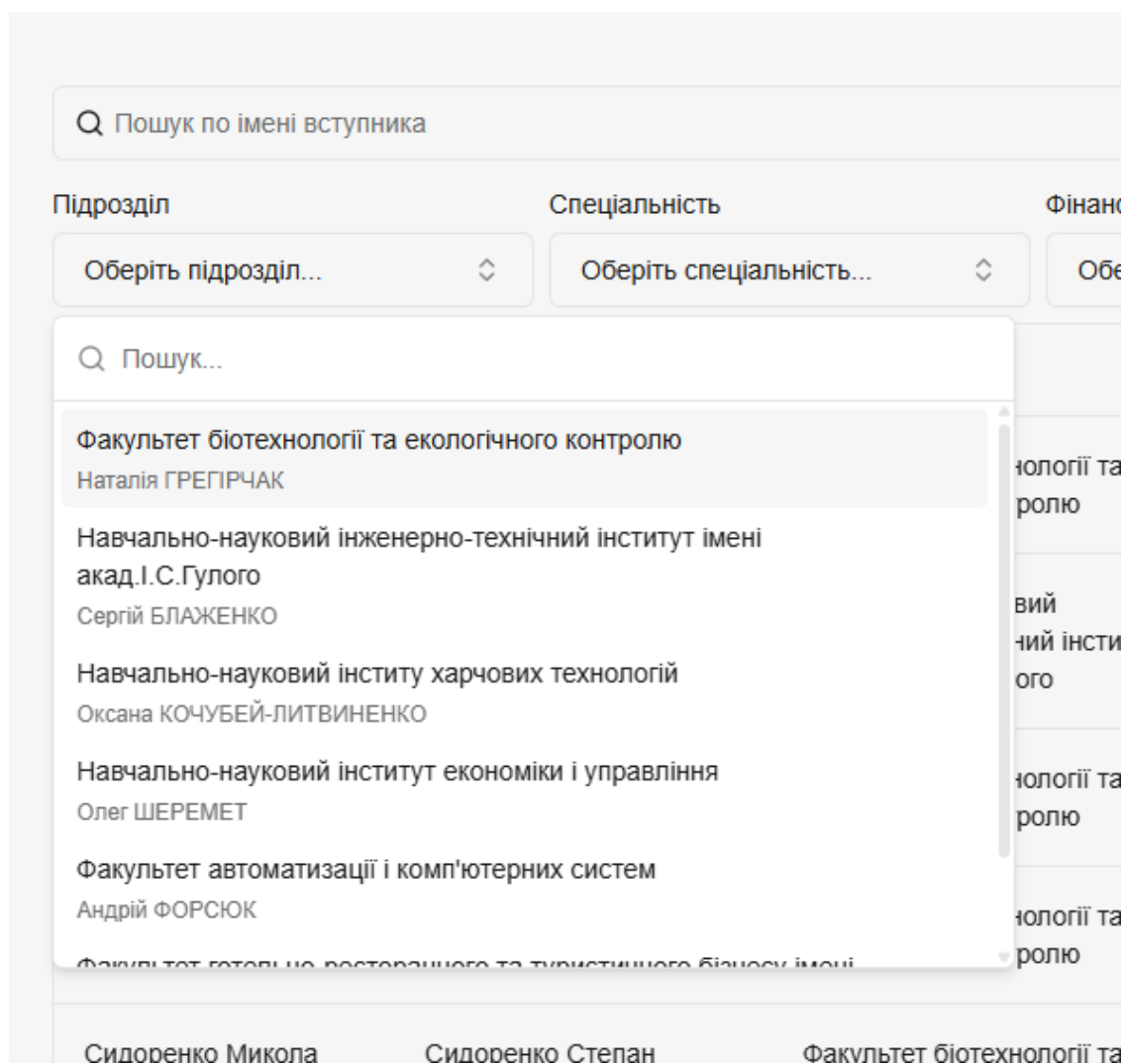


Рисунок Г.13 - Випадаючий список фільтрування за підрозділом

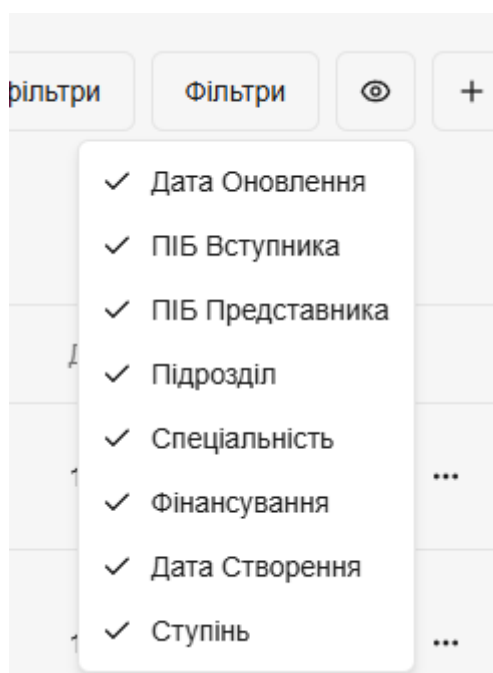


Рисунок Г.14 - Випадаючий список видимості колонок

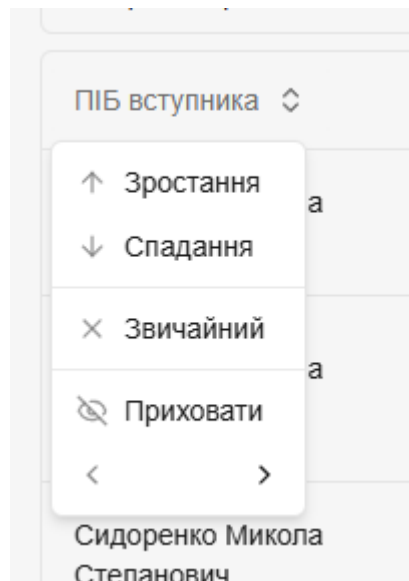


Рисунок Г.15 - Меню управління колонкою таблиці

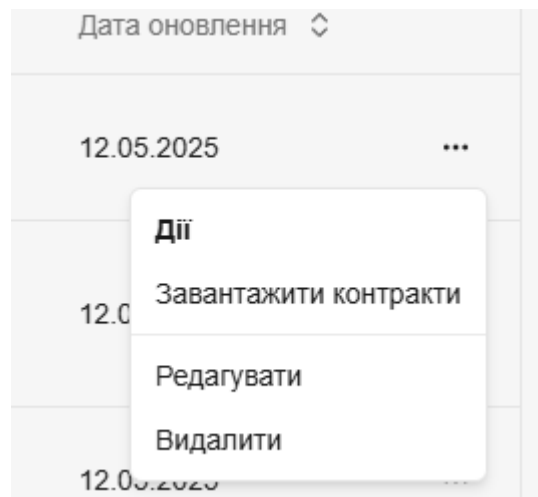


Рисунок Г.16 - Меню дій над збереженими контрактами

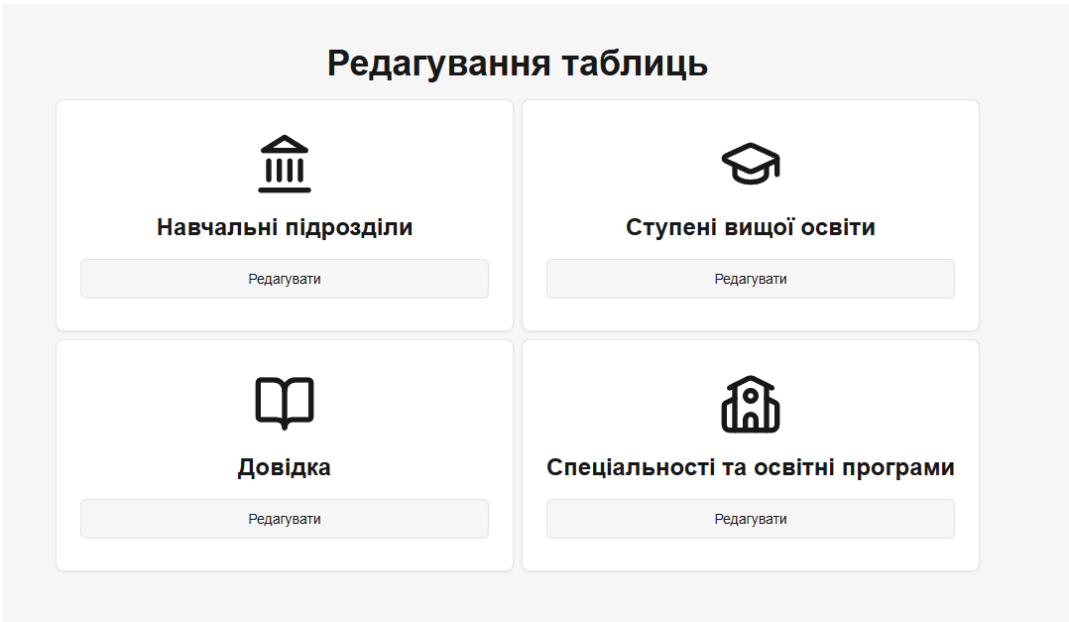


Рисунок Г.20 - Сторінка редагування таблиць

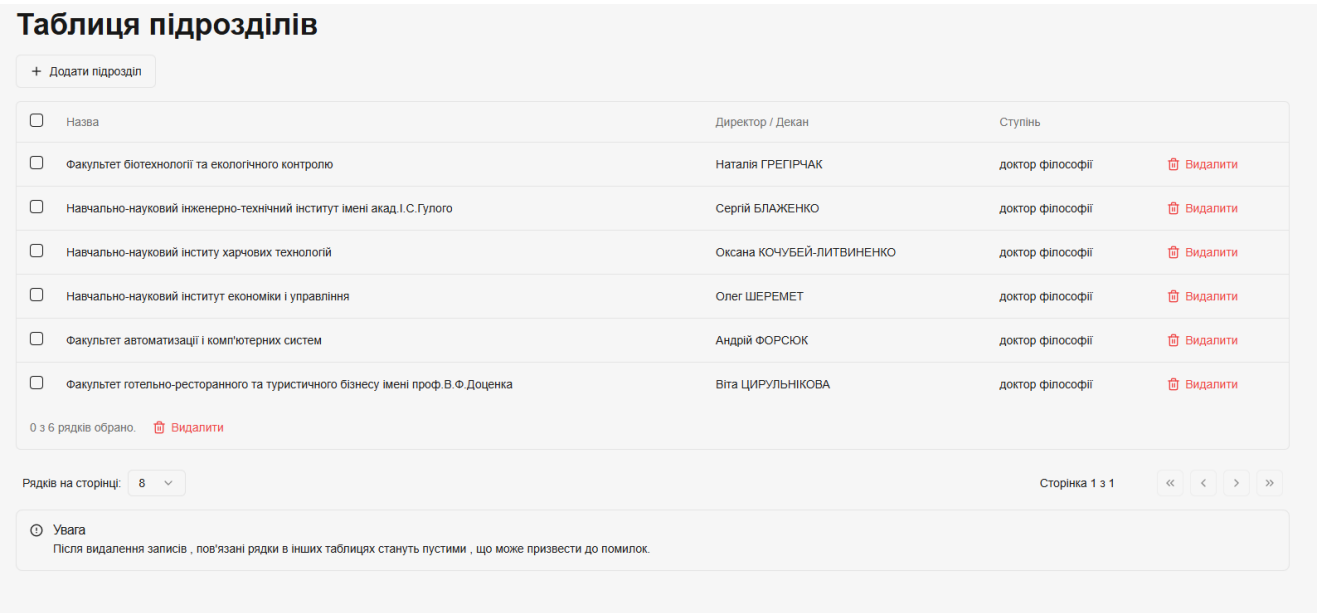


Рисунок Г.21 - Управління навчальними підрозділами

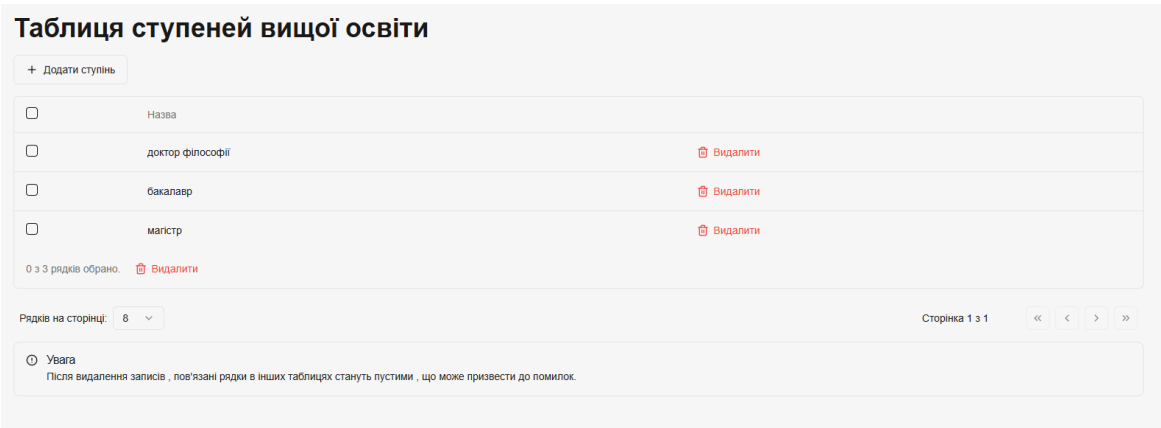


Рисунок Г.22 - Управління ступенями освіти

+ Додати новий запис			
☐ Освітня програма	Спеціальність	Підрозділ	
☐ Електроенергетика, електротехніка та електромеханіка	141 Електроенергетика, електротехніка та електромеханіка	Навчально-науковий інженерно-технічний інститут імені акад. І. С. Гулого	☐ Видалити
☐ Машини та апарати харчових, мікробіологічних та фармацевтичних виробництв	133 Галузева машинобудування	Навчально-науковий інженерно-технічний інститут імені акад. І. С. Гулого	☐ Видалити
☐ Теплоенергетика та енергоефективні технології	144 Теплоенергетика	Навчально-науковий інженерно-технічний інститут імені акад. І. С. Гулого	☐ Видалити
☐ Харчові технології	181 Харчові технології	Навчально-науковий інститут харчових технологій	☐ Видалити
☐ Економіка	051 Економіка	Пошук... - Факультет біотехнологій та екологічного контролю Наталія ГРЕПІЧАК - Навчально-науковий інженерно-технічний інститут імені акад. І. С. Гулого Сергій БЛАЖЕНКО - Навчально-науковий інститут харчових технологій Оксана КОЧУБЕЙ-ЛИТВИНЕНКО ✓ - Навчально-науковий інститут економіки і управління Олег ШЕРЕМЕТ - Факультет автоматизації і комп'ютерних систем Андрій ФОРСЮК	☐ Видалити
☐ Маркетинг	075 Маркетинг		☐ Видалити
☐ Менеджмент	073 Менеджмент		☐ Видалити
☐ Автоматизація та комп'ютерно-інтегровані технології	174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка		☐ Видалити
0 з 10 рядків обрано. ☐ Видалити			
Рядків на сторінці: 8			
Сторінка 1 з 2			

Рисунок Г.23 - Управління спеціальностями

+ Додати новий запис					
☐ Підрозділ	Ступінь	Спеціальність та освітня програма	Форма навчання	Акредитація	
☐ Навчально-науковий інженерно-технічний інститут імені акад. І. С. Гулого	доктор філософії	141 Електроенергетика, електротехніка та електромеханіка	денна	Так	✓
☐ Навчально-науковий інженерно-технічний інститут імені акад. І. С. Гулого	доктор філософії	133 Галузева машинобудування	денна	Так	✓
☐ Навчально-науковий інженерно-технічний інститут імені акад. І. С. Гулого	доктор філософії	144 Теплоенергетика	денна	Так	✓
☐ Навчально-науковий інженерно-технічний інститут імені акад. І. С. Гулого	доктор філософії	141 Електроенергетика, електротехніка та електромеханіка	заочна	Так	✓
☐ Навчально-науковий інженерно-технічний інститут імені акад. І. С. Гулого	доктор філософії	133 Галузева машинобудування	заочна	Так	✓
☐ Навчально-науковий інженерно-технічний інститут імені акад. І. С. Гулого	доктор філософії	144 Теплоенергетика	заочна	Так	✓
☐ Навчально-науковий інститут харчових технологій	доктор філософії	181 Харчові технології	денна	Так	✓
☐ Навчально-науковий інститут харчових технологій	доктор філософії	181 Харчові технології	заочна	Так	✓
0 з 20 рядків обрано. ☐ Видалити					
Рядків на сторінці: 8					
Сторінка 1 з 3					

Рисунок Г.24 - Управління довідковою інформацією

Рисунок Г.25 - Поле з типом дати

Новий запис в довідці
Введіть усі потрібні довідкові дані для створення нового запису довідки

Підрозділ	Ступінь	Форма навчання	Спеціальність та освітня програма
Оберіть підрозділ...	Оберіть ступінь...	Оберіть форму навчання	Оберіть спеціальність...
Акредитація	Обсяг освітньої програми	Період навчання	Вартість всього курсу
Ні	0		0
Вартість всього курсу (прописом)	Перший рік (період)	Вартість першого року	Вартість першого року (прописом)
		0	
Другий рік (період)	Вартість другого року	Вартість другого року (прописом)	Третій рік (період)
	0		
Вартість третього року	Вартість третього року (прописом)	Четвертий рік (період)	Вартість четвертого року
0			0
Вартість четвертого року (прописом)	Термін оплати першого року	Щорічний термін оплати	Семестровий термін оплати
	Оберіть дату		

Створити новий запис

Рисунок Г.26 - Форма створення нового запису в довідці з 4 колонками

Управління контрактами

Для зміни заготовок для контрактів використовуйте Word файли з потрібним форматуванням.
Завантажте поточний шаблон, внесіть необхідні зміни, а потім завантажте оновлений файл назад.

Увага
При завантаженні нового шаблону буде **замінено існуючий шаблон**.
Переконайтеся, що ви зробили резервну копію поточного шаблону перед завантаженням нового.

Шаблон навчального контракту

Завантажте новий шаблон навчального контракту або отримайте існуючий

Виберіть файл або перетягніть сюди
Формат DOCX (.docx)

Завантажити поточний Замінити шаблон

Шаблон контракту юридичної особи

Завантажте новий шаблон контракту для юридичної особи або отримайте існуючий

Виберіть файл або перетягніть сюди
Формат DOCX (.docx)

Завантажити поточний Замінити шаблон

Шаблон контракту фізичної особи

Завантажте новий шаблон контракту для фізичної особи або отримайте існуючий

Виберіть файл або перетягніть сюди
Формат DOCX (.docx)

Завантажити поточний Замінити шаблон

Рисунок Г.27 - Сторінка управління шаблонами

ДОГОВІР №

про навчання у Національному університеті харчових технологій

м. Київ «_____» _____ {year} р.

Національний університет харчових технологій державної форми власності в особі ректора Шевченка Олександра Юхимовича, що діє на підставі Статуту (далі – заклад), з однієї сторони, та вступник _____ {pib vstup} _____ (прізвище, ім'я та по батькові) (далі – вступник) та/або законний представник* _____ {pib legal} _____ (прізвище, ім'я та по батькові) (далі – законний представник), з іншої сторони (далі – сторони), уклали договір про нижченаведене

Загальні питання

- Предметом договору є навчання у закладі вищої освіти.
- Заклад бере на себе зобов'язання:
 - зарахувати вступника на навчання на _____ 1 _____ курс після виконання вимог умов прийому на навчання для здобуття вищої освіти, правил прийому та укладення договору;
 - здійснити підготовку вступника за _____ {education form} _____ формою здобуття освіти за освітньою програмою _____ {education program} _____ (назва освітньої програми)

за спеціальністю _____ {education specialty} _____ (код та назва спеціальності)

для здобуття _____ ступеня _____ вищої освіти _____ {degree} _____ (ступінь вищої освіти)

- забезпечити створення належних умов для дотримання правил і норм охорони праці, техніки безпеки і виробничої санітарії відповідно до законодавства.

- Освітня програма є _____ {accreditation} _____ (акредитованою або неакредитованою) строком до _____ {accreditation term} _____ (строк акредитації)
- Обсяг освітньої програми – _____ {credits} _____ кредитів Європейської кредитної трансферно-накопичувальної системи.

Рисунок Г.28 - Приклад шаблону контракту

Імпортуння даних

Для імпорту даних використовуйте шаблон Excel файлу, який можна завантажити нижче. В шаблоні вже заповнені поточні дані з бази даних та налаштовані випадаючі списки для вибору значень. Після заповнення шаблону, виберіть файл для завантаження та натисніть кнопку "Імпортувати".

ⓘ Увага

Під час імпорту даних буде **видалено всі існуючі дані** в імпортованих таблицях бази даних. Переконайтеся, що ви зробили резервну **копію бази даних** перед імпортом, щоб уникнути втрати даних при помилках.

📄 Завантажити шаблон для імпорту

Імпорт Excel файлу

Завантажте файл для імпорту даних. Перед імпортом переконайтеся, що дані були заповнені вірно, без помилок.

Вибрати файл Файл не вибрано

Імпортувати

Рисунок Г.29 - Сторінка імпортування даних

	A	B	C	D	E	F	G	H	I
1	ID	Навчальний Підрозділ	id підрозділа	Ступінь	id ступеня	Спеціальність	id спеціальності	Форма навчання	Акредит
2	1	Факультет біотехнології та	1	доктор філософії	1	162 Біотехнології та біоінженерія	10	денна	Так
3	2	Факультет біотехнології та	1	доктор філософії	1	162 Біотехнології та біоінженерія	10	заочна	Так
4	3	Навчально-науковий інже	2	доктор філософії	1	141 Електроенергетика, електротехніка та елект	1	денна	Так
5	4	Навчально-науковий інже	2	доктор філософії	1	133 Галузева машинобудування	2	денна	Так
6	5	Навчально-науковий інже	2	доктор філософії	1	144 Теплоенергетика	3	денна	Так
7	6	Навчально-науковий інже	2	доктор філософії	1	181 Харчові технології	1	заочна	Так
8	7	Навчально-науковий інже	2	доктор філософії	1	051 Економіка	2	заочна	Так
9	8	Навчально-науковий інже	2	доктор філософії	1	075 Маркетинг	3	заочна	Так
10	9	Навчально-науковий інст	3	доктор філософії	1	174 Автоматизація, комп'ютерно-інтегровані технології т	4	денна	Так
11	10	Навчально-науковий інст	3	доктор філософії	1	122 Комп'ютерні науки	4	заочна	Так
12	11	Навчально-науковий інст	4	доктор філософії	1	181 Харчові технології	5	денна	Ні
13	12	Навчально-науковий інст	4	доктор філософії	1	051 Економіка	6	денна	Ні
14	13	Навчально-науковий інст	4	доктор філософії	1	075 Маркетинг	7	денна	Ні
15	14	Навчально-науковий інст	4	доктор філософії	1	073 Менеджмент	5	заочна	Ні
16	15	Навчально-науковий інст	4	доктор філософії	1	051 Економіка	6	заочна	Ні
17	16	Навчально-науковий інст	4	доктор філософії	1	075 Маркетинг	7	заочна	Ні
18	17	Факультет автоматизації і	5	доктор філософії	1	073 Менеджмент	8	денна	Так
19	18	Факультет автоматизації і	5	доктор філософії	1	174 Автоматизація, комп'ютерно-інтегровані техн	9	денна	Ні
20	19	Факультет автоматизації і	5	доктор філософії	1	122 Комп'ютерні науки	8	заочна	Так
21	20	Факультет автоматизації і	5	доктор філософії	1	174 Автоматизація, комп'ютерно-інтегровані техн	9	заочна	Ні
22									

Рисунок Г.30 - Лист редагування довідкової інформації

Додаток Д. Фрагменти коду програми

```

version: '3.8'
services:
  app:
    build:
      context: .
      dockerfile: Dockerfile
    container_name: app
    environment:
      - MYSQL_HOST=mysql # Хост MySQL (потрібно використовувати 172.17.0.1,
                          # якщо використовується локальний MySQL)
      - MYSQL_PORT=3306 # Порт MySQL
      - MYSQL_DATABASE=db # Назва бази даних
      - MYSQL_USER=admin # Користувач MySQL
      - MYSQL_PASSWORD=admin # Пароль MySQL
      - AUTH_SECRET=OqW/n3/i/e65k0xpc6juz73UsqidP0hh5q0eV/kQptQ # Секрет для JWT
    volumes:
      - templates_data:/app/src/files # Зберігання шаблонів на локальному диску,
                                     # щоб не втратити їх при перезапуску контейнера
  mysql:
    image: mysql:latest
    container_name: mysql
    environment:
      MYSQL_ROOT_PASSWORD: admin # Пароль root
      MYSQL_DATABASE: db # Назва бази даних
      MYSQL_USER: admin # Користувач MySQL
      MYSQL_PASSWORD: admin # Пароль MySQL
    volumes:
      - mysql_data:/var/lib/mysql # Зберігання даних MySQL на локальному диску
    ports:
      - "3306:3306"
  phpMyAdmin:
    image: phpmyadmin/phpmyadmin
    container_name: phpmyadmin
    environment:
      PMA_HOST: mysql # Хост MySQL
      PMA_PORT: 3306 # Порт MySQL
    ports:
      - "8080:80"
    depends_on:
      - mysql
volumes:
  mysql_data:
    driver: local
  templates_data:
    driver: local

```

Рисунок Д.1 - Вміст конфігурації docker-compose.yml

```

providers: [
  Credentials({
    credentials: {
      email: {},
      password: {},
    },
    authorize: async (credentials) => {
      const { email, password } = credentials;
      if(typeof email !== 'string' || typeof password !== 'string'){
        throw new InvalidEmailOrPassword();
      }

      const [users] = await pool.query('SELECT * FROM users WHERE email = ?', [email]);
      const typedUsers = users as User[];
      if(typedUsers.length === 0){
        throw new InvalidLoginError();
      }
      const loginUser = typedUsers[0];

      //compare password
      const valid = await compare(password, loginUser.password);
      if(!valid){
        throw new InvalidLoginError();
      }
      return {
        id: loginUser.id.toString(),
        username: loginUser.username,
        email: loginUser.email,
        role: loginUser.role,
        subdivision_id: loginUser.subdivision_id
      }
    },
  )],
],

```

Рисунок Д.2 - Код відповідальний за авторизацію

```

export const POST = auth( async (req) => {
  try{
    if(!req.auth) return NextResponse.json({ error: "Unauthorized" }, { status: 401 });
    const initialData = await req.json();
    const requiredFields = ['directory_id', 'subdivision', 'directory_id']; // Перевірка на обов'язкові поля
    if (!requiredFields.every(field => field in initialData && initialData[field] !== '')) {
      return NextResponse.json({ error: "Обов'язкові поля відсутні" }, { status: 400 });
    }
    const data = await getVariables(initialData); // Мабинг змінних у файлах в об'єкт
    const fileNames = [EDU_FILE_NAME]; // Список файлів для генерації
    const endFileNames = [`_Договір_навчання_${new Date().getFullYear()}.docx`];
    const endPrefix = initialData?.pib_vstup.replace(' ', '_');
    if(initialData?.financing === 'phys'){
      fileNames.push(PHYS_FILE_NAME);
      endFileNames.push(`_Договір_фіз_особа_${new Date().getFullYear()}.docx`);
    }
    if(initialData?.financing === 'jur'){
      fileNames.push(JUR_FILE_NAME);
      endFileNames.push(`_Договір_юр_особа_${new Date().getFullYear()}.docx`);
    }
    const stream = new ReadableStream({ // Використовуємо ReadableStream для потокової передачі архіву
      async start(controller) {
        const archive = archiver('zip', { zlib: { level: 9 } });
        archive.on('data', (chunk) => controller.enqueue(chunk));
        archive.on('end', () => controller.close());
        archive.on('error', (err) => controller.error(err));
        // Додаємо файли до архіву
        for (const fileName of fileNames) {
          const buffer = await generateFile(data, fileName);
          archive.append(buffer, { name: endPrefix +
            endFileNames[fileNames.indexOf(fileName)]
          });
        }
        archive.finalize();
      }
    });
    return new NextResponse(stream, {
      status: 200,
      headers: {
        'Content-Type': 'application/zip',
        'Content-Disposition': `attachment; filename="contracts.zip"`,
      },
    });
  } catch (err) {
    return NextResponse.json({ error: (err as Error).message || (err as { code: string })?.code }, { status: 500 });
  }
});

```

Рисунок Д.3 - Обробник POST запиту на завантаження документів

```

FROM node:18-alpine AS base
FROM base AS deps
RUN apk add --no-cache libc6-compat
WORKDIR /app
COPY package.json yarn.lock* package-lock.json* pnpm-lock.yaml* .npmrc* ./
RUN \
  if [ -f yarn.lock ]; then yarn --frozen-lockfile; \
  elif [ -f package-lock.json ]; then npm ci --force; \
  elif [ -f pnpm-lock.yaml ]; then corepack enable pnpm && pnpm i --frozen-lockfile; \
  else echo "Lockfile not found." && exit 1; \
  fi
FROM base AS builder
WORKDIR /app
COPY --from=deps /app/node_modules ./node_modules
COPY . .
RUN \
  if [ -f yarn.lock ]; then yarn run build; \
  elif [ -f package-lock.json ]; then npm run build; \
  elif [ -f pnpm-lock.yaml ]; then corepack enable pnpm && pnpm run build; \
  else echo "Lockfile not found." && exit 1; \
  fi
FROM base AS runner
WORKDIR /app
ENV NODE_ENV=production
RUN addgroup --system --gid 1001 nodejs
RUN adduser --system --uid 1001 nextjs
COPY --from=builder /app/public ./public
COPY --from=builder --chown=nextjs:nodejs /app/.next/standalone ./
COPY --from=builder --chown=nextjs:nodejs /app/.next/static ./next/static
USER nextjs

EXPOSE 3000
ENV PORT=3000

ENV HOSTNAME="0.0.0.0"
CMD ["node", "server.js"]

```

Рисунок Д.4 – Конфігурація Dockerfile

Програмний код:

```

"use client";

import { Button } from "@components/ui/button";
import { saveAs } from "file-saver";
import React, { useState } from "react";
import { toast } from "sonner";
import { Alert, AlertDescription, AlertTitle } from "@components/ui/alert";
import { CircleAlert, Download } from "lucide-react";
import {
  Card,
  CardContent,

```

```

    CardDescription,
    CardFooter,
    CardHeader,
    CardTitle,
  } from "@components/ui/card";
import { Input } from "@components/ui/input";
import Link from "next/link";
const Page = () => {
  const [file, setFile] = useState<File | null>(null);

  const handleDownload = (fileName: string) => {
    fetch("/api/import", {
      method: "GET",
      headers: {
        "Content-Type":
          "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet",
      },
    })
      .then((res) => {
        if (!res.ok) throw new Error("Помилка завантаження");
        return res.blob();
      })
      .then((blob) => {
        saveAs(blob, `${fileName}`);
      })
      .catch((err) => console.error(err));
  };

  const handleFileChange = (e: React.ChangeEvent<HTMLInputElement>) => {
    if (e.target.files && e.target.files.length > 0) {
      setFile(e.target.files[0]);
    }
  };

  const handleUpload = async () => {
    if (!file) {
      toast.warning("Оберіть файл для завантаження");
      return;
    }

    const formData = new FormData();
    formData.append("file", file);

    try {
      const response = await fetch("/api/import", {
        method: "POST",
        body: formData,
      });

      if (!response.ok) {

```

```

const data = await response.json();
console.error("Помилка:", data);
toast.error(data.error || "Помилка завантаження файлу.");
return;
}

const data = await response.json();
console.log("Успішне завантаження:", data);
toast.success("База даних успішно імпортована");
} catch (error) {
  console.error("Помилка:", error);
  toast.error(
    (error as Error)?.message || "Помилка завантаження файлу.",
    {
      description:
        "Обов'язково перевірте цілісність БД після імпорту. Якщо  

        виникли проблеми , поверніть БД з резервної копії.",
      duration: 10000,
      position: "top-center",
    }
  );
}
};

return (
  <div className="py-8 space-y-4 flex flex-col">
    <h1 className="text-4xl font-bold">Імпортування даних</h1>
    <p>
      Для імпорту даних використовуйте шаблон Excel файлу, який можна
      завантажити нижче. В шаблоні вже заповнені поточні дані з бази
      даних та налаштовані випадючі списки для вибору значень.
    <br />
    Після заповнення шаблону, виберіть файл для завантаження та
    натисніть кнопку &quot;Імпортувати&quot;.
    </p>
    <Alert variant="destructive">
      <CircleAlert className="h-4 w-4" />
      <AlertTitle>Увага</AlertTitle>
      <AlertDescription>
        Під час імпорту даних буде <b>видалено всі існуючі дані</b>{" "}
        в імпортованих таблицях бази даних. <br />
        Переконайтеся, що ви зробили резервну{" "}
        <Link
          href="https://dev.mysql.com/doc/refman/8.4/en/mysqldump.html"
          className="underline"
          target="_blank"
        >
          <b>копію бази даних</b>
        </Link>{" "}
        перед імпортом, щоб уникнути втрати даних при помилках.

```

```

        </AlertDescription>
    </Alert>

    <Button
        onClick={() => handleDownload("Заготовка_Імпорту.xlsx")}
        size={"lg"}
        className="self-center"
    >
        <Download />
        Завантажити шаблон для імпорту
    </Button>

    <Card>
        <CardHeader>
            <CardTitle>Імпорт Excel файлу</CardTitle>
            <CardDescription>
                Завантажте файл для імпорту даних. Перед імпортом
                переконайтеся , що дані були заповнені вірно , без
                помилок.
            </CardDescription>
        </CardHeader>
        <CardContent>
            <Input
                type="file"
                accept=".xlsx"
                onChange={handleFileChange}
            />
        </CardContent>
        <CardFooter>
            <Button
                onClick={handleUpload}
                className="ml-auto"
                variant="outline"
                size="lg"
            >
                Імпортувати
            </Button>
        </CardFooter>
    </Card>
</div>
);
};

export default Page;

```

Програмний код генерації документів:

```

import { Degree, Directory, Subdivision } from "@types/db";
import { pool } from "../db";
import { promises as fs } from 'fs';

```

```

import PizZip from 'pizzip';
import Docxtemplater from 'docxtemplater';

import path from 'path';

export type AdmissionData = {
  year: string;
  pib_vstup: string;
  passport_vstup: string;
  address_vstup: string;
  ipn_vstup: string;
  phone_vstup: string;

  pib_legal: string;
  passport_legal: string;
  address_legal: string;
  ipn_legal: string;
  phone_legal: string;

  education_form: string; // денна = "денною", заочна = "заочною"

  education_program: string;
  education_specialty: string;
  degree: string;
  accreditation: string; // true = "акредитована", else "не акредитована"
  accreditation_term: string; // toLocaleDateString()
  credits: string;

  financing: string;

  pib_vstup_short: string; // Ім'я + ПРИЗВИЩЕ_КАПСОМ
  pib_legal_short: string;

  paidEntire:string,
  paidEntireW:string,

  y1Paid:string,
  y1PaidW:string,
  yy1:string,

  y2Paid:string,
  y2PaidW:string,
  yy2:string,

  y3Paid:string,
  y3PaidW:string,
  yy3:string,

  y4Paid:string,
  y4PaidW:string,

```

```

yy4:string,

y1DuePay:string,

anualPayDue:string,
semesterPayDue:string,

dean: string;

jur_name: string;
jur_pib: string;
jur_status: string;
jur_pib_short: string;
jur_req: string;

};

const getShortName = (name:string) => {
  //Отримати ім'я та прізвище з ПІБ , у вигляді 'Ім'я ПРИЗВИЩЕ_КАПСОМ'
  const [surname, namePart] = name.split(' ');

  return `${namePart} ${surname.toUpperCase()}`;
}

const formatDate = (date:Date)=> {
  const months = [
    'січня', 'лютого', 'березня', 'квітня', 'травня', 'червня',
    'липня', 'серпня', 'вересня', 'жовтня', 'листопада', 'грудня'
  ];

  const day = date.getDate();
  const month = months[date.getMonth()];
  const year = date.getFullYear();

  return `${day} ${month} ${year} р.`;
}

export const getVariables =async (initialData:{
  [key:string]:string
}) => {
  const [directory] = await pool.query('SELECT * FROM directory_with_program WHERE
id = ?', [initialData?.directory_id  ]);
  const typedDirectory = directory as Directory[];
  if(typedDirectory.length === 0){
    throw new Error('directory not found');
  }
  const curDirectory = typedDirectory[0];
  //Отримання поточного degree

```

```

    const [degree] = await pool.query('SELECT * FROM degree WHERE id = ?',
[curDirectory.degree_id]);
    const typedDegree = degree as Degree[];
    if(typedDegree.length === 0){
        throw new Error('degree not found');
    }
    const currentDegree = typedDegree[0];

    const [subdivision] = await pool.query('SELECT * FROM subdivisions WHERE id = ?',
[curDirectory.subdivision_id]);
    const typedSubdivision = subdivision as Subdivision[];
    if(typedSubdivision.length === 0){
        throw new Error('subdivision not found');
    }
    const currentSubdivision = typedSubdivision[0];

    const data:AdmissionData = {
        year: new Date().getFullYear().toString(),
        pib_vstup: initialData?.pib_vstup || '[ПІБ ВСТУПНИКА НЕ ЗНАЙДЕНО]',
        passport_vstup: initialData?.passport_vstup || '[ПАСПОРТ ВСТУПНИКА НЕ
ЗНАЙДЕНО]',
        address_vstup: initialData?.address_vstup || '[АДРЕСА ВСТУПНИКА НЕ
ЗНАЙДЕНО]',
        ipn_vstup: initialData?.ipn_vstup || '[ІПН ВСТУПНИКА НЕ ЗНАЙДЕНО]',
        phone_vstup: initialData?.phone_vstup || '[ТЕЛЕФОН ВСТУПНИКА НЕ ЗНАЙДЕНО]',

        pib_legal: initialData?.pib_legal || '',
        passport_legal: initialData?.passport_legal || '',
        address_legal: initialData?.address_legal || '',
        ipn_legal: initialData?.ipn_legal || '',
        phone_legal: initialData?.phone_legal || '',
        education_form: (initialData?.education_form === 'денна') ? 'денною' :
            (initialData?.education_form === 'заочна') ? 'заочною' :
'[ФОРМА НАВЧАННЯ НЕ ВКАЗАНА]',

        education_program: curDirectory.educationalProgram || '[ОСВІТНЯ ПРОГРАМА НЕ
ЗНАЙДЕНА]',
        education_specialty: curDirectory.specialtyName || '[СПЕЦІАЛЬНІСТЬ НЕ
ЗНАЙДЕНА]',
        degree: currentDegree.name || '[РІВЕНЬ НЕ ЗНАЙДЕНО]',
        accreditation: curDirectory.accreditation ? 'акредитована' : 'не
акредитована',
        accreditation_term: curDirectory.accreditationPeriod?.toLocaleDateString('uk-
UA') || '',
        credits: curDirectory.educationScope.toString() || '[КРЕДИТИ НЕ ВКАЗАНО]',

        financing: initialData?.financing === 'budget' ? 'державного бюджету' :

```

```

        (initialData?.financing === 'phys' || initialData?.financing ===
'jur' ) ? "коштів фізичних та/або юридичних осіб"
        : '[ФІНАНСУВАННЯ НЕ ВКАЗАНО]',

        pib_vstup_short: getShortName(initialData?.pib_vstup || '[ПІБ ВСТУПНИКА НЕ
ЗНАЙДЕНО]'),
        pib_legal_short: getShortName(initialData?.pib_legal || '[ПІБ ПРЕДСТАВНИКА НЕ
ЗНАЙДЕНО]'),

        paidEntire: curDirectory.paidCostEntireCourse.toLocaleString('uk-UA') ||
'[СУМА ЗА ВСІ КУРСИ НЕ ВКАЗАНА]',
        paidEntireW: curDirectory.paidCostEntireCourseWritten || '[СУМА ЗА ВСІ КУРСИ
НЕ ВКАЗАНА]',

        y1Paid: curDirectory.firstYearCost.toLocaleString('uk-UA') || '[СУМА ЗА
ПЕРШИЙ РІК НЕ ВКАЗАНА]',
        y1PaidW: curDirectory.firstYearCostWritten || '[СУМА ЗА ПЕРШИЙ РІК НЕ
ВКАЗАНА]',
        yy1: curDirectory.firstYearPeriod || '[ПЕРШИЙ РІК НЕ ВКАЗАНО]',

        y2Paid: curDirectory.secondYearCost.toLocaleString('uk-UA') || '[СУМА ЗА
ДРУГИЙ РІК НЕ ВКАЗАНА]',
        y2PaidW: curDirectory.secondYearCostWritten || '[СУМА ЗА ДРУГИЙ РІК НЕ
ВКАЗАНА]',
        yy2: curDirectory.secondYearPeriod || '[ДРУГИЙ РІК НЕ ВКАЗАНО]',

        y3Paid: curDirectory.thirdYearCost.toLocaleString('uk-UA') || '',
        y3PaidW: curDirectory.thirdYearCostWritten || '',
        yy3: curDirectory.thirdYearPeriod || '',

        y4Paid: curDirectory.fourthYearCost.toLocaleString('uk-UA') || '',
        y4PaidW: curDirectory.fourthYearCostWritten || '',
        yy4: curDirectory.fourthYearPeriod || '',

        //y1Due Pay - дата в такому форматі : 1 вересня 2024 р.
        y1DuePay: formatDate(curDirectory.firstYearPayDue) || '[ТЕРМІН ОПЛАТИ ЗА
ПЕРШИЙ РІК НЕ ВКАЗАНО]',

        anualPayDue: initialData?.payment_term === "щороку" ?
curDirectory.anualPayDue : "",
        semesterPayDue: initialData?.payment_term === "кожного семестру" ?
curDirectory.semesterPayDue : "",

        dean: currentSubdivision.dean || '[ДЕКАНА НЕ ЗНАЙДЕНО]',

        jur_name: initialData?.jur_name || '[НАЗВА ЮРИДИЧНОЇ ОСОБИ НЕ ВКАЗАНА]',
        jur_pib: initialData?.jur_pib || '[ПІБ ПРЕДСТАВНИКА ЮРИДИЧНОЇ ОСОБИ НЕ
ВКАЗАНО]',

```

```

        jur_status: initialData?.jur_status || '[ПОСАДА ПРЕДСТАВНИКА ЮРИДИЧНОЇ ОСОБИ
НЕ ВКАЗАНА]',
        jur_pib_short: initialData?.jur_pib_short || '[ІП ПРЕДСТАВНИКА ЮРИДИЧНОЇ
ОСОБИ НЕ ВКАЗАНО]',
        jur_req: initialData?.jur_req
        || '[ЮРИДИЧНІ ДАНІ НЕ ВКАЗАНІ]',

    }
    return data;
}

```

```

export const generateFile = async (variables: AdmissionData , fileName:string) => {
    const filePath = path.join(process.cwd(), "/src/files", fileName);

    const content = await fs.readFile(filePath, 'binary');
    const zip = new PizZip(content);
    const doc = new Docxtemplater(zip, {
        paragraphLoop: true,
        linebreaks: true,
    });

    doc.render(variables);

    const buffer = doc.getZip().generate({ type: 'nodebuffer' });

    return buffer;
}

```

```

export const getOriginalFile = async (fileName:string) => {
    const filePath = path.join(process.cwd(), "/src/files", fileName);

    const content = await fs.readFile(filePath, 'binary');

    const buffer = Buffer.from(content, 'binary');
    return buffer;
}

```