



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) АКС

Кафедра Інформаційних систем

Освітній ступінь бакалавр

Спеціальність 122 «Комп'ютерні науки»

(код і назва)

Освітньо-професійна програма Комп'ютерні науки

(назва)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри **Чумаченко С.М.**

“ ” 20 року

**З А В Д А Н Н Я**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА**

**Сулимка Роман Орестович**

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення інформаційної системи обліку даних про випускників кафедри інформаційних систем НУХТ

керівник роботи Костіков Микола Павлович, к.т.н., доц,

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

затверджені наказом закладу вищої освіти від “29” квітня 2021 року № 248-кв

2. Строк подання здобувачем роботи 27 травня 2021 р.

3. Вихідні дані до роботи Загальна характеристика кафедри інформаційних систем Національного університету харчових технологій, вимоги щодо об'єктів автоматизації, Інформація про випускників кафедри, сайт кафедри ІС

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

1) Провести системний аналіз діяльності кафедри інформаційних систем НУХТ

2) Розроблення технічного завдання. 3) Опис комплексу задач автоматизації

4) Охорона праці

5. Перелік графічного матеріалу

1) Організаційна структура кафедри інформаційних систем НУХТ 2)

Схема ВРWin для складання функцій системи. 3) Макет головної сторінки. 4)

Макет внутрішніх сторінок. 5) Концептуальна модель (ТО-ВЕ) системи, 6)

Схеми БД, 7) Скріншоти роботи системи.

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	доц.Костіков М.П.		
2	доц.Костіков М.П.		
3	доц.Костіков М.П.		
4	доц.Костіков М.П.		

7. Дата видачі завдання 29 квітня 2021 р.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження діяльності роботи кафедри інформаційних систем НУХТ		
2	Аналіз методів пошуку та використання інформації		
3	Аналіз існуючих систем		
4	Розробка системи для покращення збору інформації		
5	Розробка програмного продукту		
6	Оформлення роботи та розробка презентації		

**Здобувач**

\_\_\_\_\_

( підпис )

Сулимка Р.О.  
(прізвище та ініціали)

**Керівник роботи**

\_\_\_\_\_

( підпис )

Костіков М.П.  
(прізвище та ініціали)

## АНОТАЦІЯ

Об'єктом дослідження є можливість розробки і створення WEB-додатку для ведення обліку випускників.

Мета роботи – полягає у формуванні єдиного електронного інформаційного середовища для ведення обліку даних про випускників університету. Такий підхід передбачає використання відкритої інформаційної системи з можливістю додавання необхідних модулів та їх адаптації під зміни навчального процесу.

По своїй суті дипломний проєкт носив більш практичний характер, більше уваги приділялося саме створенню WEB-додатку з використанням сучасних засобів розробки. Робота велася з урахуванням можливості удосконалення.

Кінцевим результатом моєї роботи є розробка інформаційної системи обліку даних про випускників кафедри інформаційних систем НУХТ, створення дизайну та наповнення додатку, який дозволить отримати потрібну інформацію абітурієнтам, студентам, випускникам та викладацькому складу.

Головними перевагами в використанні таких технологій були:

- платформна незалежність мови Java;
- динамічність та багатопотоковість мови
- високопродуктивність та безпечність, об'єктно-орієнтований підхід;
- швидкість розробки WEB-сервісу за допомогою Angular та TypeScript;
- захист даних на сервері за допомогою Spring Security.

Кваліфікаційна робота бакалавра обсягом у 88 сторінок, містить 73 сторінки пояснювальної записки, 12 сторінок графічних матеріалів, 23 літературних джерел, 4 додатки.

**КЛЮЧОВІ СЛОВА:** JAVA, ANGULAR, SPRING BOOT, SPRING DATA, SPRING SECURITY, SPRING JPA, REST, TYPESCRIPT, POSTGRESQL, INTELLIJ IDEA, IDE.

## ANNOTATION

The object of research is the possibility of developing and creating a WEB-application for accounting for graduates.

The purpose of the work is to form a single electronic information environment for keeping records of data on university graduates. This approach involves the use of an open information system with the ability to add the necessary modules and adapt them to changes in the learning process.

In essence, the thesis project was more practical, more attention was paid to the creation of a WEB-application using modern development tools. The work was carried out taking into account the possibility of improvement.

The end result of my work is the development of an information system for data accounting data about the graduates of the Department of Information Systems NUFT, creating a design and content of the application, which will provide the necessary information to applicants, students, graduates and faculty.

The main advantages in using such technologies were:

- platform independence of the Java language;
- dynamism and multithreading of language
- high performance and safety, object-oriented approach;
- speed of WEB-service development using Angular and TypeScript;
- data protection on the server with Spring Security.

The qualification work of the bachelor occupies 88 pages, contains 73 pages of the explanatory note, 12 pages of graphic materials, 23 literary sources, 4 appendices.

**KEYWORDS:** JAVA, ANGULAR, SPRING BOOT, SPRING DATA, SPRING SECURITY, SPRING JPA, REST, TYPESCRIPT, POSTGRESQL, INTELLIJ IDEA, IDE.

## ЗМІСТ

<b>ВСТУП</b> .....	8
<b>РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ</b> .....	10
<b>1.1. Загальна характеристика НУХТ та кафедри інформаційних систем.</b>	10
<b>1.2. Організаційна структура НУХТ та кафедри інформаційних систем.</b>	11
<b>1.2.1. Загальна схема організаційної структури.</b> .....	11
<b>1.2.2. Структура кафедри інформаційних систем.</b> .....	12
<b>1.3. Аналіз нинішнього стану комп'ютеризації.</b> .....	18
<b>1.4. Розроблення функціональної моделі та аналіз наявних бізнес-процесів.</b>	19
<b>1.4.1. Функціональна модель обліку інформації про випускників кафедри</b>	19
<b>1.4.2. Проблеми, виявлені при аналізі</b> .....	21
<b>1.4.3. Задачі автоматизації.</b> .....	21
<b>1.5. Огляд наявних рішень-аналогів для розв'язання виявлених проблем</b>	22
<b>1.6. Обґрунтування доцільності проєктування й розроблення проєкту «Облік інформації про випускників кафедри».</b> .....	24
<b>1.7. Концептуальна модель системи.</b> .....	25
<b>1.8. Розрахунок економічного ефекту від впровадження системи.</b> ..	26
<b>Розділ 2. Технічне завдання на проєктування</b> .....	31
<b>2.1. Тематика і призначення Веб-сайту</b> .....	31
<b>2.2. Мова Веб-сайту</b> .....	31
<b>2.3. Структура і функціональність Веб-сайту</b> .....	31
<b>2.4. Функціональні можливості та дизайн розділів</b> .....	32
<b>2.5. Дизайн Веб-сайту</b> .....	34
<b>2.6. Інформаційне наповнення Веб-сайту</b> .....	35
<b>2.7. Панель адміністрування (CMS)</b> .....	35
<b>2.8. Процес розробки</b> .....	36

2.9. Технічна підтримка Веб-сайту.....	38
Розділ 3. Опис комплексу задач автоматизації.....	39
3.1. Обґрунтування вибору програмних засобів розроблення системи .....	39
3.2. Інформаційне забезпечення системи.....	41
3.3. Алгоритмізація та реалізація комплексу задач автоматизації.....	44
3.4. Інструкція користувача.....	48
3.5. Технічне та системне забезпечення розробки.....	57
3.5.1. Обґрунтування вибору ОС та протоколу обміну даними.....	57
3.5.2. Розробка і обґрунтування стратегії адміністрування системи..	59
3.5.3. Заходи захисту від несанкціонованого доступу до системи.....	60
4. ОХОРОНА ПРАЦІ .....	61
4.1. Вимоги до робочого місця користувача ЕОМ.....	61
4.2. Профілактичні заходи з охорони праці.....	62
ВИСНОВКИ .....	68
ДОДАТКИ.....	71
ДОДАТОК А. ОРГАНІЗАЦІЙНА СТРУКТУРА УНІВЕРСИТЕТУ ТА КАФЕДРИ ІНФОРМАЦІЙНИХ СИСТЕМ НУХТ.....	71
ДОДАТОК Б. ФУНКЦІОНАЛЬНІ МОДЕЛІ.....	72
ДОДАТОК В. МОДЕЛІ ТА СИСТЕМИ БАЗИ ДАНИХ.....	75
ДОДАТОК Г. СКРІН-ШОТИ.....	78
ДОДАТОК Д. ЛІСТИНГ ПРОГРАМИ.....	84

## ВСТУП

Сучасний навчальний заклад характеризується великою кількістю автоматизованих інформаційних систем які дозволяють збирати дані про університет та їх випускників.

На сьогоднішній день, мобільні пристрої та гаджети керують світом. Багато людей на можуть уявити день без улюбленого мобільного пристрою, який використовується скрізь: пошук інформації, зв'язок з друзями чи сім'єю, планування дня та багато інших речей. Але часто є потреба в додатку, який виконував би конкретні цілі та відповідав поставленим завданням. Ефективна система автоматизації процесів діяльності повинна будуватись за методологією дзеркал та відбивати результати діяльності або представляти плани та результати. Аналіз кожного модуля та потреби в його мобільності дозволяє сформуванню схеми відповідності та форм роботи з кожним модулем. Одним з модулів системи є формування даних про працевлаштування студентів які закінчили університет.

Створення інформаційної системи для перегляду даних про випускників є одним із важливих завдань, що вирішуються при плануванні навчального процесу. В першу чергу, це пов'язано з тим, що ці дані дозволять університету вивчити найбільш популярні напрямки підготовки студентів, модернізацію програми навчання, тощо.

WEB-додаток повинен мати зручний та зрозумілий інтерфейс, підключення до сервера для синхронізації та обміну даними, а також механізм сповіщень у разі настання непередбаченої події.

Програмний продукт можна використовувати на будь-якій операційній системі, для перегляду зі сторони користувача необхідний виключно сучасний браузер, зі сторони сервера теж немає обмежень, оскільки серверна частина програми реалізована на крос платформній мові програмування Java.

Розробка будь-якої програми, як правило, займає багато часу й потребує певного професійного досвіду. Однією з найпопулярніших технологій для

створення інформаційних систем є об'єктно-орієнтована мова Java з фреймворком Spring Boot, та фреймворк Angular з використанням мови програмування TypeScript.

Важливим для реалізації WEB-додатків є також середовище розробки (Integrated development environment або IDE). На даний момент найкращим середовищем для мови програмування Java є IDE від JetBrains – IntelliJ IDEA.

Також IntelliJ IDEA містить спеціалізовані засоби розробки та аналізу коду, специфічні для створення WEB-сайтів. При розробці даного проєкту використовувалася саме це середовище IDE.

З метою ознайомлення абітурієнтів, студентів та викладачів з інформацією про місце працевлаштування колишніх випускників кафедри інформаційних систем було створено WEB-додаток: «Облік випускників».

## РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ

### 1.1. Загальна характеристика НУХТ та кафедри інформаційних систем.

Університет почав свою роботу 1884 року, що робить його одним з найстаріших навчальних закладів, який готує високоякісних спеціалістів для галузей харчової промисловості.

До складу університету входять 42 кафедри, 24 філії, а також 8 навчально-науково-виробничих комплексів на передових підприємствах. Навчально-науковий комплекс «Національний університет харчових технологій» налічує приблизно 5000 працівників, серед яких: понад 120 професорів, докторів наук; приблизно 800 доцентів, кандидатів наук. Серед них 21 академік, 16 українських лауреатів національних премій, українські працівники науки і техніки, почесні працівники вищих навчальних закладів та видатні українські студенти.[1]

### Історія кафедри інформаційних систем

Кафедра інформаційних систем заснована в 2002 році, здійснює професійну підготовку бакалаврів і магістрів за напрямом 122 «Комп'ютерні науки», серед яких освітньо-професійні програми (ОПП) бакалаврів «Комп'ютерні науки» і «Інформаційні системи та штучний інтелект», а також ОПП магістрів «Інформаційні управляючі системи і технології» та «Комп'ютерний еколого-економічний моніторинг». Об'єктом професійної діяльності майбутніх фахівців є інформаційні та автоматизовані системи, засоби моніторингу контролю та керування, їх математичне, інформаційне, технічне й програмне забезпечення; способи і методи проєктування, розробки, впровадження, а також експлуатація. Для наукових досліджень — моделювання, аналізу та проєктування інформаційних систем використовуються 5 комп'ютерних аудиторій із актуальним програмним забезпеченням. Для самостійної і науково-дослідної робіт здобувачів організовано локальну комп'ютерну мережу в аудиторіях із

безкоштовним доступом в мережу інтернет. Талановиті студенти приєднуються до науково-дослідної роботи під керівництвом доцентів і професорів кафедри, виступають на наукових конференціях, публікують результати досліджень у фахових наукових виданнях, розробляють програмне забезпечення для відділів та служб університету, беруть участь у проєктуванні, створенні та впровадженні електронних засобів навчання. Крім того, студенти кафедри є активними учасниками міжнародних і всеукраїнських олімпіад із програмування: Міжнародна відкрита студентська олімпіада із програмування ім. С.О. Лебедєва та В.М. Глушкова «КРІ-OPEN», що проводиться на базі НТУУ «КПІ», Міжнародна Літня школа із програмування, Міжнародна Зимової школа із програмування [2] тощо.

## 1.2. Організаційна структура НУХТ та кафедри інформаційних систем.

### 1.2.1. Загальна схема організаційної структури.

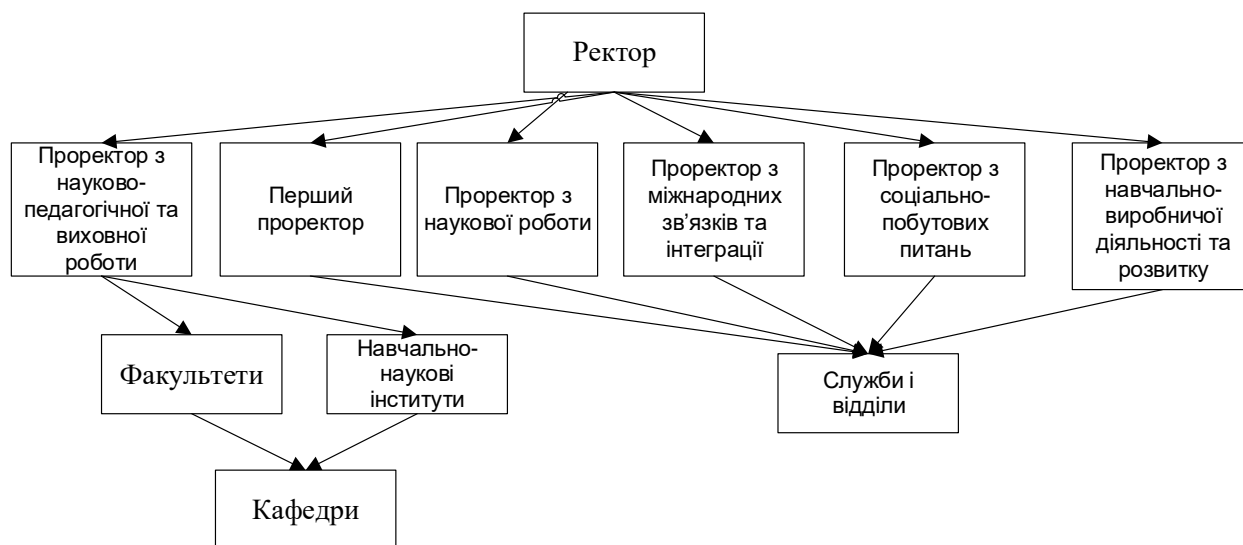


Рисунок 1.1. Загальна схема структури Національного університету харчових технологій.

### 1.2.2. Структура кафедри інформаційних систем.

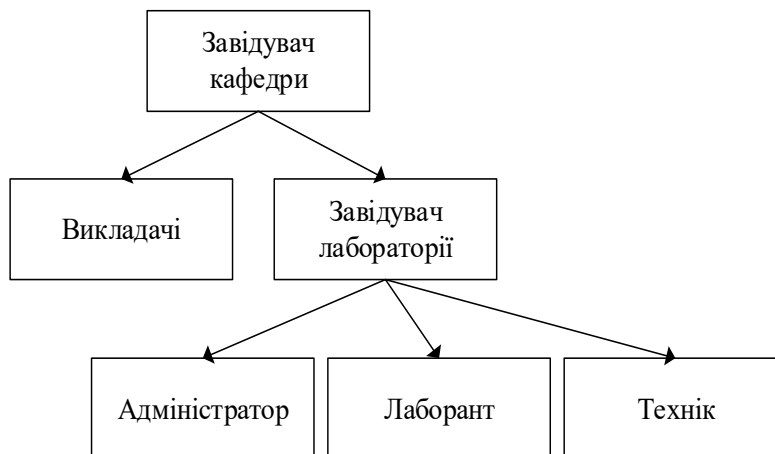


Рисунок 1.2. Структура кафедри інформаційних систем.

На кафедрі інформаційних систем студентів навчають за спеціальністю «Комп'ютерні науки», бакалаврів готують за ОПП «Комп'ютерні науки» та «Інформаційні системи та штучний інтелект», а магістрів за ОПП «Інформаційні управляючі системи і технології» та «Комп'ютерний еколого-економічний моніторинг». [3].

Таблиця 1.1. Функції кафедри.

№	Функція
Освітня діяльність	
1	Проведення на сучасному рівні освітнього процесу на всіх формах навчання різних видів навчальних занять відповідно до затверджених навчальних планів і графіків освітнього процесу за освітніми ступенями бакалавра та магістра.
2	Підготовка ліцензійних справ на ведення освітньої діяльності за спеціальностями і підготовка справ по акредитації освітньо-професійних програм.
3	Здійснення науково-методичного забезпечення навчальних дисциплін кафедри: підготовка навчальних і робочих програм дисциплін,

	підручників, посібників, розроблення навчально-методичних матеріалів для всіх видів навчальних занять, а також самостійної роботи здобувачів й інших рекомендацій, ефективне використання відео-, аудіо- та комп'ютерної техніки.
4	Впровадження Європейської кредитної трансферно-накопичувальної системи в освітній процес та рейтингового оцінювання академічних досягнень здобувачів вищої освіти.
5	Здійснення постійного контролю якості навчання студентів із дисциплін кафедри, аналіз результатів модульного, рубіжного і підсумкового контролів, забезпечуючи належні умови для їх якісного проведення.
6	Організація та контроль самостійної роботи студентів і сприяння у вивченні ними навчальних дисциплін кафедри.
7	Забезпечення середовища освітнього процесу необхідними електронними освітніми ресурсами.
8	Ведення електронних журналів обліку навчальної роботи студента, журналів обліку роботи науково-педагогічних працівників для аналізу результатів навчання з метою вживання заходів для підвищення якості освіти.
9	Впровадження і вдосконалення системи забезпечення якості освітньої діяльності, а також якості вищої освіти (система внутрішнього забезпечення якості).
10	Дотримання академічної доброчесності науково-педагогічними працівниками: розроблення й застосування засобів діагностування змісту та якості навчання (завдання тестів, модульних контрольних робіт, екзаменаційні білети та інше), а також систем оцінювання рівнів знань, умінь, навичок та інших компетенцій.
11	Внесення пропозицій деканові факультету/директору інституту щодо

	складу Екзаменаційних комісій за спеціальностями.
12	Організація атестації випускників, забезпечення умов ефективної роботи Екзаменаційних комісій.
13	Розвиток зв'язків із організаціями, установами, підприємствами для вдосконалення практичної підготовки здобувачів, вивчення сучасних технологій виробництва та форм організації праці, керування виробничими процесами; підвищення кваліфікації фахівців, популяризація наукових знань і працевлаштування випускників.
14	Дослідження регіонального ринку освітніх послуг за спеціальностями й розроблення пропозицій по оптимізації ступеневих освітніх програм.
З навчально-методичної діяльності	
15	Розроблення та вдосконалення переліку компетентностей для підготовки бакалаврів, магістрів та аспірантів за спеціальностями.
16	Розроблення на основі стандартів вищої освіти освітніх програм, структурно-логічних схем і навчальних планів зі спеціальностей різних рівнів вищої освіти.
17	Розроблення, узгодження та затвердження робочих навчальних планів, навчальних програм дисциплін та робочих програм навчальних дисциплін.
18	Виконання освітніх програм і робочих навчальних планів щодо вільного вибору студентів за спеціальностями.
19	Розроблення та затвердження засобів діагностики рівня знань, вироблення чітких критеріїв виконання та оцінювання всіх видів робіт, завдань, що передбачені робочою програмою навчальної дисципліни;
20	Розроблення та затвердження наскрізних програм та програм практик.
21	Розроблення та впровадження освітньо-професійних, освітньо-наукових

	програм за ліцензованими спеціальностями.
22	Удосконалення змісту навчання з урахуванням сучасних досягнень науки, техніки, технологій і організації операційної діяльності.
23	Забезпечення впровадження прогресивних методів і новітніх методик, сучасних інформаційних і інноваційних технологій навчання.
24	Розроблення та затвердження тематики курсових, дипломних бакалаврських/магістерських робіт (проектів).
25	Підготовка наукових монографій, підручників, посібників, ін. навчально-методичних матеріалів згідно із поточними та перспективними планами кафедри.
26	Планування заходів підвищення педагогічної майстерності науково-педагогічних працівників.
З наукової та інноваційної діяльності:	
27	Підготовка аспірантів і докторантів, контроль за термінами виконання освітньо-наукової програми та публічного захисту дисертацій.
28	Проведення попереднього атестування аспірантів і докторантів.
29	Здійснення підготовки та підвищення кваліфікації науково-педагогічних кадрів, розгляд наукових дисертацій, підготовлених до захисту членами кафедри чи здобувачами за дорученням ректора.
30	Ведення науково-дослідної роботи (НДР) з теоретичних, науково-технічних і соціально-економічних проблем свого профілю та проблем педагогіки вищої школи на базі бюджетного, договірною та грантового фінансування.
31	Здійснення маркетингових заходів щодо просування на ринки й комерціалізація результатів виконання НДР.

32	Обговорення і впровадження результатів НДР у практичну діяльність і освітній процес, видача рекомендацій для публікації.
34	Забезпечення підготовки й видання публікацій (монографій, словників, довідників, стандартів, наукових журналів, статей, доповідей тощо).
35	Організація, проведення та участь науково-педагогічних працівників та здобувачів вищої освіти у наукових семінарах, міжнародних, всеукраїнських конференціях, конкурсах, виставках тощо.
36	Створення умов для залучення здобувачів вищої освіти до наукової роботи.
37	Здійснення контролю за здобувачами вищої освіти, які беруть участь у: виконанні науково-дослідної роботи, студентських наукових гуртках, наукових конференціях, семінарах.
38	Організація міжнародної співпраці в напрямі академічної мобільності, розширення участі науково-педагогічних працівників, студентів та аспірантів Університету в міжнародних програмах академічних обмінів.
З організаційної роботи	
39	Розгляд індивідуальних планів роботи науково-педагогічних працівників кафедри; вивчення, узагальнення та розповсюдження кращого досвіду роботи та щорічне визначення їх педагогічного та наукового рейтингу.
40	Участь у складанні розкладів навчальних занять, заліків й екзаменів та роботи Екзаменаційних комісій.
41	Сприяння у проходженні стажування та підвищенні кваліфікації науково-педагогічних працівників кафедри, контроль за виконанням плану підвищення кваліфікації НПП кафедри.
42	Проведення організаційної та індивідуальної роботи з науково-педагогічними працівниками кафедри, студентами, аспірантами,

	докторантами, спрямованої на їх активну участь в освітньому процесі, на розвиток професійної культури, дотримання етичних норм, правил внутрішнього розпорядку Університету, забезпечення атмосфери вимогливості, доброзичливості та взаємної поваги у стосунках між працівниками, науково-педагогічними працівниками та здобувачами вищої освіти.
43	Встановлення творчих зв'язків з закладами освіти, галузевими організаціями, підприємствами, науково-дослідними інститутами, іншими підприємствами та організаціями незалежно від форм власності, зокрема з іноземними.
44	Організація та контроль за проведенням науково-педагогічними працівниками кафедри навчальних занять, практик тощо.
45	Організація роботи кураторів академічних груп.
46	Участь в організації та проведенні вступної кампанії.
47	Проведення профорієнтаційної роботи з метою залучення осіб на навчання за спеціальностями, за якими кафедра готує фахівців.
48	Проведення роботи щодо забезпечення систематичного зв'язку з випускниками Університету.
49	Підготовка договорів на проведення практик.
50	Організація і супроводження веб-сторінки кафедри.
51	Організація висвітлення результатів діяльності кафедри на інформаційних стендах, у соціальних мережах і в засобах масової інформації.
52	Організація діяльності з поширення платних освітніх послуг відповідно до законодавства України та нормативних документів Університету.
53	Організація проведення Всеукраїнських студентських олімпіад з дисциплін та спеціальностей.

Таблиця 1.2. Інформація яку підрозділ отримує, а яку передає.

Підрозділ	Кафедра надає	Кафедра одержує
Відділ кадрів	- список вільних посад та критерії до кандидатів;	- список потенційних кандидатів на посаду;
Технічний відділ	- список обладнання, яке потрібно для роботи; - перелік обладнання, яке потрібно відремонтувати або замінити;	- обладнання для роботи; - нове або відремонтоване обладнання;
Виконавці НДР	- звіти по НДР; - публікації; - статті;	- терміни роботи зі звітами; - теми НДР;

### 1.3. Аналіз нинішнього стану комп'ютеризації.

Зараз на кафедрі функціонує WEB-версія інформаційної системи про випускників яка функціонує на сайті кафедри. В теперішній реалізації системи дуже малий функціонал, відсутня можливість перегляду більшої кількості даних, таких як:

- історія роботи;
- посилання на соцмережі;
- можливість пошуку за прізвищем;
- дата останнього оновлення;

Низький рівень інтуїтивності інтерфейсу в сумі з повільною швидкістю роботи програми приводить нас до висновку, що систему слід покращувати, створивши грамотний UI та UX дизайн, створити таблиці з пагінацією, щоб користувач мав змогу встановлювати кількість записів на сторінці, та доповнити дані про випускників.

## 1.4. Розроблення функціональної моделі та аналіз наявних бізнес-процесів.

### 1.4.1. Функціональна модель обліку інформації про випускників кафедри

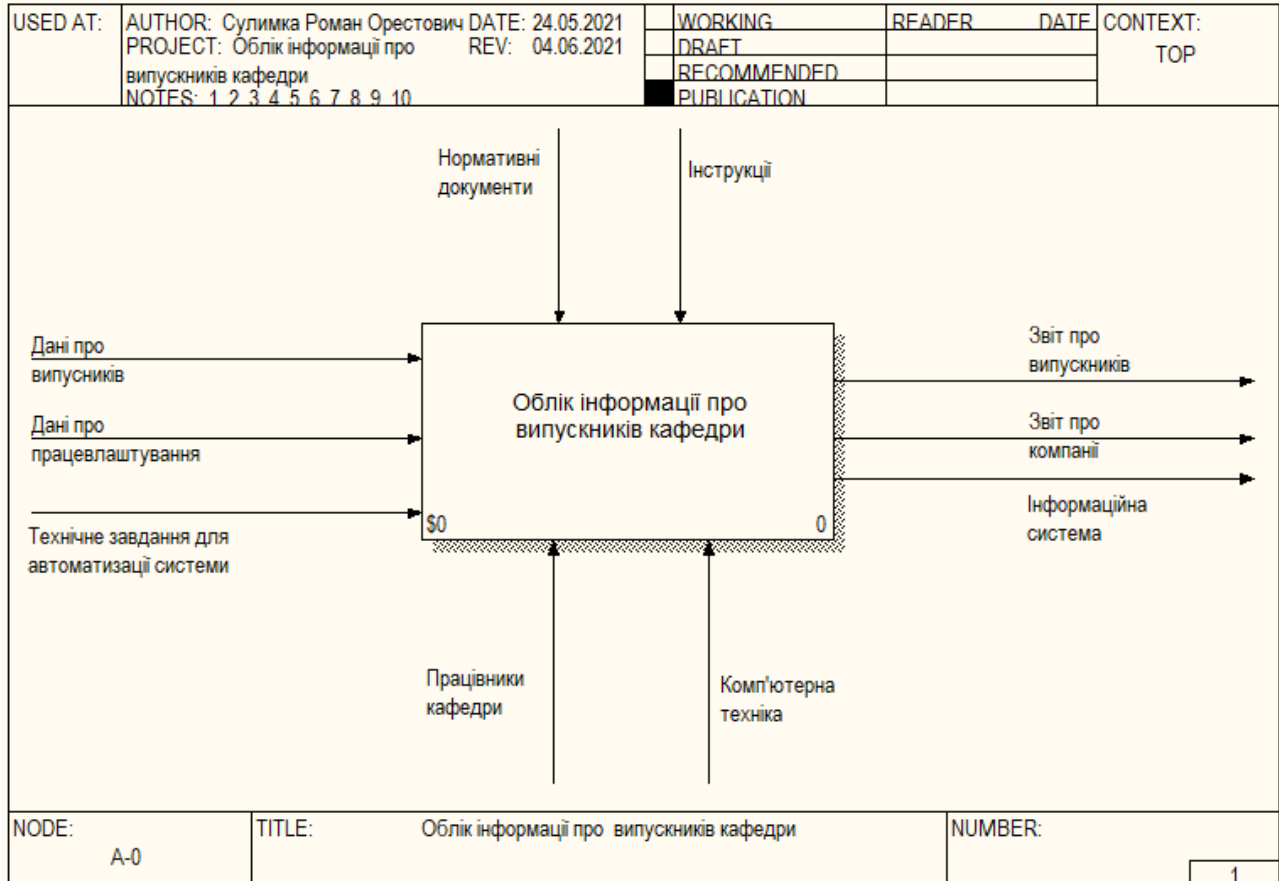


Рисунок 1.1. Контекстна діаграма моделі

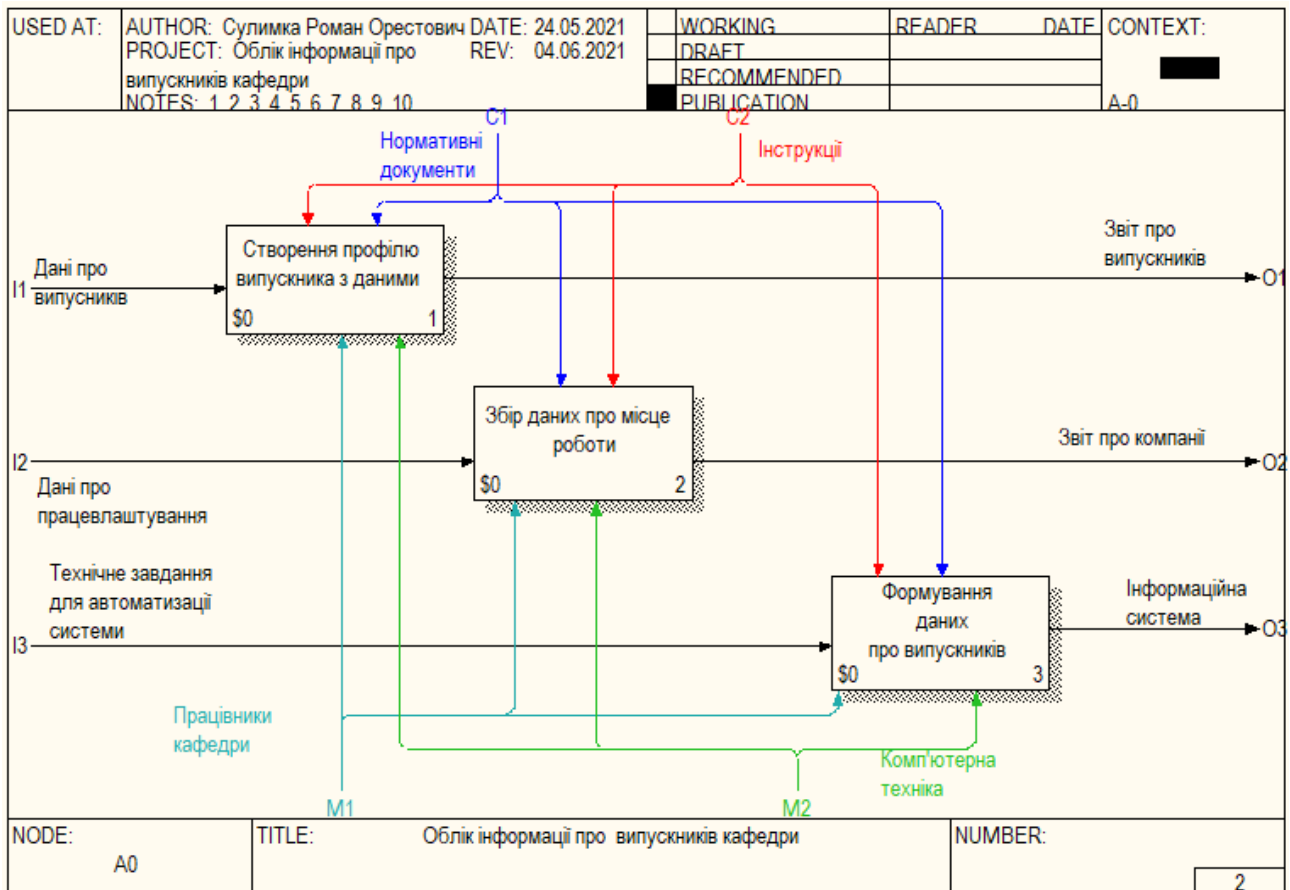


Рисунок 1.2. Перший рівень декомпозиції

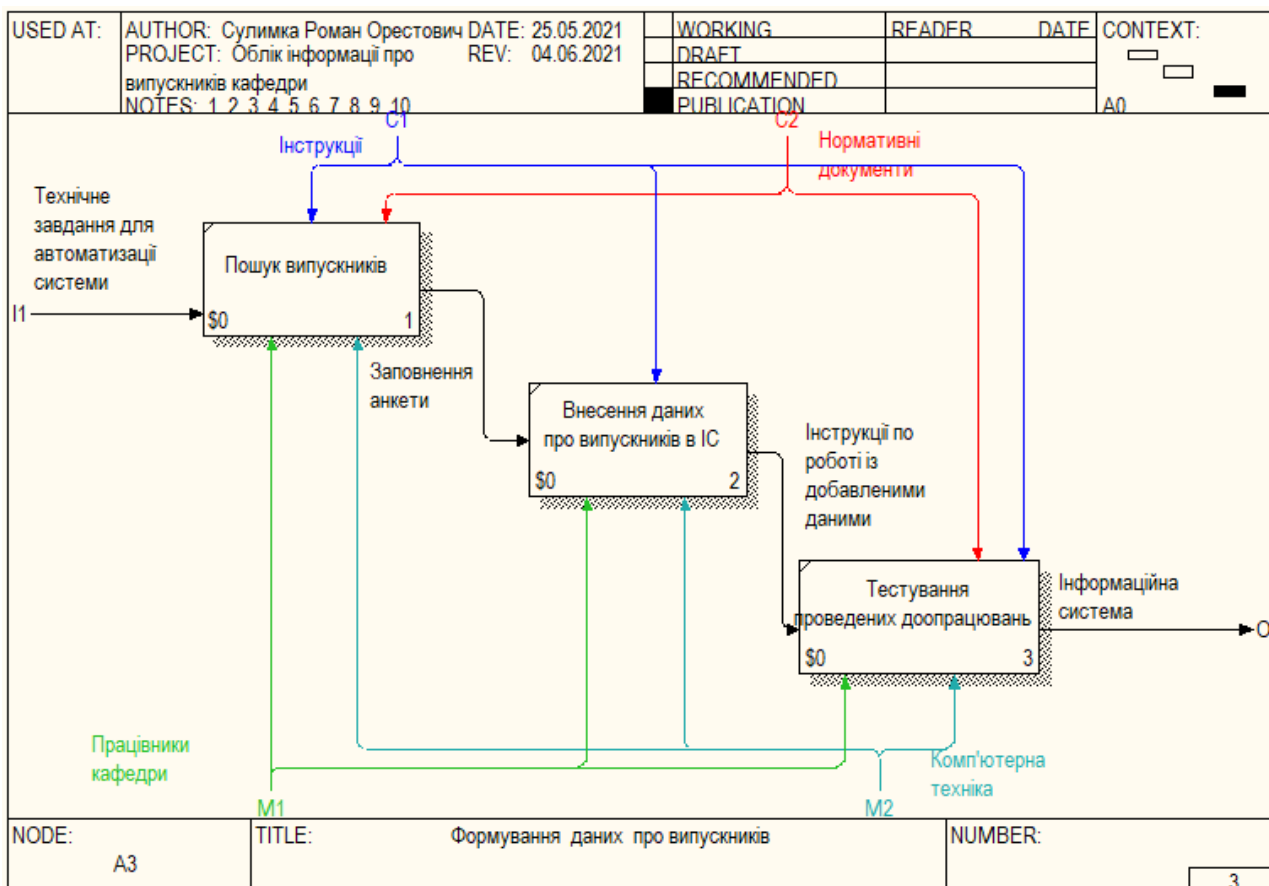


Рисунок 1.3. Другий рівень декомпозиції.

### 1.4.2. Проблеми, виявлені при аналізі

Після аналізу інформаційної системи були виявлені такі недоліки, як:

- Відсутня можливість пошуку випусників за прізвищем;
- Неможливо переглянути історію працевлаштування випускника;
- Відсутня інформація про дату останнього оновлення інформації про випускника;
- Немає можливості переглянути контакти випускника в такій мережі як LinkedIn.
- Відсутня інформація про рік закінчення бакалаврату, магістратури та ОКР Спеціаліст.
- Немає змоги проаналізувати технології, які випускники використовують у своїй роботі

### 1.4.3. Задачі автоматизації.

Для усунення зазначених проблем необхідно:

- Реалізувати можливість пошуку випускників за прізвищем, для зручної навігації між користувачами;
- Реалізувати діалогове вікно з розширеною інформацією про місця працевлаштування, посади, та роки випуску з освітніх спеціальностей випускника;
- Створити поле, яке буде показувати дату останнього оновлення інформації про випускника;
- Створення кнопок з посиланням на соціальні мережі випускників;
- Реалізувати аналіз технологій які використовують випускників.

### **1.5. Огляд наявних рішень-аналогів для розв'язання виявлених проблем**

На ринку представлені такі програми конкуренти як: LinkedIn, Facebook.

LinkedIn — Соціальна мережа для пошуку та встановлення ділових контактів, яка була придбана компанією Microsoft.

LinkedIn надає можливість зареєстрованим користувачам створювати і підтримувати список ділових контактів. Контакти можуть бути запрошені як з сайту, так і ззовні, проте LinkedIn вимагає попереднє знайомство з контактами. У випадку, коли користувач не має прямого зв'язку з контактом, він може бути представленим через інший контакт.

Користувачі LinkedIn можуть використовувати список контактів у різних цілях:

- Бути представленими через існуючі контакти і розширювати зв'язки;
- Здійснювати пошук компаній, людей, груп за інтересами;
- Публікувати професійні резюме і здійснювати пошук роботи;
- Рекомендувати і бути рекомендованими;
- Публікувати вакансії;
- Створювати групи за інтересами.

Facebook Workplace - Головна соціальна мережа планети, яка з кожним днем все більше розширює свій функціонал. Вже зараз через Facebook можна робити практично все: спілкуватися, читати новини, дізнаватися про заходи, купувати і

продавати речі, при великому бажанні можна навіть забронювати столик в ресторані і записатися на стрижку.

Інструменти для пошуку роботи також активно впроваджуються в структуру соцмережі. В даний момент, наприклад, тестується функція, що дозволяє публікувати на сайті вакансії і отримувати на них відгуки.

Таблиця 1.3. Порівняння існуючих систем.

Назва	Linkedin	Facebook Workplace
Наявність безкоштовної версії	+	-
Ціна розширеної версії	29.99 USD/month	8 USD/month
Максимальна кількість пошуку профілів поза контактами	10 профілів/день	Без обмеження
Мова інтерфейсу	Англійська, Російська, Німецька, Польська	Англійська, Іспанська, Італійська
Пошук профілів за ВНЗ	+	-

Програми аналоги не дадуть нам можливості переглядати дані студентів саме одного університету та однієї кафедри, також випускники не завжди зареєстровані в цих програмах.

## **1.6. Обґрунтування доцільності проєктування й розроблення проєкту**

### **«Облік інформації про випускників кафедри».**

Як бачимо, наявний стан автоматизації обліку інформації про випускників має недоліки, а існуючі системи не можуть покрити всіх виявлених проблем.

У зв'язку з цим вирішено створити власну систему, в якій буде реалізовано всі поставлені задачі автоматизації.

Завдяки цьому функціонал буде значно розширений, проєкт переміститься з сторінки на сайті кафедри на окремий web-сайт. Користувачі зможуть отримати більше корисних даних про випускників.

Беручи до уваги недоліки існуючої системи, та систем аналогів, рекомендовано створити новий додаток, в якому будуть виправлені виявлені недоліки, та реалізований більш обширний функціонал.

Програму можна наділити такими функціями, які не присутні в жодному аналогу, наприклад:

- статистика компаній, в яких працюють випускники;
- статистика технологій, які використовують випускники;
- модерування даних всіх випускників;

### 1.7. Концептуальна модель системи.

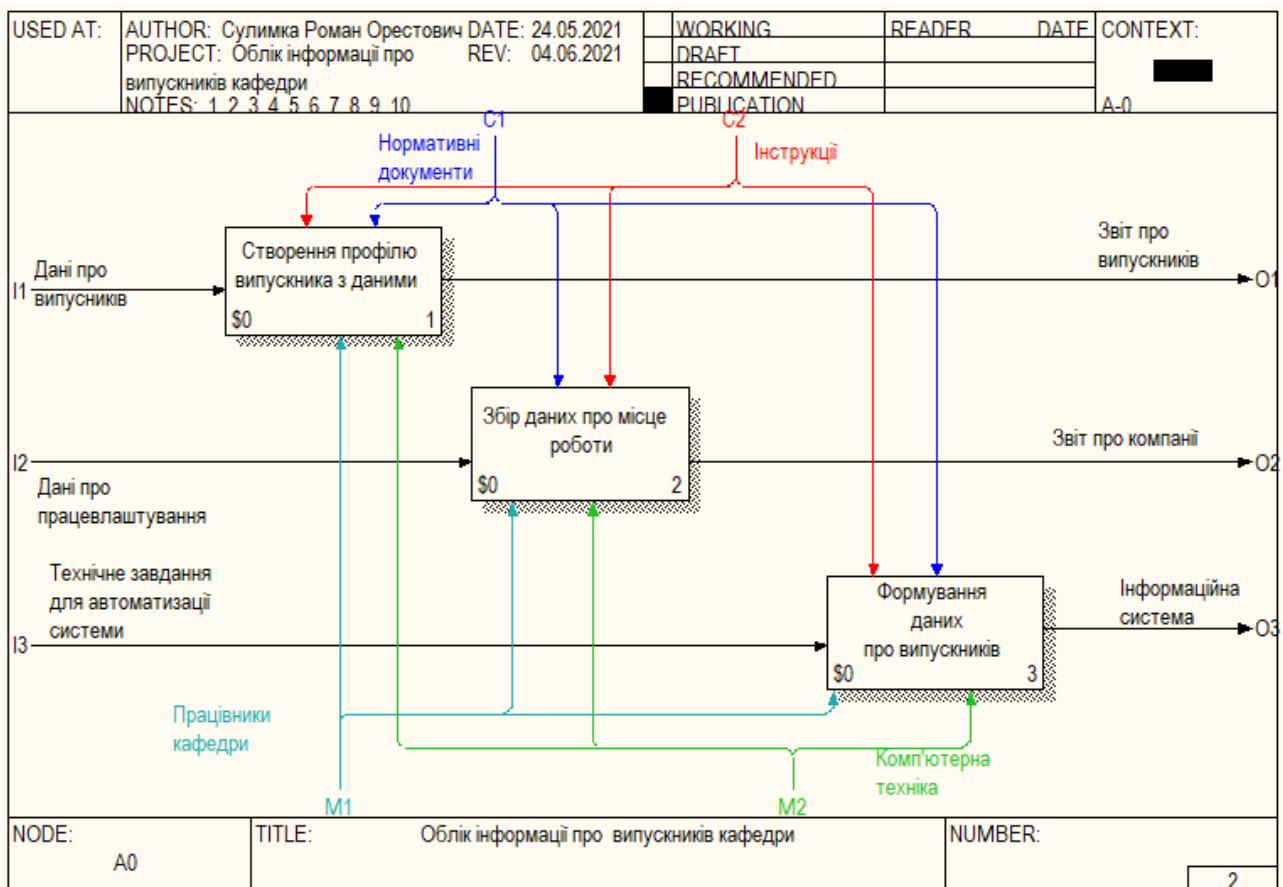


Рисунок 1.4. Модель AS-IS

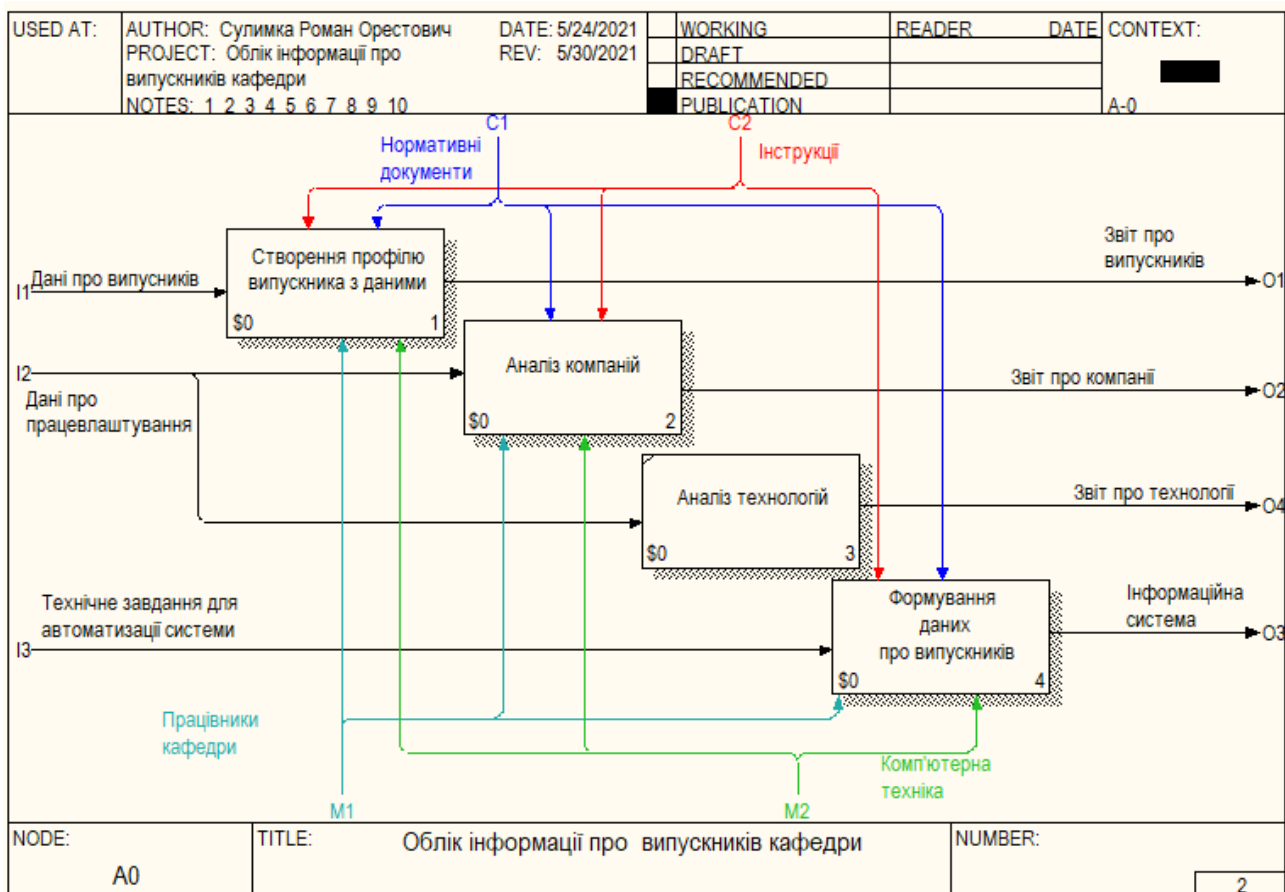


Рисунок 1.5. Модель To-Be.

### 1.8. Обчислення економічного ефекту від упровадження системи.

Вхідні дані для обчислення

Визначення розміру оплати праці

Вид системи:

Удосконалення системи обліку випусників.

Розрахунки проводять для кожної стадії розроблення системи. При цьому визначаються такі стадії:

- ескізний проєкт (передпроєктне дослідження);
- технічне завдання на проєкування;
- технічний проєкт;
- робочий проєкт;
- упровадження.

Ступінь новизни розроблюваних задач – В.

Група складності алгоритму за їх характеристикою:

Алгоритми обліку, звітності, статистики пошуку.

Визначається вид інформації, що використовується, на базі аналізу вхідної та вихідної інформації з функціональної моделі системи.

Таблиця 1. Визначення видів інформації.

Вид інформації	Позначення	К-ть наборів даних
К-ть видів змінної інформації	ЗІ	m=4
К-ть видів нормативно-довідкової інформації	НДІ	n=4
К-ть банків (баз) даних	БД	p=1
Оброблення в режимі реального часу	РЧ	Так
Забезпечення телекомунікаційної обробки даних та управління віддаленими об'єктами	ТОУ	Ні
Входи	ВХ	4
Виходи	ВИХ	5

Визначаються затрати часу на розроблення ескізного проєкту (передпроектного дослідження) T1 та технічного завдання T2 за такими даними:

Ескізний проєкт, T1 = 34.

Технічне завдання, T2 = 20.

Визначаються затрати часу на стадіях «технічний проєкт», «робочий проєкт» і «впровадження».

$$ТБ3=60$$

$$ТБ4=124$$

$$ТБ5=44$$

– Визначення затрат часу для стадії «технічний проєкт» (T3).

$$k_{\text{П}} = k_1 * m + k_2 * n + k_3 * p_{m+n+p}$$

$$k_{\text{П}} = (1 * 4 + 0.72 * 3 + 2.08 * 1) / (4 + 3 + 1) = 1.03$$

$$T_3 = T_{\text{Б3}} * k_{\text{П}} * k_{\text{О}}$$

$$T_3 = 60 * 1.03 * 1.26 = 77.868$$

– Визначення затрат часу на стадії «робочий проект» (T4).

$$k_{\text{П}} = k_1 * m + k_2 * n + k_3 * p_{m+n+p}$$

$$K_{\text{р}} = (1 * 4 + 0.58 * 4 + 0.48 * 1) / (4 + 4 + 1) = 0.56$$

$$T_4 = T_{\text{Б4}} * k_{\text{П}} * k_{\text{О}} * k_{\text{С}}$$

$$T_4 = 124 * 0.56 * 1.32 * 1 = 91.66$$

– Визначення затрат часу на стадії «упровадження» (T5).

$$k_{\text{П}} = k_1 * m + k_2 * n + k_3 * p_{m+n+p}$$

$$K_{\text{р}} = (1 * 4 + 0.58 * 4 + 0.48 * 1) / (4 + 4 + 1) = 0.56$$

$$T_5 = T_{\text{Б5}} * k_{\text{П}} * k_{\text{О}} * k_{\text{С}}$$

$$T_5 = 44 * 0.56 * 1.32 * 1 = 32.52$$

Для визначення загальних затрат часу на розроблення системи використовуємо:

$$T_{\Sigma} = T_1 + T_2 + T_3 + T_4 + T_5 = 34 + 24 + 77.686 + 91.66 + 32.52 = 259.86 \text{ годин}$$

Визначення чисельності необхідних виконавців:

$$Ч = T_{\Sigma} / \Phi$$

$$\Phi = 75 \text{ днів}$$

$$Ч = 259.86 / 75 = 4 \text{ виконавці.}$$

Оплата праці виконавців підраховуємо за формулою:

$$V_1' = Ч * M * ЗППР = 4 * 3 * 10\,000 = 120\,000 \text{ грн}$$

Затрати, пов'язані із розробленням програми на ПК

Розрахунок річного фонду часу роботи ПК в годинах:

$$T_{\text{ПК}} = T_{\text{ОП}} - (6 * 8 + 5 * 12) = 2000 - (6 * 8 + 5 * 12) = 1892 \text{ год}$$

$$T_{\text{ПК}}' = T_{\text{ПК}} * R_{\text{ТОП}} = 1892 * (459 / 2000) = 425.7 \text{ год}$$

Поточні затрати на експлуатацію V1”:

$$Ц_{\text{ПК}} = Ц_{\text{р}} * (1 + k_{\text{УН}})$$

$$ЦПК=5000 * (1+0.12)= 5600 \text{ гривень.}$$

$$ЗАМ=ЦПКНА$$

$$Зам = 5600/5=1120 \text{ гривень}$$

$$ЗЕЛ=РПК*ТПК*ЦЕЛ*А = 633.15 \text{ грн}$$

$$РПК = 0.7 \text{ кВт,}$$

$$ТПК = 450 \text{ год,}$$

$$ЦЕЛ = 0,9 \text{ грн/кВт,}$$

$$А = 0.9.$$

$$Зр=ЦПК*0.06$$

$$Зр=21\ 280*0.06=1276.8 \text{ грн}$$

$$ЗМАТ=ЦПК*0.05$$

$$Змат = 4100 * 0.05 = 280 \text{ грн}$$

$$V1''=ЗОП+ЗАМ+ЗЕЛ+Зр+ЗМАТ$$

$$V1''=0 + 1120 + 145 + 336 + 280 = 1888$$

$$V1=V1'+V1''$$

$$V1=45000 + 1888=46888 \text{ гривень.}$$

Затрати на придбання та встановлення ПК V2.

$$V2 = 7000 \text{ грн}$$

Затрати на підготовлення приміщення V3.

$$V3=2000 \text{ грн}$$

Витрати на навчання персоналу — V4:

$$V4=5000 \text{ грн}$$

Загальна вартість розроблення та впровадження в дію системи.

$$V\Sigma=V1+V2+V3+V4$$

$$V = 46888 + 7000 + 2000 + 5000 = 60888 \text{ грн.}$$

$$Vp=V\SigmaНА$$

$$Vp=60888 / 5 = 12177,6 \text{ грн.}$$

Додаткове фінансування Пр = 12000 грн.

$$KE\Phi = PrVp$$

$$Ke\phi = Pr/U_p = 12000/12177,6 = 0,98$$

Термін окупності розробки дорівнюватиме  $T_{ок} = 1/Ke\phi = 1/0,98 = 1,02$  р.

## **Розділ 2. Технічне завдання на проєктування.**

### **2.1. Тематика і призначення Веб-сайту**

Дане Технічне завдання визначає вимоги та порядок розробки веб-сайту: «Розроблення інформаційної системи обліку даних про випускників кафедри інформаційних систем НУХТ».

Мета сайту - створити зручну веб-платформу, що зберігає в одному місці дані про випускників кафедри інформаційних систем НУХТ і дає швидко зрозуміти ким та де на якій позиції працюють випускники.

### **2.2. Мова Веб-сайту**

Мовна локалізація Web-сайту – українська;

### **2.3. Структура і функціональність Веб-сайту**

Сайт повинен складатися з наступних розділів:

Пункти меню:

- Про Проєкт
- Випускники
- Користувачі
- Контакти
- Статистика
- Про автора

Об'єкти сайту:

- Форма реєстрації
- Таблиця випускників
- Пошукова система
- Форма зворотного зв'язку
- Сторінка Додати дані про випускника

## 2.4. Функціональні можливості та дизайн розділів

### Ф1. Логотип

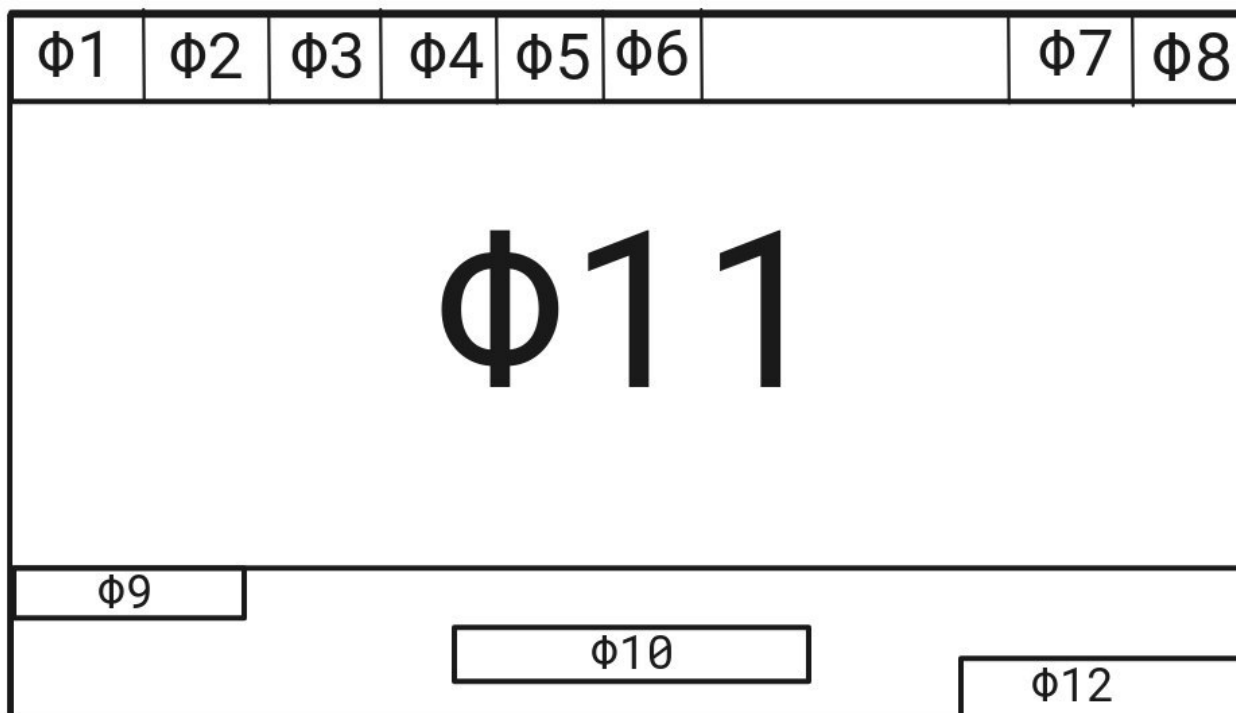


Рисунок 2.1. Модель головної сторінки інформаційної системи.

### Ф2. Головна сторінка

Дизайн головної сторінки необхідно виконати в спокійній кольоровій гамі. Дизайн повинен бути виконаний в стилі мінімалізму (нічого зайвого, відволікаючого увагу). Види і розміри шрифтів повинні бути комфортними для перегляду користувачами. Кнопки меню повинні бути чіткими, має бути присутнім чіткий розділ між пунктами меню.

На головній сторінці відображається коротка інформація про проєкт, достатньо виділене посилання-перехід на сторінку «Створити Випускника».

### Ф3. Адмін-панель

Розміщені форми з даними на яких знаходяться всі дані про користувача. Якщо користувач з роллю Admin або Moderator, він може редагувати збережені дані. Також з цього пункту можна перейти на детальну історію роботи випускника. Також тут знаходиться форма для користувачів, присутні функції: змінити роль користувача, ім'я, електронну пошту та пароль.

#### Ф4. Користувачі

Форма для користувачів з роллю User, щоб переглянути користувачів в системі.

#### Ф5. Випускники

Містяться форми з випускниками, для перегляду їх даних. На кожній формі містяться дві кнопки, щоб переглянути коротку версію даних, та більш обширну, які охоплюють такі дані, як: Прізвище, Ім'я, По батькові, email, стать, рік випуску з бакалаврату, магістратури та ОКР Спеціаліст. Компанії в яких працював випускник, позиція яку він займав, роки початку за закінчення роботи в компанії, дата створення анкети випускника, та дата останнього оновлення.

#### Ф6. Статистика

Містяться дві діаграми з даними про компанії в яких працюють випускники та про технології, які використовують випускники.

#### Ф7. Кабінет користувача

Міститься форма де зберігається детальна інформація про користувача який увійшов в систему.

#### Ф8. Форма реєстрації

Обов'язкові поля для реєстрації:

- Username
- Email
- Пароль
- Підтвердження пароля

#### Ф9. Контакти

Міститься форма де зберігається детальна інформація про автора веб-додатку та керівника дипломної роботи.

#### Ф10. Про сайт

Інформація про ліцензію та пошта розробника.

#### Ф11. Вміст

Вміст сторінки.

#### Ф12. Посилання

Посилання на інформаційні сторінки кафедри: Instagram, Facebook, Web Site.

## 2.5. Дизайн Веб-сайту

При розробці Веб-сайту Виконавець використовує надану Замовником корпоративну символіку, а також корпоративні кольори, шрифти та інші графічні елементи корпоративного стилю Замовника. Фото та інші графічні матеріали для створення дизайну та оформлення Веб-сайту надаються Замовником, якщо не обумовлено інше.

Вимоги Замовника до зовнішнього вигляду Веб-сайту включають:

- Дизайн необхідно виконати в спокійній кольоровій гамі.
- Дизайн повинен бути виконаний в стилі мінімалізму (нічого зайвого, відволікаючого увагу).
- Види і розміри шрифтів повинні бути комфортними для перегляду користувачами.
- Кнопки меню повинні бути чіткими, має бути присутнім чіткий розділ між пунктами меню.
- Дизайн сайту має коректно відображатись в наступних основних браузерах: IE 8.0 і вище, останні версії Mozilla Firefox, Opera, Chrome, а також в основних мобільних браузерах.

Виконавець створює початковий статичний макет дизайну головної сторінки Веб-сайту з використанням елементів фірмового стилю і вимог Замовника. Припускається внесення до 5 (п'яти) правок до обраного Замовником варіанту дизайну, якщо такі зміни не тягнуть за собою кардинального перегляду загального стилю сайту. На основі затвердженого Замовником макету дизайну головної сторінки розробляються макети внутрішніх сторінок сайту.

Правки дизайну виконуються лише на стадії макетування та включають в себе такі зміни, як заміну фото, шрифту, заголовку, кольору окремого елемента дизайну, незначну зміну яскравості та / або насиченості кольору тла сайту і т.п. Претензії по невідповідності макету особистим естетичним очікуванням, які тягнуть за собою повну переробку макету, за умови виконання Технічного

Завдання, не приймаються. В такому разі Замовник може замовити створення додаткових макетів дизайну.

Будь-які правки до дизайну, які не вказані у Технічному Завданні, не відповідають або суперечать йому, вносяться за умови створення і підписання окремого протоколу.

## **2.6. Інформаційне наповнення Веб-сайту**

Модифікація вмісту розділів повинна здійснюватися за допомогою адміністраторського веб-інтерфейсу (системи управління сайтом), який без застосування спеціальних навичок програмування (без використання програмування і спеціального кодування або форматування) повинен передбачати можливість редагування інформаційного вмісту сторінок сайту.

Початкове інформаційне наповнення розділів Ф1-Ф12 визначених п. 3 Технічного Завдання, а також пошукова оптимізація метатегів title / description згідно списку ключових слів, наданих Замовником здійснюється Виконавцем через систему керування контентом. Всі матеріали в електронному форматі та готовому до розміщення надаються Замовником. Інші розділи, визначені в п. 2.3 Технічного Завдання заповнюються Замовником.

## **2.7. Панель адміністрування (CMS)**

Система управління контентом (адміністративна частина сайту) повинна надавати можливість додавання, редагування та видалення вмісту статичних та динамічних сторінок.

Система управління контентом повинна мати стандартний для програм інтерфейс, який відповідає таким вимогам:

- реалізація в графічному віконному режимі;
- єдиний стиль оформлення;
- інтуїтивно зрозуміле, зручне призначення елементів інтерфейсу;

- відображення на екрані тільки тих можливостей, які доступні конкретному користувачу;
- для операцій з масового введення інформації повинна бути передбачена мінімізація кількості натискань на клавіатуру для виконання стандартних дій.

Таблиця 2.1. Вимоги до технічного забезпечення системи

№ п/п	Основні характеристики комп'ютера
<b>Рекомендоване технічне забезпечення для сервера</b>	
1	Процесор: чотирьох-ядерний 2,5 Ghz\RAM: 4098 мб\SSD 500Gb\ LAN 1 Gbit\Операційна система на базі ядра Linux
<b>Рекомендоване технічне забезпечення для клієнта</b>	
1	Процесор: одно-ядерний 1,5 GHz; RAM: 2048 Mb; HDD: 250 Gb;
2	Монітор: Будь-який
3	Миша: USB
4	Клавіатура: USB
5	Браузер: Google Chrome, FireFox, Opera, Microsoft Edge

## 2.8. Процес розробки

Процес розробки Веб-сайту складається з послідовного виконання наступних етапів:

- Одержання й аналіз загальної інформації від Замовника.
- Розробка Технічного Завдання і затвердження його Замовником.
- Розробка системи навігації, архітектури і логіки Веб-сайту.
- Створення макету дизайну головної сторінки, затвердження дизайну Замовником.
- Створення дизайну внутрішніх сторінок на основі затвердженого макету головної сторінки.
- Верстка, кодування і крос браузерне програмування Веб-сайту згідно структури визначеної в п.3 Технічного Завдання з підтримкою роботи в наступних основних браузерах: IE 8.0 і вище, останніх (на момент виконання робіт) версіях Mozilla Firefox, Opera, Chrome.
- Створення та налаштування системи керування сайтом, всіх необхідних функціональних елементів, визначених в п.3 Технічного Завдання.

- Тестування Веб-сайту в основних браузерях: ІЕ 8.0 і вище, останніх (на момент виконання робіт) версіях Mozilla Firefox, Opera, Chrome.
- Демонстрація системи керування сайтом та інструктаж з її використання.

Детальний опис етапів роботи зі створення сайту наведено в таблиці 2.2.

Таблиця 2.2. Етапи створення сайту.

№	Склад і зміст робіт	Термін розробки (в робочих днях)	Дата здачі роботи
1	Розробка технічного завдання	-	
2	Розробка шаблонів дизайну в форматі AI (Adobe illustartor) для всіх типових сторінок веб-сайту (Етап 1)	7	
3	Верстка шаблонів дизайну в Angular Material Design, розробка таблиці стилів CSS (Етап 2)	4	
4	Розробка основного функціоналу сайту, описаного в ТЗ на мові програмування Java з допомогою фреймворку Spring Boot. (Етап 3)	12	
5	Тестування на помилки (Етап 4)	5	
	<b>Загальна тривалість надання послуг (з урахуванням резервного терміну на налагодження і виправлення помилок) та термін завершення розробки сайту</b>	<b>28</b>	<b>Не пізніше 27.05.21</b>

## **2.9. Технічна підтримка Веб-сайту**

Протягом двох років з моменту завершення робіт з розробки Веб-сайту Виконавець надаватиме Замовнику послугу технічної підтримки, забезпечуючи роботу Веб-сайту у веб-мережі ( вартість розробки сайту включає вартість технічного супроводу протягом зазначеного періоду), адміністрування та внесення незначних змін протягом 1-го року з дати розробки сайту .

Здача-приймання виконаних робіт повинна здійснюватися при пред'явленні Виконавцем комплектів відповідних документів і завершуватися оформленням акту здачі-приймання, підписаного Виконавцем та затвердженого Замовником.

Випробування сайту проводяться силами Виконавця і Замовника.

Приймання сайту повинно проводитися приймальною комісією, до складу якої мають входити представники Замовника та Виконавця, протягом місяця після завершення робіт. Після введення сайту в промислову експлуатацію протягом місяця представники Замовника відстежують працездатність сайту на предмет помилок, представники Виконавця повинні виправити помилки знайдені Замовником. Результати роботи комісії повинні оформлятися актом, підписаним членами комісії та затвердженим Замовником.

## Розділ 3. Опис комплексу задач автоматизації.

### 3.1. Обґрунтування вибору програмних засобів розроблення системи

Для розробки системи було використано:

- мови програмування — Java, TypeScript;
- фреймворки — Spring Boot, Angular;
- середовище бази даних — PostgreSQL;
- IDE — IntelliJ IDEA, Visual Studio Code;
- візуальні компоненти — Material Design;

Java — об'єктно-орієнтована мова програмування, яка працює на основі JVM.

Spring Boot — програмний каркас для платформи Java, який полегшує створення додатків.

Spring Framework складається з кількох модулів, які надають широкий спектр послуг:

- Контейнер Інверсії управління: Конфігурація компонентів додатків і управління життєвим циклом об'єктів Java, здійснюється головним чином через Інверсію управління
- Аспектно-орієнтоване програмування: дозволяє реалізувати наскрізні процедури
- Доступ до даних: робота з реляційною системою управління базами даних на платформі Java з використанням JDBC і об'єктно-реляційні відображення та інструментів з NoSQL баз даних
- Управління транзакціями: об'єднує кілька API, управління транзакціями та координує операції для Java-об'єктів
- Модель-Вигляд-Управління (Model-View-Controller): програмний каркас на основі HTTP сервлета, що забезпечує створення веб-додатків і веб-служб RESTful.
- Аутентифікація і авторизація: налаштовувані процеси безпеки, які підтримують цілий ряд стандартів, протоколів, інструментів і практик за

допомогою підпроєкту Spring Security (колишня система безпеки Acegi для Spring).

- Віддалене керування: конфігураційний вплив і управління Java-об'єктами для місцевої (локальної) або віддаленої конфігурації через JMX.
- Тестування: підтримка класів для написання юніт-тестів та інтеграційних тестів.

TypeScript — мова програмування, яка розширює можливості JavaScript, використовується як основна мова для фреймворка Angular. Дозволяє створювати RESTful додатки [3].

Angular - написаний на TypeScript front-end фреймворк з відкритим кодом, який розробляється під керівництвом Angular Team у компанії Google, а також спільнотою приватних розробників та корпорацій[4].

PostgreSQL - об'єктно-реляційна система керування базами даних. Є альтернативою як комерційним СКБД, так і СКБД з відкритим кодом.

Material Design - принципи дизайну сайтів, програмного забезпечення і застосунків, а також правила дизайну інтерфейсів для операційної системи Android від компанії Google.

IntelliJ Idea - комерційне інтегроване середовище розробки для різних мов програмування від компанії JetBrains. Перша версія IntelliJ IDEA з'явилася у січні 2001 року й швидко здобула популярність, як перша Java IDE із широким набором інтегрованих інструментів для рефакторингу, що дозволяла програмістам швидко реорганізувати сирцевий код програм. Дизайн середовища орієнтовано на продуктивність праці програмістів, дозволяючи їм сконцентруватися на розробці функціональності, тоді як IntelliJ IDEA бере на себе виконання рутинних операцій.

До складу IntelliJ IDEA включені напрацювання, створені в результаті спільної роботи з компанією Google, яка використовувала IntelliJ IDEA як базис для своєї нового відкритого середовища розробки Android Studio. Завдяки співпраці істотно розширені штатні можливості IntelliJ IDEA з розробки застосунків для платформи Android.

Visual Studio Code — засіб для створення, редагування та зневадження сучасних вебзастосунків і програм для хмарних систем.

### 3.2. Інформаційне забезпечення системи.

В середовищі ERWin було створено логічну модель бази даних.

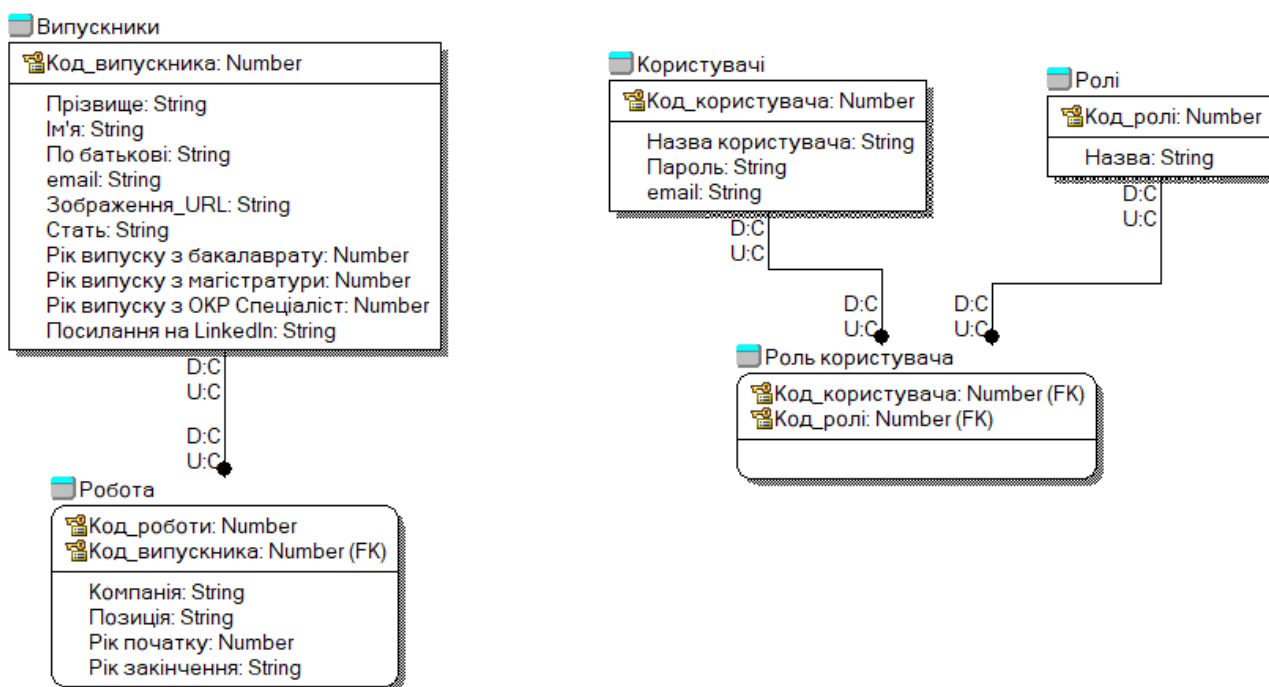


Рисунок 3.1 – Логічна модель бази даних.

Також було побудовано фізичну модель бази даних.

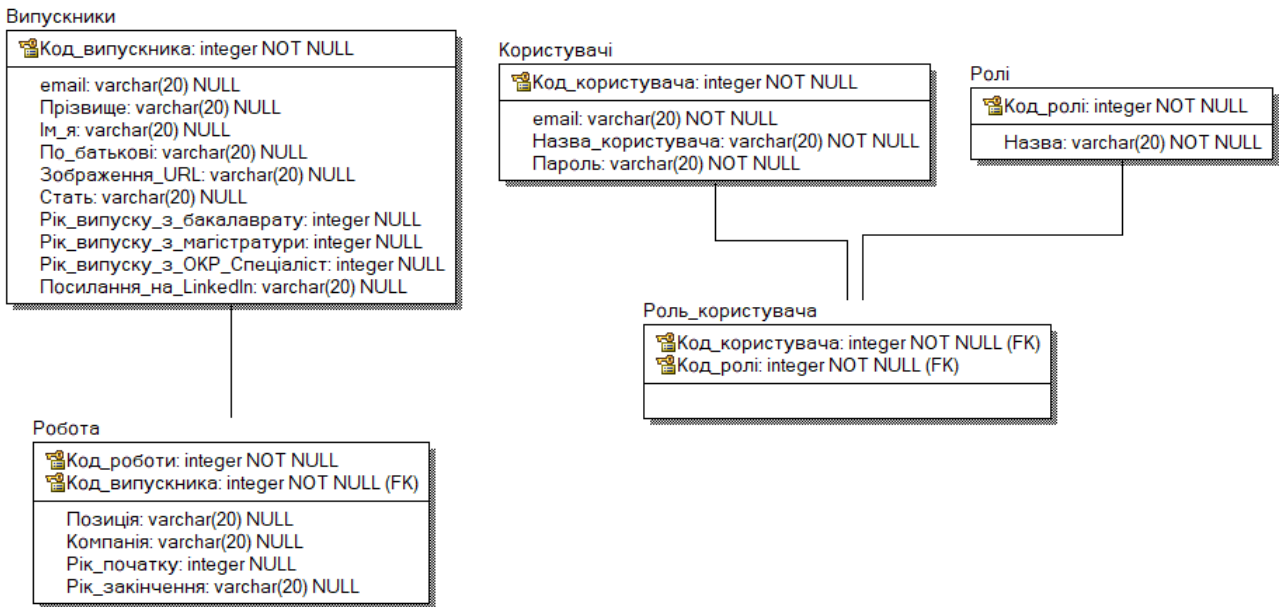


Рисунок 3.2. – Фізична модель бази даних.

Опис логічної, фізичної моделей та структури БД з посиланням на додатки.

Для створення бази даних в середовищі PostgreSQL була використана технологія Hibernate, яка дозволяє швидко створювати таблиці та зв'язки між ними.

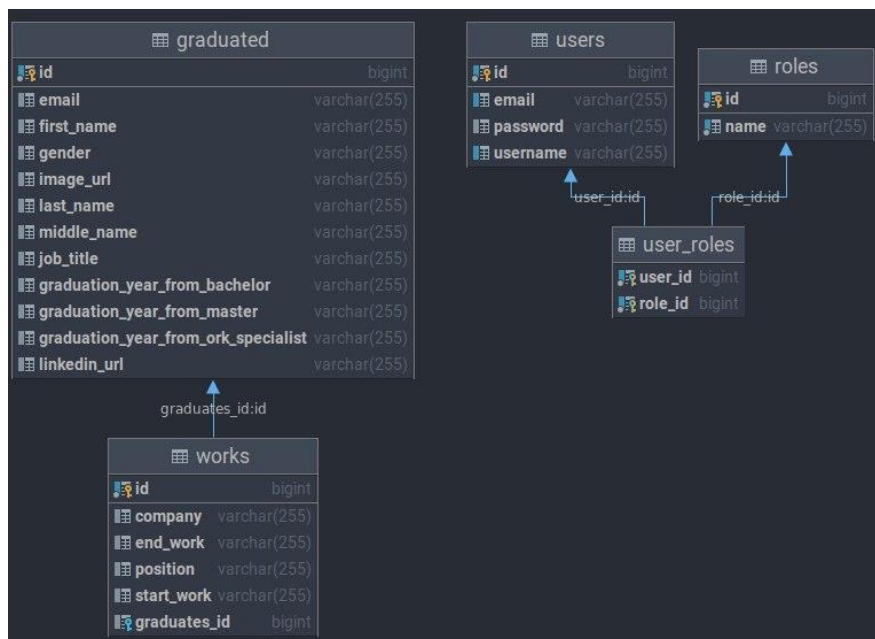
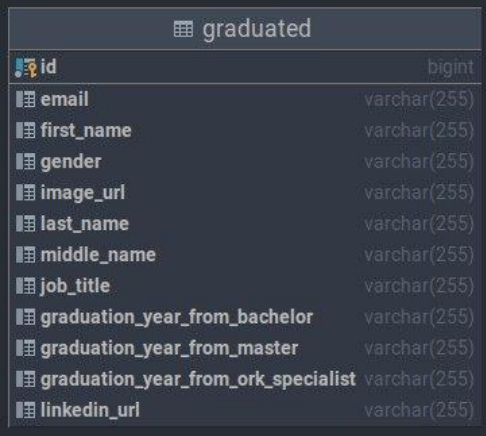


Рисунок 3.3. - Структура БД.

Загалом було створено 5 таблиць, такі як: Випускники, Робота, Ролі, Користувачі та підтаблиця з розподіленням ролей.

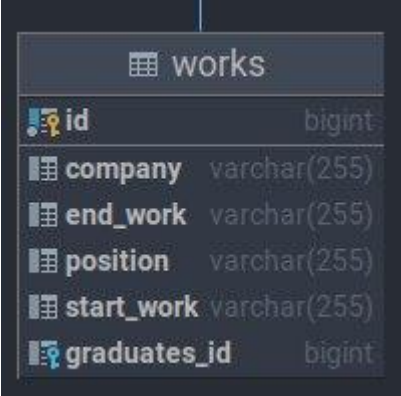
Таблиця Випускники має такий вигляд:



graduated	
id	bigint
email	varchar(255)
first_name	varchar(255)
gender	varchar(255)
image_url	varchar(255)
last_name	varchar(255)
middle_name	varchar(255)
job_title	varchar(255)
graduation_year_from_bachelor	varchar(255)
graduation_year_from_master	varchar(255)
graduation_year_from_ork_specialist	varchar(255)
linkedin_url	varchar(255)

Рисунок 3.4 – Таблиця Випускники.

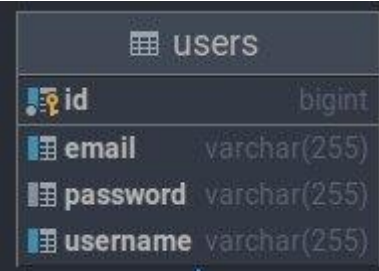
Таблиця Робота:



works	
id	bigint
company	varchar(255)
end_work	varchar(255)
position	varchar(255)
start_work	varchar(255)
graduates_id	bigint

Рисунок 3.5 – Таблиця Робота.

Таблиця Користувачі:



users	
id	bigint
email	varchar(255)
password	varchar(255)
username	varchar(255)

Рисунок 3.6 – Таблиця Користувачі.

Таблиця Ролі:

roles	
id	bigint
name	varchar(255)

Рисунок 3.7 – Таблиця Ролі.

Таблиця Розподіл по ролях

user_roles	
user_id	bigint
role_id	bigint

Рисунок 3.8 – Таблиця розподіл по ролях

### 3.3. Алгоритмізація та реалізація комплексу задач автоматизації.

Програмний продукт був реалізований за допомогою архітектури мережеских протоколів REST API, яка дозволяє зручно передавати дані з сервера користувачеві.

Передача даних відбувалась через контролери, які дозволяють зберігати, редагувати, видаляти та переглядати дані.

Для того щоб створити випусника був використаний такий контролер:

```
@PostMapping("/add")
public ResponseEntity<Graduated> addGraduated(@RequestBody Graduated graduated){
    Graduated newGraduated = graduatedService.addGraduated(graduated);
    return new ResponseEntity<>(newGraduated, HttpStatus.CREATED);
}
```

Рисунок 3.9 – Контролер для створення випусника.

За розподіл користувачів по ролях відповідає такий фреймворк, як Spring Security. Також він дає можливість убезпечити сайт від різновидних атак та dependency injection.

Код програми, який захищає додаток від сторонніх атак, та встановлює дані які може бачити користувач який не ввійшов в систему:

```

@Override
protected void configure(HttpSecurity http) throws Exception {
    http.cors().and().csrf().disable() HttpSecurity
        .exceptionHandling().authenticationEntryPoint(unauthorizedHandler).and()
        .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS).and()
        .authorizeRequests().antMatchers(...antPatterns: "/api/auth/**").permitAll() ExpressionUrlAuthorizationConfigurer
        .antMatchers(...antPatterns: "/api/test/**").permitAll()
        .antMatchers(...antPatterns: "/graduates/**").permitAll()
        .antMatchers(...antPatterns: "/api/**").permitAll()
        .anyRequest().authenticated();

    http.addFilterBefore(authenticationJwtTokenFilter(), UsernamePasswordAuthenticationFilter.class);
}

```

Рисунок 3.10 – Конфігурація Spring Security.

Також нам потрібно реалізувати шифрування паролів в базі даних, так як їх потрібно убезпечити від сторонніх осіб. Для цього ми використовуємо Spring Security, та JWT-токен, який шифрує паролі за вказаним секретним ключем.

Приклад шифрування за допомогою JWT-токена наведений в Рис.3.5.2.2.

```

public String generateJwtToken(Authentication authentication) {

    UserDetailsImpl userPrincipal = (UserDetailsImpl) authentication.getPrincipal();

    return Jwts.builder()
        .setSubject((userPrincipal.getUsername()))
        .setIssuedAt(new Date())
        .setExpiration(new Date((new Date()).getTime() + jwtExpirationMs))
        .signWith(SignatureAlgorithm.HS256, jwtSecret)
        .compact();
}

```

Рисунок 3.11. – Шифрування паролів за допомогою JWT-токена.

Щоб переглянути результат шифрування, заїдемо в БД та переглянемо збережені паролі користувачів.

id	email	password	username
1	6 roma.sulimak@gmail.com	\$2a\$10\$tngSijX4GT6PppiII3M0zus5Tv4JJWhz34GU8UW196jx...	admin
2	7 admin@gmail.com	\$2a\$10\$78mrgELYFA7itG7nLjo6meyZWq93oTrv0RHcPu22exmE...	admin2
3	8 user@gmail.com	\$2a\$10\$cIdqd632I72VSAfCa07uvuM4at33lnS5p00hzjo1Yk4k...	user
4	10 moder@gmail.com	\$2a\$10\$I0heod0wjXGmZwU/I3kA/e0b.TxuCvM4QJW05TIqLAQS...	moder

Рисунок 3.12. - Результат шифрування паролів.

Для роботи з базою даних мені потрібно під'єднати її до проекту, це здійснювалось за допомогою підключення конфігурації до Spring Boot.

```
spring.datasource.platform=postgres
spring.datasource.url=jdbc:postgresql://localhost:5432/registergraduatesRest
spring.datasource.username=postgres
spring.datasource.password=root
```

Рисунок 3.13. – Підключення бази даних до проєкту.

Щоб створити таблиці я скористався можливостями Hibernate, та створив таблиці за допомогою Java-коду. Приклад створення таблиці наведений в Рис.3.3.5.

```
@EqualsAndHashCode
@ToString
@Data
@Entity
@Table(name = "works")
public class Work implements Serializable {
    @Id
    @Column(nullable = false, updatable = false)
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String position;
    private String company;
    private String startWork;
    private String endWork;

    @JsonBackReference
    @ManyToOne
    @JoinColumn(name = "graduates_id")
    private Graduated graduated;
```

Рисунок 3.14. – Створення таблиці в БД за допомогою Hibernate.

Для роботи з базою даних я використовував Spring Data JPA, яка дає можливість за допомогою назви методу генерувати запити до бази даних.

```

@Repository
public interface UserRepository extends JpaRepository<User, Long> {
    Optional<User> findById(Long id);
    Optional<User> findByUsername(String username);
    Boolean existsByUsername(String username);
    Boolean existsByEmail(String email);
    User getUserByUsername(String username);
}

```

Рисунок 3.15. – Приклад генерації запитів до БД.

Для того, щоб запити працювали правильно, необхідно оголосити анотацію Repository, та успадкувати інтерфейс JpaRepository.

Для обробки даних зі сторони клієнта використовується фреймворк Angular.

Щоб отримати дані з сервера використовуємо такі контролери, які працюють на окремому порті:

```

@Injectable({providedIn: 'root'})
export class WorkService{
    private apiUrl=environment.apiUrl;

    constructor(private http: HttpClient){}

    public getWorks(): Observable<Work[]>{
        return this.http.get<Work[]>(`${this.apiUrl}/api/works/all`);
    }

    public addWork(graduateId: number, work: Work): Observable<Work>{
        return this.http.post<Work>(`${this.apiUrl}/api/graduates/${graduateId}/add`, work);
    }

    public updateWork(graduateId: number, work: number, value: Work): Observable<Work>{
        return this.http.put<Work>(`${this.apiUrl}/api/graduates/${graduateId}/works/${work}`, value);
    }

    public deleteWork(workId: number): Observable<void>{
        return this.http.delete<void>(`${this.apiUrl}/api/works/delete/${workId}`);
    }

    public findWork(id: number): Observable<void>{
        return this.http.delete<void>(`${this.apiUrl}/api/graduates/${graduateId}/delete/${id}`);
    }
}

```

Рисунок 3.16. – Обробка даних зі сторони користувача.

Для ілюстрації даних користувачу використовувалась мова розмітки HTML та графічний компонент Material Design.

```
<mat-tab-group>
  <mat-tab label="Company charts">
    <div>
      <div>
        <div class="chart">
          <canvas baseChart
            [data]="pieChartData"
            [labels]="pieChartLabels"
            [chartType]="pieChartType"
            [options]="pieChartOptions"
            [plugins]="pieChartPlugins"
            [colors]="pieChartColors"
            [legend]="pieChartLegend">
          </canvas>
        </div>
      </div>
    </div>
  </mat-tab>
  <div>
    <button mat-button mat-raised-button color="primary" (click)="pieChartLegend=!pieChartLegend">Toggle Legend</button>
    <button mat-button mat-raised-button color="primary" (click)="changeLegendPosition()" style="margin-bottom: 20px;">Change Legend Position</button>
  </div>
</mat-tab>
```

Рисунок 3.17. – Виведення діаграми на web-сторінку.

### 3.4. Інструкція користувача.

При завантаженні сторінки користувач потрапляє на головну сторінку, на якій міститься інформація про функції системи, та меню, на якому містяться кнопки для навігації.

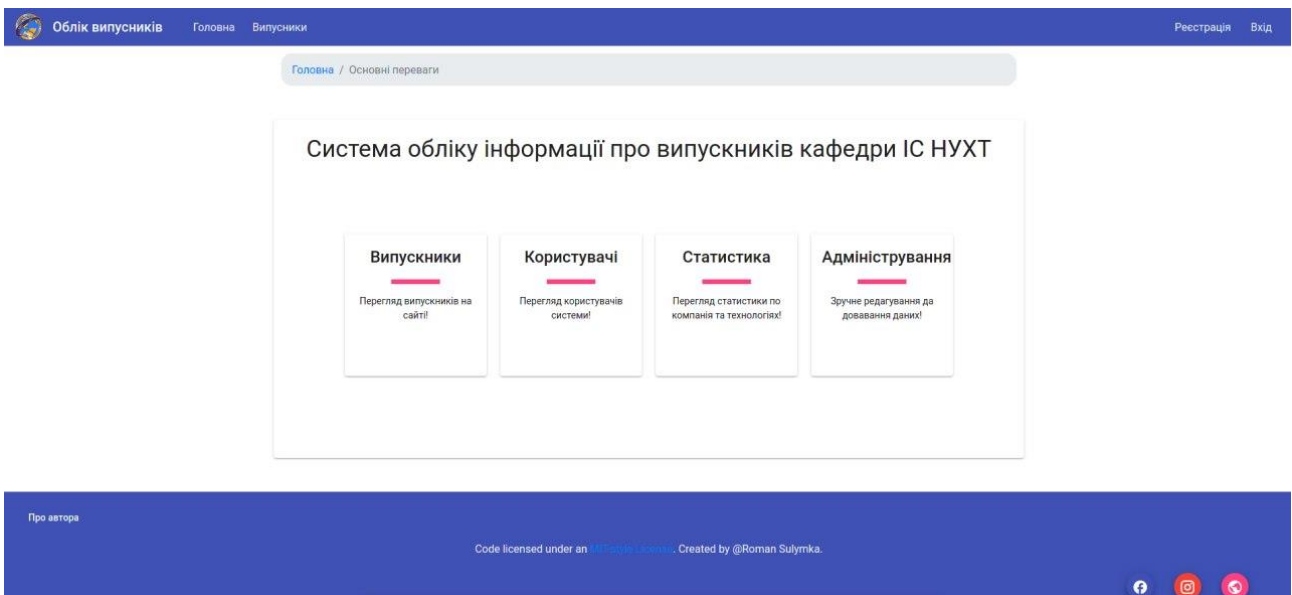


Рисунок 3.18. – Головна сторінка.

Для того, щоб незареєстровані користувачі могли побачити дані про випусників, на навігаційній панелі розміщена кнопка «Випусник», вона дозволяє переглянути список та інформацію про випусників, які занесені в систему.

Для створення акаунту необхідно натиснути кнопку «Реєстрація», та ввести необхідні дані.

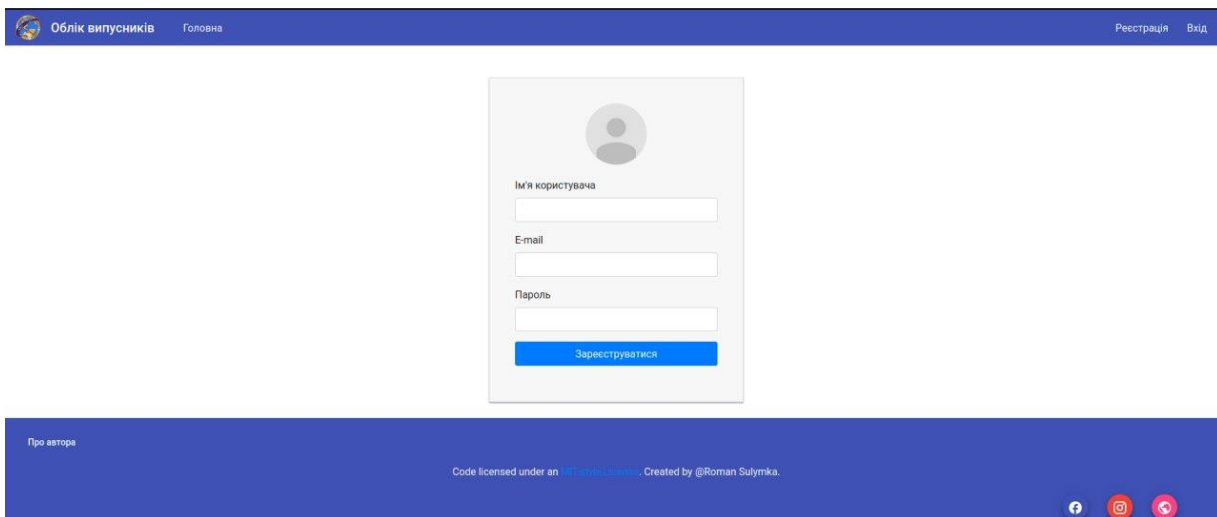


Рисунок 3.19. - Сторінка реєстрації.

Для входу в систему потрібно натиснути кнопку «Login» та ввести ім'я користувача і пароль для входу в систему.

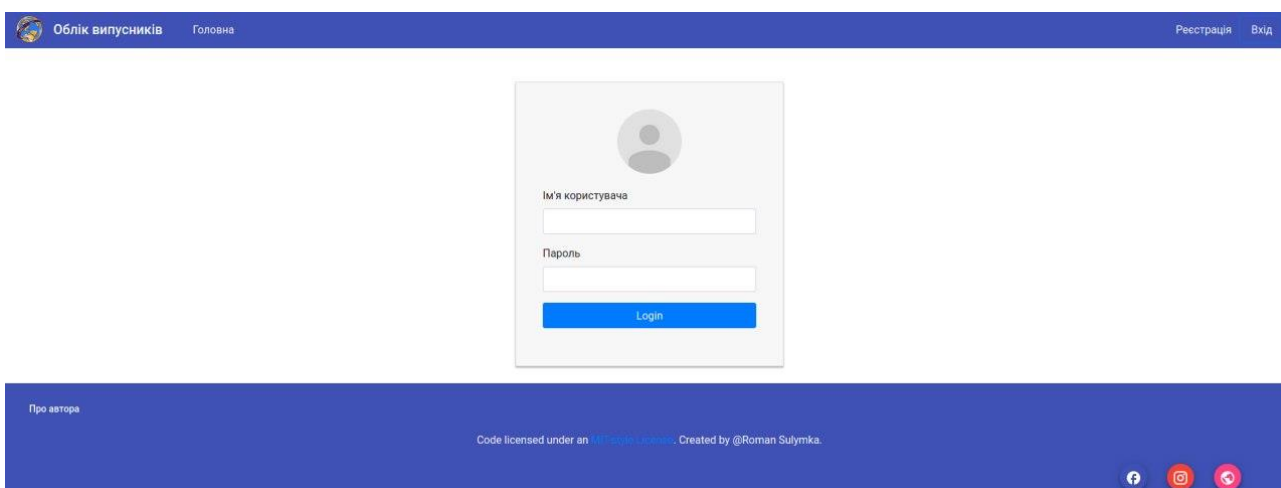


Рисунок 3.20. - Сторінка авторизації.

При введенні даних потрібно внести більше 6 символів в поле «пароль», інакше буде виведена помилка.

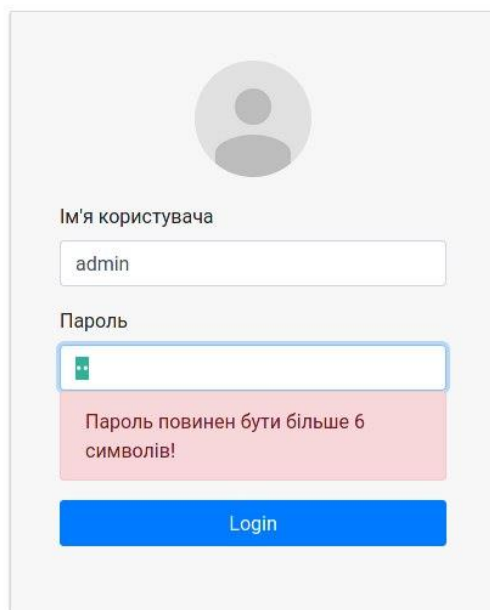


Рисунок 3.21. – Обробка виключень.

Після авторизації користувачам які авторизувались під роллю User будуть доступні такі вкладки: «Головна», «Користувачі», «Випускники», «Статистика».

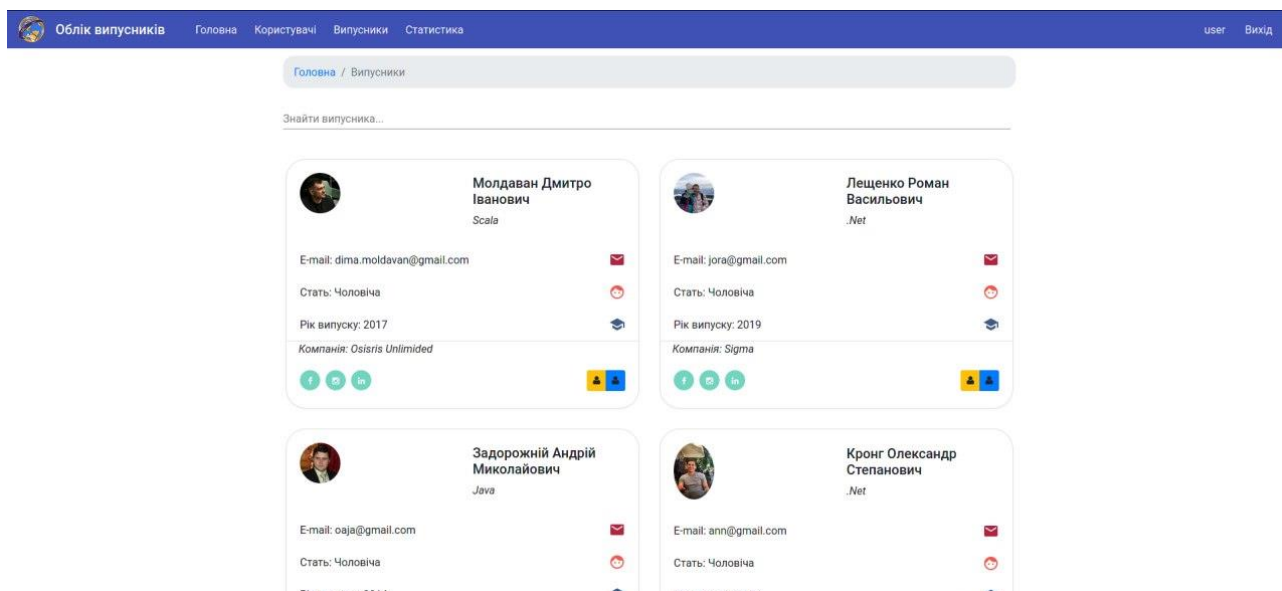


Рисунок 3.22. – Вигляд сайту зі сторони користувача.

Після авторизації користувач попадає в свій особистий кабінет, де зазначена електронна скринька користувача та його роль.

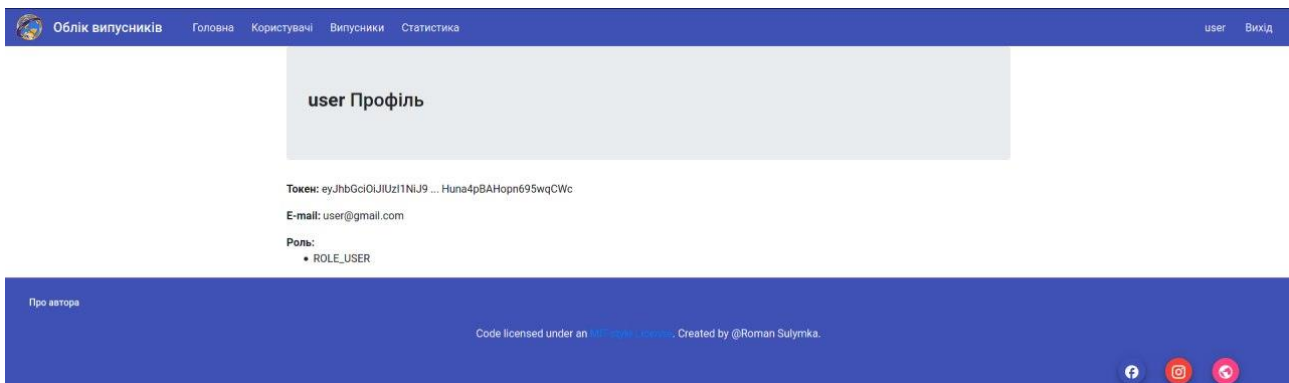


Рисунок 3.23. – Особистий кабінет.

Для зареєстрованих користувачів є можливість переглянути всіх користувачів системи, для цього потрібно натиснути в меню «Користувачі»

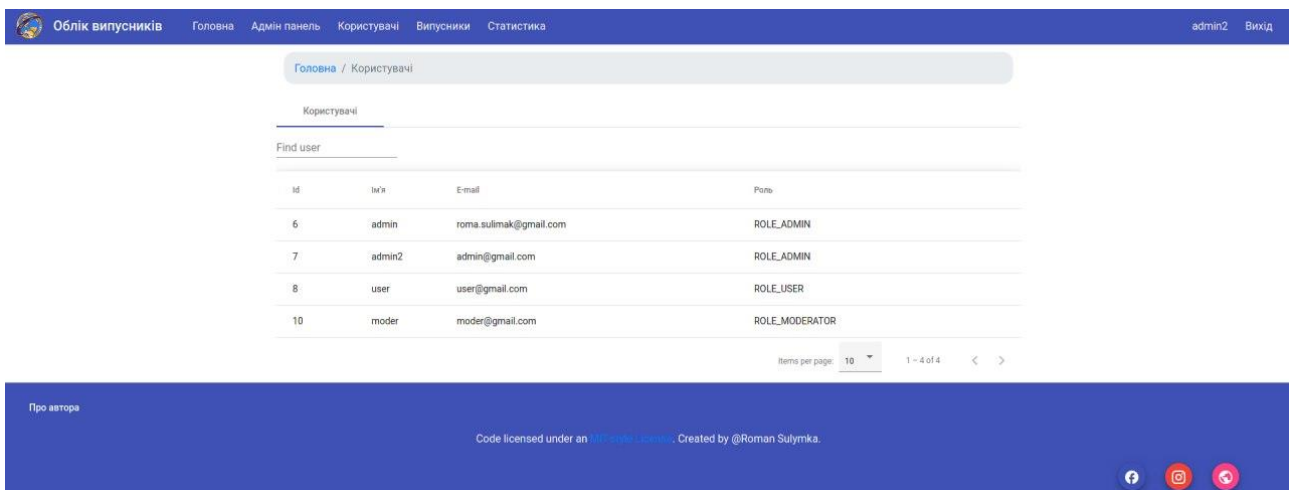


Рисунок 3.24. - Сторінка користувачі.

Для перегляду випускників необхідно натиснути в меню «Випускники», після цього відкриється зазначена сторінка.

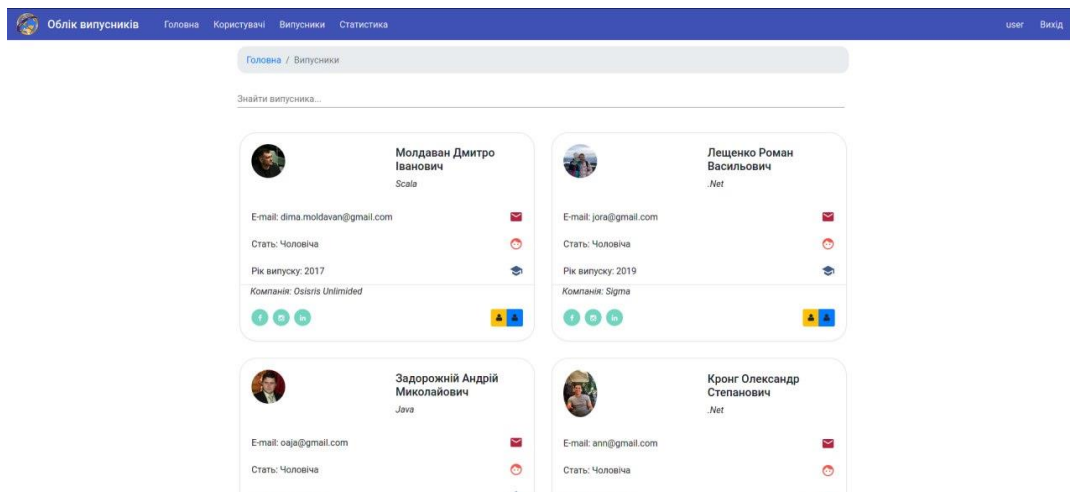


Рисунок 3.25. - Сторінка випускники.

Для перегляду детальнішої інформації необхідно натиснути ці кнопки:

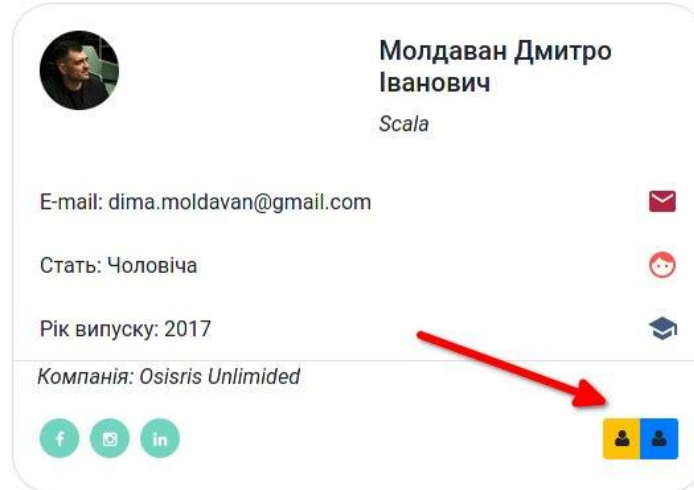


Рисунок 3.26. – Кнопки для перегляду інформації про випусника.

Після натиску першої кнопки відкриється коротка інформація про даного випусника.

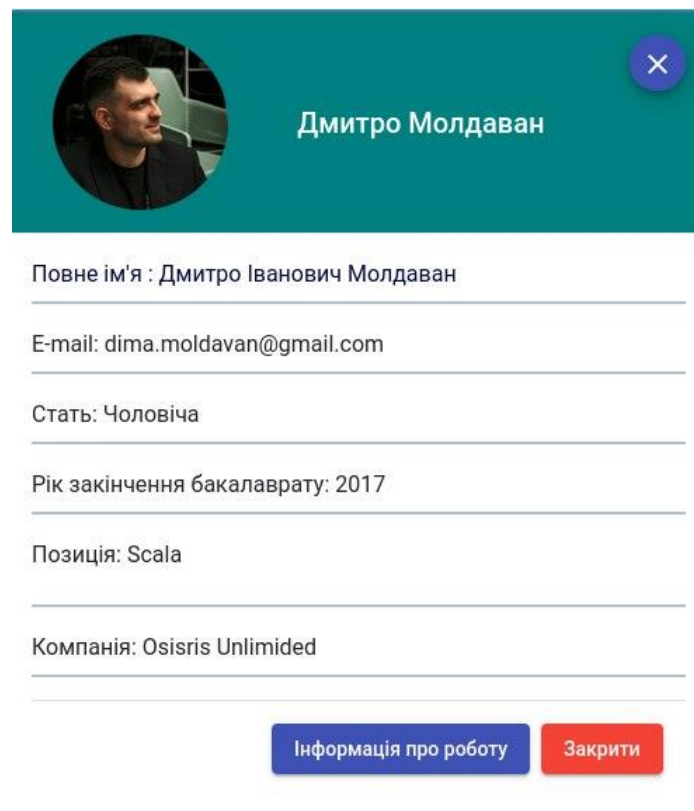


Рисунок 3.27. – Вікно перегляду короткої версії інформації про випусника.

При натиску на наступну кнопку можна переглянути історію роботи випусника, та дату останнього оновлення даних.

**Дмитро Молдаван**

Повне ім'я: Дмитро Іванович Молдаван

E-mail: dima.moldavan@gmail.com

Стать: Чоловіча

Рік випуску з бакалаврату: 2017

Теперішня позиція: Scala

Теперішня компанія: Osisris Unlimided

Анкету створено:

Останнє оновлення: 2021-05-31

Досвід роботи:

Компанія	Позиція	Рік початку	Рік закінчення
Osisris Unlimided	Java	2016	2018
Osisris Unlimided	Scala	2018	-

Закрити

Рисунок 3.28. – Вікно перегляду даних випускників включаючи історію роботи.

При натиску на наступний пункт меню з'явиться сторінка з статистикою.

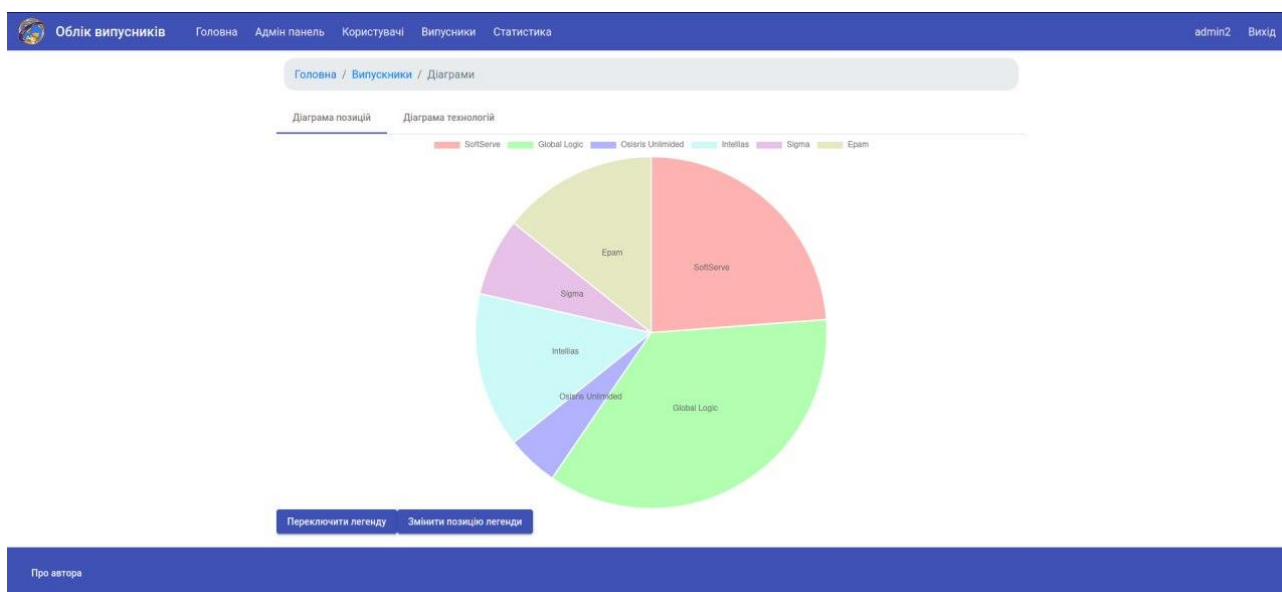


Рисунок 3.29. - Сторінка з статистикою.

На цій сторінці доступні дві вкладки: «Статистика компаній» в яких працювали випускники» та «Статистика технологій», які використовують випускники». Також на них знаходяться дві кнопки: «Переключити легенду» та «Змінити позицію легенди».

Також в програмі присутня інформація про розробника, та наукового керівника

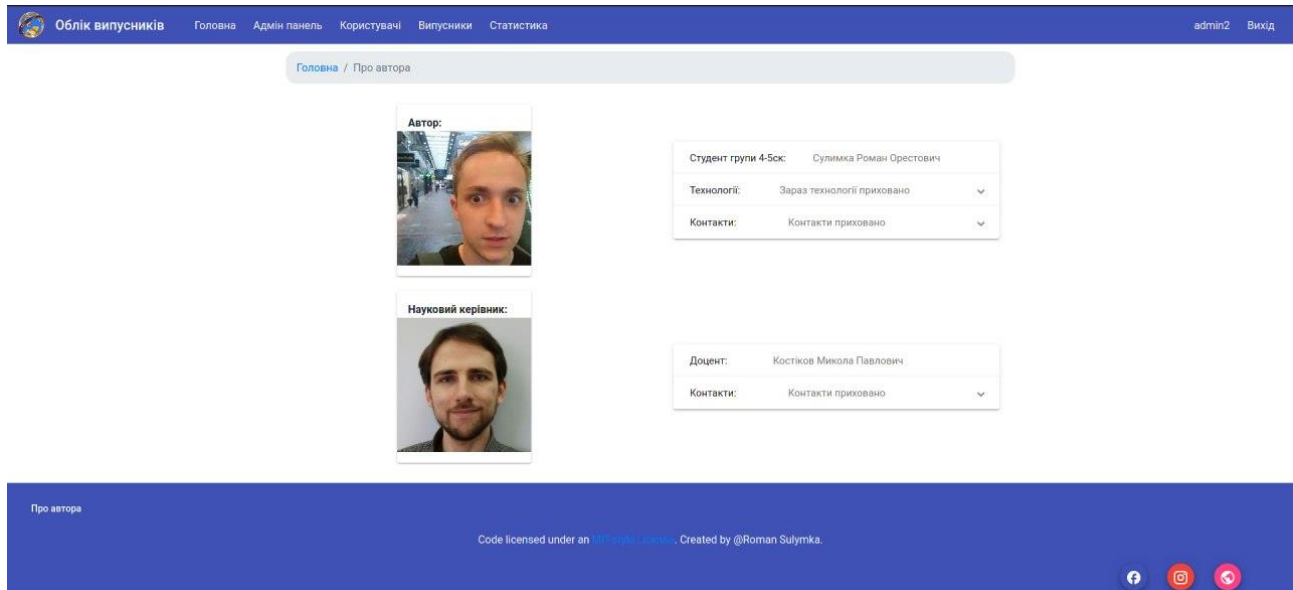


Рисунок 3.32. – Сторінка «Про автора».

Адміністраторам доступно більше можливостей, в них з’являється пункт меню «Адмін панель», за допомогою якого можна керувати обліковими записами, створювати анкету випускника, редагувати збережені дані, та видаляти їх.

При переході на сторінку «Адмін панель» адміністратору доступно 3 вкладки: «Користувачі», «Випускники», «Статистика».

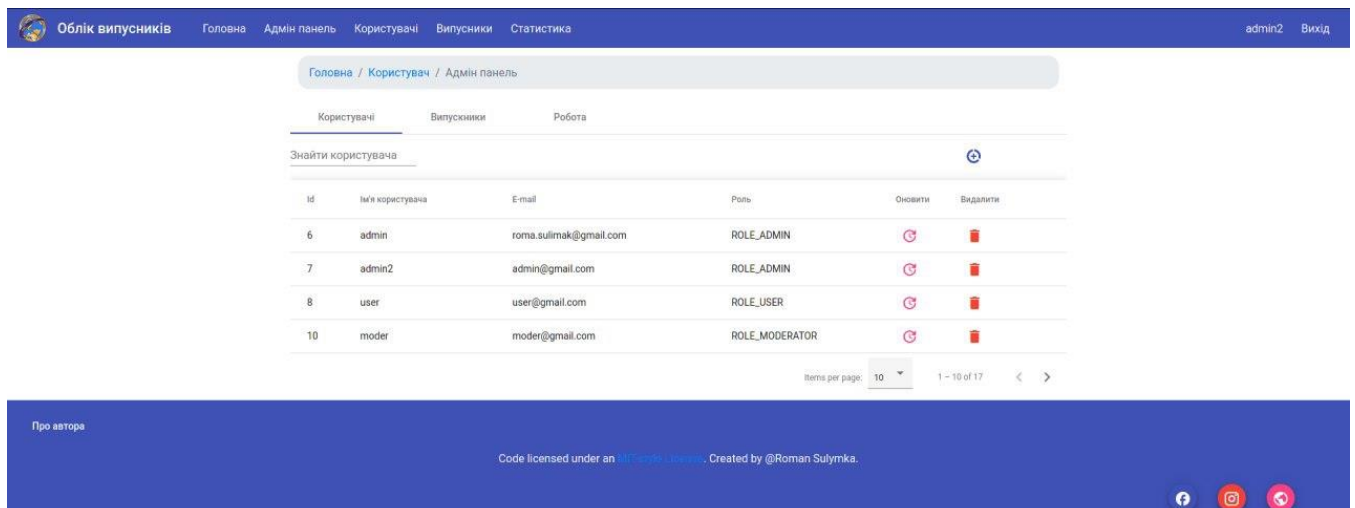


Рисунок 3.34. - Сторінка адмін-панель.

На кожній таблиці розташовані кнопки для керування даними.

Головна / Користувач / Адмін панель

Користувачі    Випускники    Робота

Знайти користувача +

Id ↑	Ім'я користувача	E-mail	Роль	Оновити	Видалити
6	admin	roma.sulimak@gmail.com	ROLE_ADMIN		
7	admin2	admin@gmail.com	ROLE_ADMIN		
8	user	user@gmail.com	ROLE_USER		
10	moder	moder@gmail.com	ROLE_MODERATOR		

Items per page: 10    1 – 10 of 17    < >

Рисунок 3.35. – Список користувачів системи.

При переході на сторінку «Випускники» відображається коротка інформація про них:

Головна / Користувач / Адмін панель

Користувачі    **Випускники**    Робота

Знайти випускника +

Id	Ім'я	По батькові	Прізвище	E-mail	Компанія	Стать	Додати роботу	Оновити	Видалити
3	Роман	Васильович	Лещенко	jora@gmail.com	<i>Sigma</i>	Чоловіча			
4	Святослав	Володимирович	Костенко	ivan@gmail.com	<i>Soft Serve</i>	Чоловіча			
5	Вадим	Олегович	Брацький	roma.sulimak@gmail.com	<i>NeoLab</i>	Чоловіча			
6	Дмитро	Іванович	Молдаван	dima.moldavan@gmail.com	<i>Osisris Unlimided</i>	Чоловіча			
7	Антон	Васильович	Богомол	d@gmail.com	<i>Global Logic</i>	Чоловіча			
8	Ольга	Орестівна	Сулимка	olya.sulimak@gmail.com	<i>Soft Serve</i>	Жіноча			
9	Олександр	Степанович	Кронг	ann@gmail.com	<i>Peek Logic</i>	Чоловіча			
10	Руслан	Олександрович	Смик	dii@gmail.com	<i>Global Logic</i>	Чоловіча			
11	Андрій	Миколайович	Задорожній	oaja@gmail.com	<i>Soft Serve</i>	Чоловіча			
12	Дар'я	Володимирівна	Несвітайло	isers@gmail.com	<i>Amazon</i>	Жіноча			

Items per page: 10    0 of 0    < >

Рисунок 3.36. – Список випускників.

Тут розміщено такі кнопки, як: «Додати випускника», «Додати роботу», «Оновити дані», «Видалити запис».

При натисканні кнопки «Видалити» з'являється діалогове вікно з підтвердженням

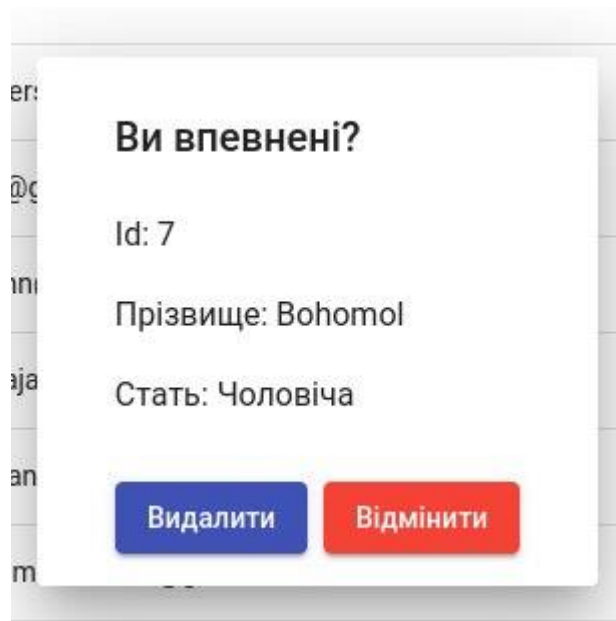


Рисунок 3.37.– Вікно підтвердження видалення випусника.

При переході на таблицю «Робота» ми бачимо всі місця працевлаштування випусника. Адміністратор має змогу редагувати ці записи або видалити їх.

Головна / Користувач / Адмін панель

Користувачі    Випусники    **Робота**

Знайти роботу

Id	Позиція	Компанія	Рік початку	Рік закінчення	Оновити	Видалити
12	DevOps	Intellias	2007	2010		
13	DevOps	Global Logic	2007	2010		
3	.Net	Soft Serve	2007	2010		
7	Java	Osisris Unlimided	2016	2018		
1	.Net	Sigma	2018	-		
14	Scala	Netflix	2019	-		
9	Scala	SoftServe	2014	-		
18	Java Middle	Soft Serve	2015	-		
4	.Net	Global Logic	2007	2010		
5	Scala	Soft Serve	2019	-		

Items per page: 10    0 of 0    < >

Рисунок 3.38. – Список робота випусників.

При оновленні з'явиться наступне діалогове вікно:



Оновити роботу: Intellias, Позиція: DevOps

Id  
12

Компанія \*  
Intellias

Позиція \*  
DevOps

Рік початку \*  
2007

Рік закінчення \*  
2010

Зберегти Скасувати

Рисунок 3.39. – Вікно оновлення даних про місце роботи.

Решту сторінок ви можете побачити в додатку Г.

### **3.5. Технічне та системне забезпечення розробки.**

#### **3.5.1. Обґрунтування вибору ОС та протоколу обміну даними.**

Серверна частина інформаційної системи може функціонувати на всіх операційних системах оскільки працює на основі JVM. Але, для найкоректнішої роботи рекомендується використовувати системи на основі ядра Linux.

Додаток був реалізований на операційній системі Arch Linux, оскільки саме на цій операційній системі найзручніше працювати з програмним забезпеченням для створення крос платформних додатків.

Для обміну даними був вибраний протокол HTTP.

HTTP — це протокол передачі даних прикладного рівня, який використовується в комп'ютерних мережах.

Для запитів до сервера використовуються такі методи:

- OPTIONS - Повертає методи HTTP, які підтримуються сервером. Цей

метод може служити для визначення можливостей веб сервера.

- GET - Запитує вміст вказаного ресурсу. Запитаний ресурс може приймати параметри.
- HEAD - Аналогічний методу GET, за винятком того, що у відповіді сервера відсутнє тіло. Це корисно для витягання метаданих, заданої в заголовках відповіді, без пересилання всього вмісту.
- POST - Передає призначені для користувача дані (наприклад, з HTML-форми) заданому ресурсу. Наприклад, в блогах відвідувачі зазвичай можуть вводити свої коментарі до записів в HTML-форму, після чого вони передаються серверу методом POST, і він поміщає їх на сторінку. При цьому передані дані (у прикладі з блогами — текст коментаря) включаються в Тіло запиту (Request body). На відміну від методу GET, метод POST не вважається ідемпотентним, тобто багаторазове повторення одних і тих же запитів POST може повертати різні результати (наприклад, після кожного відправлення коментаря з'являтиметься одна копія цього коментаря).
- PUT - Завантажує вказаний ресурс на сервер.
- PATCH - Завантажує певну частину ресурсу на сервер.
- DELETE - Видаляє вказаний ресурс.
- TRACE - Повертає отриманий запит так, що клієнт може побачити, що проміжні сервери додають або змінюють в запиті.
- CONNECT - Для використання разом з проксі-серверами, які можуть динамічно перемикаються в тунельний режим SSL.

Для перегляду відповіді з сервера використовуються коди статусу:

Коди статусу:

- 1xx — інформаційний: запит прийнятий, продовжуй процес.
- 2xx — успіх: дія була успішно передана, зрозуміла, та прийнята.
- 3xx — перенаправлення: наступні дії мають бути успішно виконані для реалізації запиту.

- 4xx — помилка клієнта: запит містить синтаксичні помилки або не може бути виконаний.
- 5xx — помилка сервера: сервер не зміг виконати правильно сформований запит.

Найбільш поширені статуси:

- 200 OK — запит виконано успішно.
- 301 Moved Permanently — ресурс переміщено.
- 403 Forbidden — доступ до запитаного ресурсу заборонений.
- 404 Not Found — ресурс не знайдений.
- 503 Service Unavailable — сервіс недоступний.

### **3.5.2. Розроблення та обґрунтування стратегії адміністрування системи.**

Адмініструванням системи можуть займатись користувачі з ролями ADMIN та MODERATOR, для простих користувачів функції адміністрування системи будуть відключені.

Адміністратори можуть додавати будь які дані, також в них є можливість редагувати на видаляти вже існуючі дані.

Користувач з роллю ADMIN має доступ до перегляду, редагування та видалення всіх даних в системі.

Користувачі з роллю MODERATOR можуть редагувати дані випускників, додавати та видаляти, але відмінність від користувачів з роллю ADMIN полягає в тому, що в других присутня можливість редагувати користувачів системи.

Користувачі з роллю USER мають можливість переглядати всі дані про випускників, випускників та діаграми з статистикою, але функції редагування для них обмежені.

Користувач який не ввійшов в систему має можливість переглядати виключно дані про випускників, для більших функцій йому потрібно зареєструватися в систему.

### **3.5.3. Заходи захисту від несанкціонованого доступу до системи.**

Для захисту від несанкціонованого доступу був використаний такий компонент фреймворка Spring Boot як Spring Security.

Цей компонент дозволяє налаштувати доступ до системи, створення форми аутентифікації, додає можливість шифрування даних в БД, та надає захист від CSRF- атак.

Компонент дозволяє за допомогою методів редагувати доступ до даних інформаційної системи.

Для шифрування паролів я використав JSON Web Token. Цей стандарт токена доступу функціонує на основі JSON, для шифрування вводиться підпис, та алгоритм шифрування.

## 4. ОХОРОНА ПРАЦІ

### 4.1. Вимоги до робочого місця користувача ЕОМ.

До фізично шкідливих та небезпечних факторів належать: підвищені рівень електромагнітного, рентгенівського, ультрафіолетового та інфрачервоного випромінювання; підвищений вміст позитивних аеронів і понижений вміст негативних аеронів у повітрі робочої зони; підвищення рівня статичної зони і запиленого повітря робочої зони; підвищений рівень близькості та сліпоті; нерівність розподілу яскравості в полі зору підвищене значення напруги у електричному ланцюзі, замикання якого може відбутись через тіло людини.

Хімічно шкідливі і небезпечні компоненти.

Хімічні, шкідливі та небезпечні компоненти такі: збільшення вмісту в повітрі робочої зони вуглекислого газу, озону, аміаку, фенолу та формальдегіду.

Психофізичні шкідливі та небезпечні фактори.

Психофізіологічні шкідливі та небезпечні фактори: напруга зору і уваги; інтелектуальні, емоційне та тривале статичне навантаження; монотонність праці; велика кількість інформації, що обробляється за одиницю часу; нераціональна організація робочого місця.

Типові відчуття, які відчуває оператор ПК в кінці робочого дня: головний біль, сльозотеча в очах, біль у м'язах шії, рук та спини, свербіж тощо. Щодня переживаючи ці захворювання, вони викликають мігрень, часткову втрату зору, сколіоз, подразнення шкіри та інші побічні ефекти. Розвиток хвороби не тільки знижує працездатність, але й шкодить здоров'ю людей. На стан здоров'я оператора ПК можуть впливати і такі шкідливі чинники, як тривале незмінне положення тіла, яке викликає м'язово-скелетні порушення; постійне напруження очей; вплив радіаційного опромінення; вплив електростатичних і електромагнітних полів і ін. Тривала й інтенсивна робота за комп'ютером може бути джерелом серйозних захворювань, таких як часті фізичні вправи (ТПН), які поступово накопичуються внаслідок хвороб, перехідних при захворюваннях нервів, м'язів та сухожилів кистей.

Професійні захворювання, пов'язані з ТПН, включають:

- тендовагініт – запалення сухожиль кисті, зап'ястка, плеча;
- тендосиновіт – запалення синовіальної оболонки сухожильної основи кисті та зап'ястка;
- синдром зап'ясного каналу (СЗК) – викликається затиском серединного нерва в зап'ясному каналі. Травма, яка накопичується, викликає утворення продуктів розпаду в області зап'ясного каналу, через що спершу виникає набряк, а потім СЗК.

#### 4.2. Профілактичні заходи з охорони праці.

Слід передбачити раціональне планування робочого місця: для запобігання загального дискомфорту, зменшення втоми працівників, підвищення його продуктивності, оптимального розміщення інструментів та товарів робітників. Робоча зона повинна бути такою, аби працівник не робив зайвих рухів та не відчував дискомфорту під час виконання роботи. Також важливо мати також можливість змінити робочу позу, тобто положення корпусу, рук, ніг. Однак краще виключити або зменшити всі фізично ненормальні та нездорові стани організму.

Основні ергономічні вимоги щодо проектування робочого місця в системі "людина – техніка – виробниче середовище" наведено на рис. 4.1:

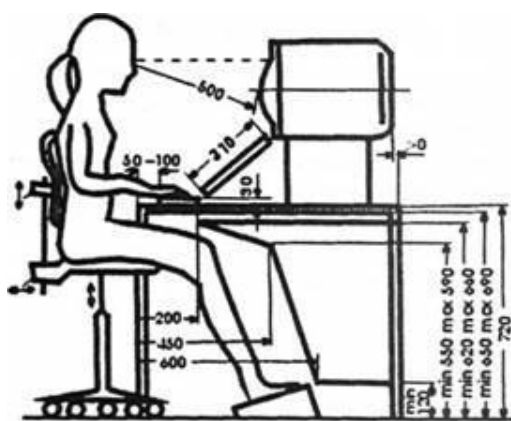


Рисунок 4.1. Робоче місце (стіл) та розташування користувача ПК.

Антропометричні вимоги. Антропологічні характеристики людини (зріст, розмір тіла та окремі рухові ланки) визначають відповідність технічних

конструкцій. Показниками є раціональна робоча поза, оптимальна зона дії, робочий рух.

Фізіологічні та психофізіологічні вимоги визначають здатність працівників бачити, обробляти інформацію, здатність приймати рішення та реалізовувати, а також адаптуватися до навколишнього середовища.

Загальні принципи організації на місці роботи:

- на робочому місці не має бути нічого зайвого. Усі предмети, необхідні для роботи, повинні бути поруч із працівником, але не заважати йому;
- предмети, що використовуються частіше, розташовуються ближче, ніж ті предмети, які використовує працівник рідше;
- предмети, взяті лівою рукою, повинні бути з лівого боку, а речі, взяті правою рукою, повинні бути з правого боку;
- якщо використовуються обидві руки, місцезрештування пристосувань обирається із урахуванням зручності захоплення їх обома руками;
- на робочому місці не повинно бути плутанини;
- організація робочого місця повинна забезпечувати необхідну видимість.

Вимоги до здоров'я визначають здатність жити і працювати в процесі взаємодії з технологіями та навколишнім середовищем; Показники включають рівень освітленості, температуру, вологість, шум, вібрацію, токсичність тощо.

Як правило, робочі місця з ПК повинні розташовуватися в окремих кімнатах. У разі робочих місць у спеціальних залах чи приміщеннях із джерелами шкідливих (шкідливих) компонентів продуктів, вони повинні розташовуватися в повністю захищених приміщеннях з природним освітленням та інтегрованим повітрообміном. Площа, на якій розташоване робоче місце з ВДТ, не повинна бути менше 6,0 м<sup>2</sup>, площа приміщення - не менше 20 м<sup>3</sup>.

Для оздоблення приміщень з ВДТ варто використовувати дифузно-віддзеркалюючі матеріали із такими коефіцієнтами відбиття:

- для стелі - 0,7-0,8;
- для стін - 0,4-0,5;

- для підлоги - 0,2-0,3.

Поверхня підлоги повинна бути рівною, неслизькою, а також легко очищатись і мати антистатичні властивості.

Заборонено використовувати полімерні матеріали, що виділяють шкідливі хімікати. Вміст шкідливих хімічних речовин у приміщеннях із ВДТ не має перевищувати концентрацію, зазначену в стандарті ГОСТ 12.1.005-88, а саме «Общие санитарно-гигиенические требования к воздуху рабочей зоны».

Високі температури повітря негативно позначається на функціональному стані людини.

Оптимальні та прийнятні мікрокліматичні параметри в приміщеннях при використанні ПК слід враховувати характеристики технічного процесу. Зокрема, технічні умови експлуатації більшості типів комп'ютерів включають прийнятні робочі діапазони параметрів мікроклімату:

- ця температура повинна бути від 10 до 40°C;
- -відносна вологість повітря може коливатися від 40 до 90 %.

За даними ВООЗ, оптимальне значення температури становить 19-23 °С, відносної вологості - 55 %, швидкість повітря не повинна перевищувати 0,1 м/с на рівні обличчя. При поверхневому нагріванні (більше 45°C), забезпечується пристрій для стискання, охолодження або ізоляції людини. Особлива увага приділяється вентиляційним отворах для запобігання перегріванню або протягам.

Згідно з діючими в нашій країні нормативних актів (ВСНиПРВЦ, СН 4088-86 "Микроклимат производственных помещений" та ГОСТ 12.1.005-88) у холодні пори року:

- температура повітря має бути 22-24°C;
- швидкість руху повітря - 0,1 м/с;
- вологість - 40-60 %/

У теплу пору року:

- температура повітря 23-25 °С;
- швидкість руху повітря - 0,1-0,2 м/с;
- відносна вологість - 40-60 %.

Температура може коливатись від 22 до 26 °С при дотриманні інших параметрів мікроклімату у вищезазначених межах.

Необхідно забезпечити можливість індивідуального регулювання розподілу повітря в приміщеннях, встановивши кондиціонер.

Процес ВДТ змінює концентрацію іонів у повітрі робочої зони. Застосування примусової вентиляції, захисних штор та іонізаторів матиме загальний вплив на іонний склад повітря в робочій зоні.

Тютюновий дим є одним із найпоширеніших забруднювачів повітря.

Робоче місце - це місце, оснащене технологією (відображення інформації, елементи керування, допоміжним обладнанням), де виконавці (або групи виконавців) ведуть свою діяльність. Вимоги до освітленості для візуального сприйняття користувачами інформації з 2 різних носіїв (з екрана ПК та паперового носія) різні. Занадто мала яскравість під час читання паперу послаблює сприйняття інформації, а занадто сильно зменшує контрастність символів на екрані. Зниження на 10 % освітленості викликає зниження працездатності на 1 %. Освітленість може варіюватися від 300 до 700 лк. Оптимальне освітлення в робочих приміщеннях для роботи з відеотерміналами становить від 300 до 500 лк.

Освітлення місця праці має бути змішаним (природним і штучним). Бажано, щоб напрямок світлового отвору для кімнати з ВДТ був на північ. Природне освітлення має здійснюватись у вигляді бічного освітлення й відповідати нормальним рівням згідно зі СНиП 11-4-79 «Естественное и искусственное освещение. Нормы проектирования».

Слід забезпечити наявність сонцезахисного крему, який зменшує перепади яскравостей між природним освітленням і яскравістю екрана комп'ютера. Необхідно використовувати плівки з металізованим покриттям або жалюзі з регульованим вертикальними розташуванням. Світлорозсіювальні штори

повинні мати коефіцієнт відбиття 0,5-0,7.

Оператор комп'ютера повинен мати робочу область, щоб вікна або освітлювальні прилади не потрапляли в поле зору. Коли штучне світло змішується з природним, рекомендується використовувати лампи зі спектральною структурою, найбільш близькою до сонячного світла.

У приміщеннях з комп'ютерами штучне освітлення експлуатується як комбінована система із використанням люмінесцентних джерел світла в звичайних освітлювальних приладах, розташованих над робочою поверхнею в рівномірно-прямокутному порядку. Вібрація люмінесцентних ламп не повинна перевищувати 10 %. Бажано розмістити вікно з одного боку робочої зони.

Співвідношення яскравості поверхні навколо екрану комп'ютера в робочій області не повинно перевищувати 3:1.

Клавіатура, екран та відеотермінал (ВДТ) з обох сторін повинні бути рівномірно освітлені на робочому місці, головним чином за рахунок розподілу відбитого або забутого світла.

Пожежі в офісах з оргтехнікою особливо небезпечні, оскільки вони пов'язані з великою кількістю фізичних та матеріальних збитків. Пожежі можуть виникати при взаємодії легкозаймистих речовин та джерел займання. До легкозаймистих матеріалів належать будівельні та оздоблювальні матеріали, пластикові корпуси, шнури тощо. Джерелами займання можуть бути електронні схеми комп'ютерів, принтерів, пристроїв живлення, де різні порушень спричиняють перегрів компонентів, утворення електричних іскор та дуг, які можуть запалити горючі речовини.

Під час робіт з технічного обслуговування, ремонту використовуються різні легкозаймисті рідини, розміщуються тимчасові електричні провідники та проводиться пайка. Виникає додаткова пожежна небезпека, що потребує відповідних заходів захисту від пожеж. До засобів пожежогасіння, які призначені для локалізації невеликих займань, відносять вогнегасники, сухий пісок, азбестові ковдри. Приміщення, у якому встановлені комп'ютери і де не потрібно встановлювати автоматичну систему пожежогасіння, має бути

обладнано переносним вуглекислотними з розрахунку по 2 шт. на кожні 20 м<sup>2</sup> в приміщеннях. Звуко поглинаюче облицччя стін, стелі приміщень повинні бути виготовлені з негорючих та важко горючих матеріалів.

Для виявлення початкової стадії займання необхідно використовувати автоматичне пожежогасіння за правилами пожежної безпеки.

Основним обладнанням на робочому місці користувача ПК є системний блок, монітор і клавіатура [7].

Робочі місця повинні знаходитись на відстані не менш ніж 1,5 м від стіни з вікнами, 1 м від інших стін, не менше 1,5 м одна від одної. Відносно вікон робоче місце доцільно розташовувати так, аби природне світло на нього падало збоку, переважно з лівої сторони.

Згідно зі статтею 18 Закону України «Про охорону праці», працівник зобов'язаний:

- а) знати та виконувати вимоги нормативних актів щодо охорони праці, правила поведіння із устаткуванням і іншими засобами виробництва, користуватись засобами колективного й індивідуального захисту [5];
- б) дотримуватись зобов'язань по охороні праці, передбачених колективним договором і правилами внутрішнього трудового розпорядку підприємства;
- в) співпрацювати із власником щодо організації безпечних та нешкідливих умов праці, а також особисто вживати посильних заходів по усуненню будь-якої виробничої ситуації, що створює загрозу його життю або здоров'ю, чи людей, які його оточують, повідомляти щодо небезпеки свого безпосереднього керівника чи іншу посадову особу [6].

## **ВИСНОВКИ**

Розробив систему для обліку випускників кафедри інформаційних систем, додаток суттєво розширює функціонал попередньої інформаційної системи, яка використовується на сайті кафедри.

Система дозволяє вести ефективний облік випускників, оскільки функціонал розширився такими корисними функціями, як:

- Перегляд детальної інформації;
- Зручність в редагуванні та додаванні даних;
- Пошук за ініціалами;
- Присутнє поле з датою останнього оновлення;
- Облік компаній та технологій випускників;

Розроблена система дозволяє всім студентам, працівникам та абітурієнтам університету ознайомитися з місцем працевлаштування випускників, також за допомогою цієї інформаційної системи можна провести аналіз найбільш затребуваних технологій на ринку, щоб грамотніше налаштувати програму навчання.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. [1] [https://uk.wikipedia.org/wiki/Національний\\_університет\\_харчових\\_технологій](https://uk.wikipedia.org/wiki/Національний_університет_харчових_технологій).
2. [2] <https://nuft.edu.ua/fakultet-aks/kafedra-nformaczjnix-sistem/>.
3. [3] <https://uk.wikipedia.org/wiki/TypeScript>
4. [4] [https://uk.wikipedia.org/wiki/Angular\\_\(фреймворк\)](https://uk.wikipedia.org/wiki/Angular_(фреймворк))
5. [5] [https://otherreferats.allbest.ru/life/00026513\\_0.html](https://otherreferats.allbest.ru/life/00026513_0.html)
6. [6] <osvita.ua/vnz/reports/bjd/23706/>
7. [7] <https://studfile.net/preview/9816410/page:5/>
8. <https://www.baeldung.com/spring-boot>
9. <https://spring.io/projects/spring-data>
10. <https://www.baeldung.com/spring-boot-angular-web>
11. «Java. Полное руководство. 10-е издание» - Герберт Шилдт, Киев: Диалектика, 1488 с.
12. «Чистий код. Створення і рефакторинг" за допомогою Agile» – Роберт Мартін, 448 с.
13. «JAVA FROM EPAM» - И.Н. Блинов, В.С. Романчик Минск: Четыре четверти, 2020. — 560 с.
14. «JAVA. МЕТОДЫ ПРОГРАММИРОВАНИЯ» - И.Н. Блинов, В.С. Романчик Минск: издательство «Четыре четверти», 2013. — 896 с.
15. TypeScript [Веб-сайт]. – Режим доступа: <https://www.typescriptlang.org>;
16. Java [Веб-сайт]. – Режим доступа: <https://www.java.com/ru/>;
17. CSS [Веб-сайт]. – Режим доступа: <http://htmlbook.ru/css>;
18. EcmaScript 9 [Веб-сайт]. – Режим доступа: <https://www.ecma-international.org/ecma-262/9.0/index.html>
19. John Resig, Bear Bibeault, and Josip Maras. Secrets of the JavaScript Ninja, Second Edition. 2016;

20. Ethan Brown. Learning JavaScript: JavaScript Essentials for Modern Application Development, 3rd Edition. 2016;
21. Alex Banks, Eve Porcello. Learning React: Functional Web Development with React and Redux 1st Edition. 2017.
22. Github[Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://github.com>;
23. JavaScript [Веб-сайт]. – Режим доступу: <https://developer.mozilla.org/uk/docs/Web/JavaScript>.

# ДОДАТКИ

## ДОДАТОК А. ОРГАНІЗАЦІЙНА СТРУКТУРА УНІВЕРСИТЕТУ ТА КАФЕДРИ ІНФОРМАЦІЙНИХ СИСТЕМ НУХТ.

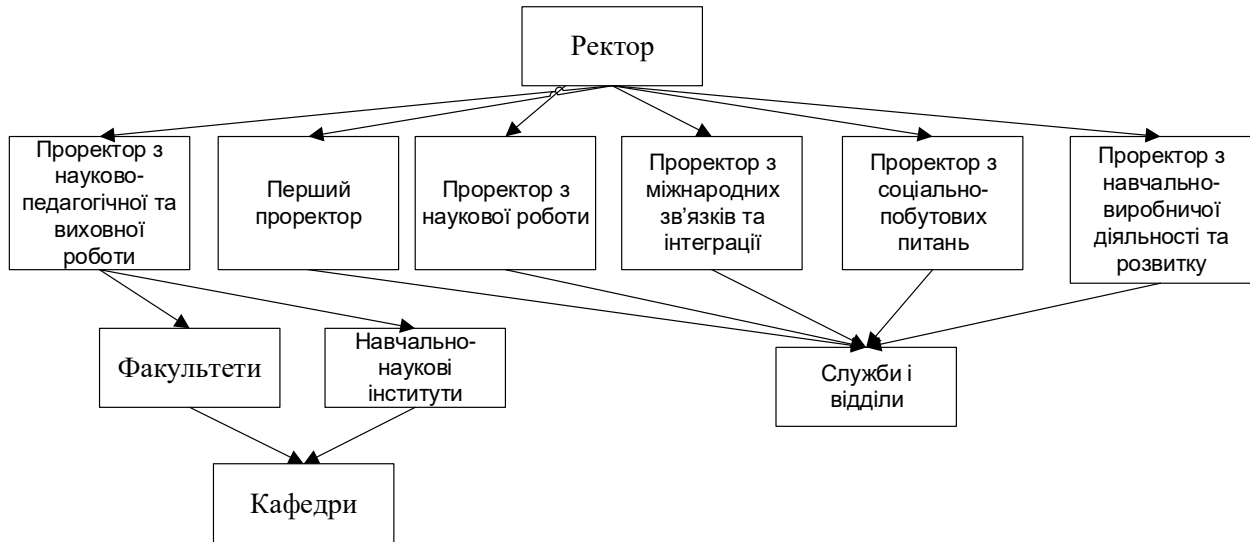


Рисунок 1. Організаційна структура університету.

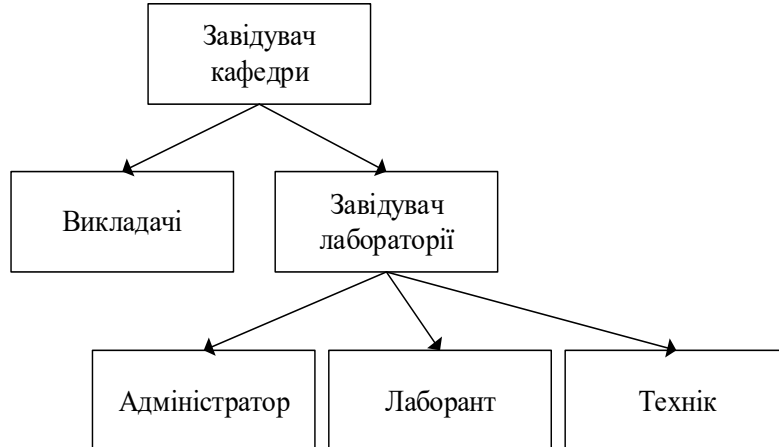


Рисунок 2. Організаційна структура кафедри інформаційних систем НУХТ.

## ДОДАТОК Б. ФУНКЦІОНАЛЬНІ МОДЕЛІ.

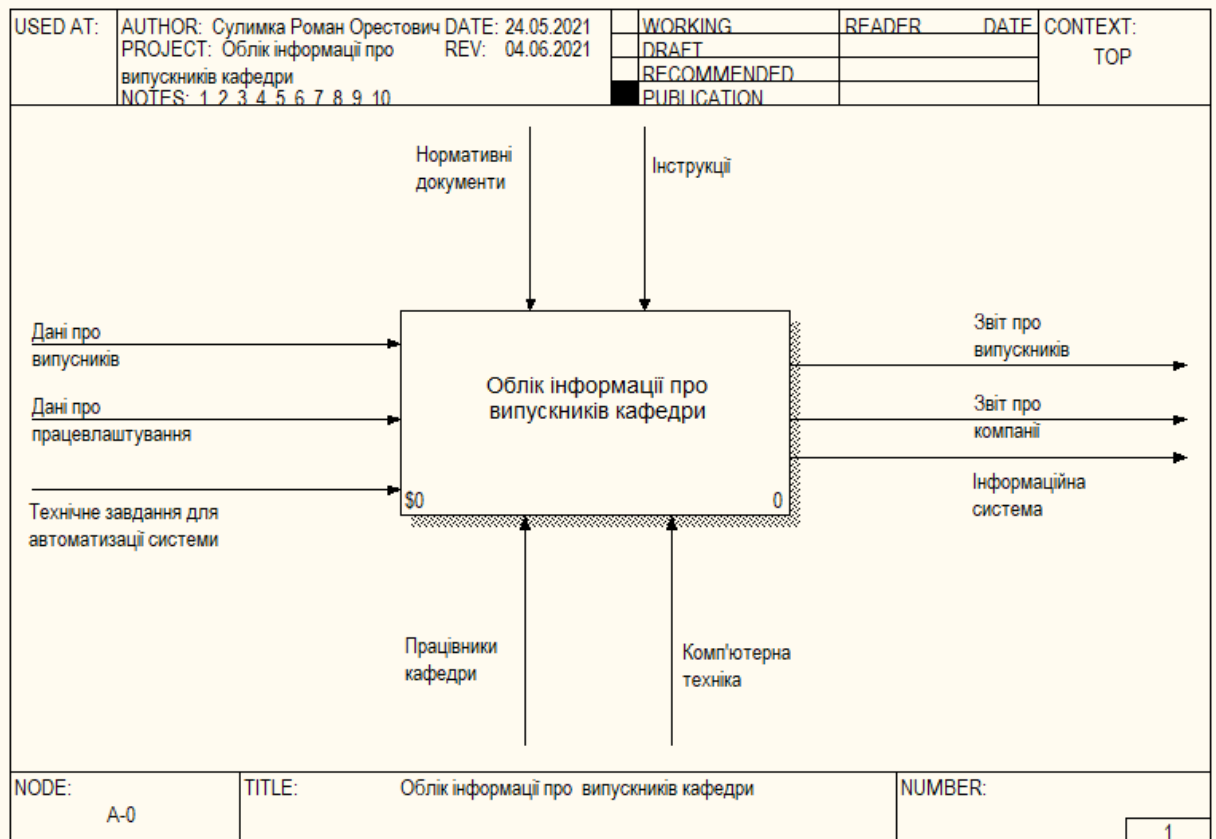


Рисунок 3. Контекстна діаграма моделі

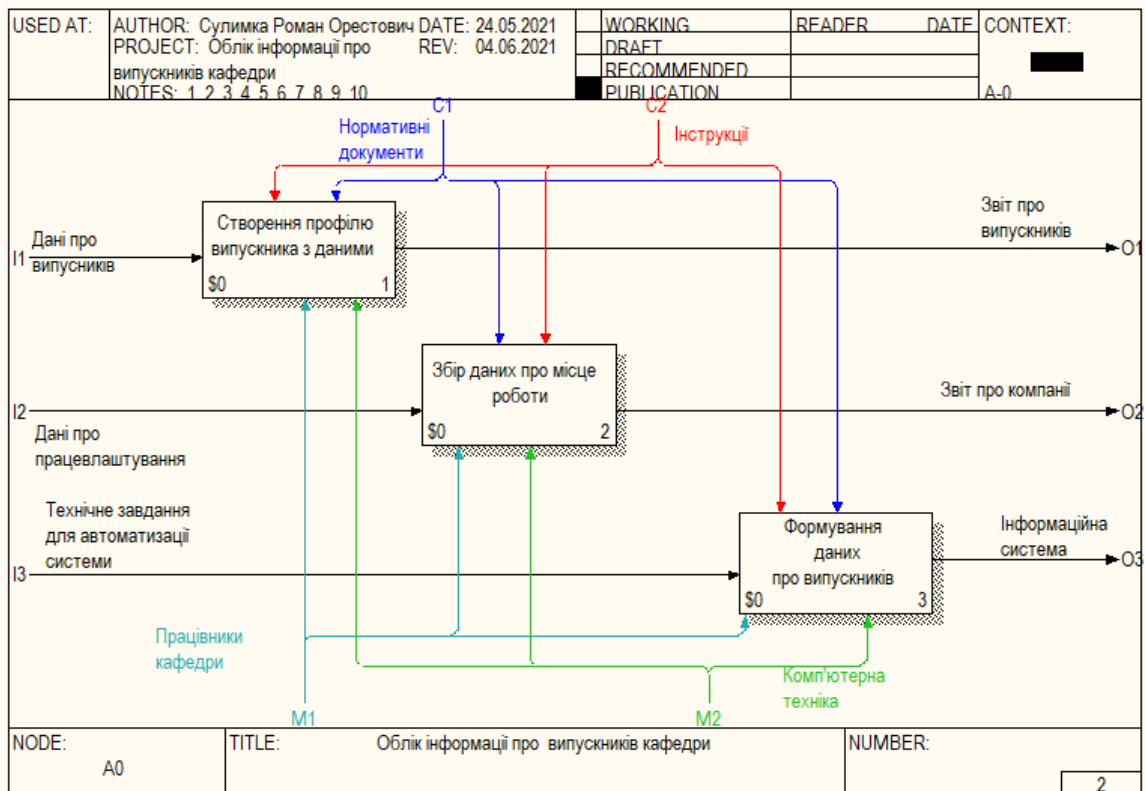


Рисунок 4. Перший рівень декомпозиції

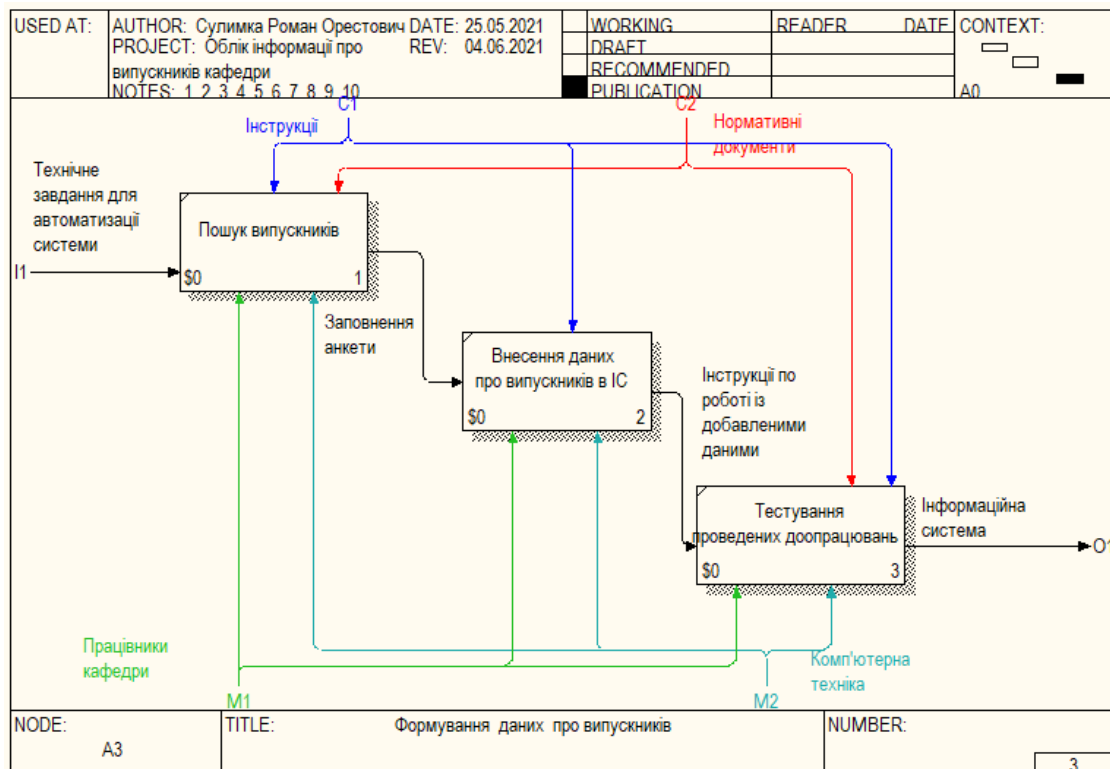


Рисунок 5. Другий рівень декомпозиції.

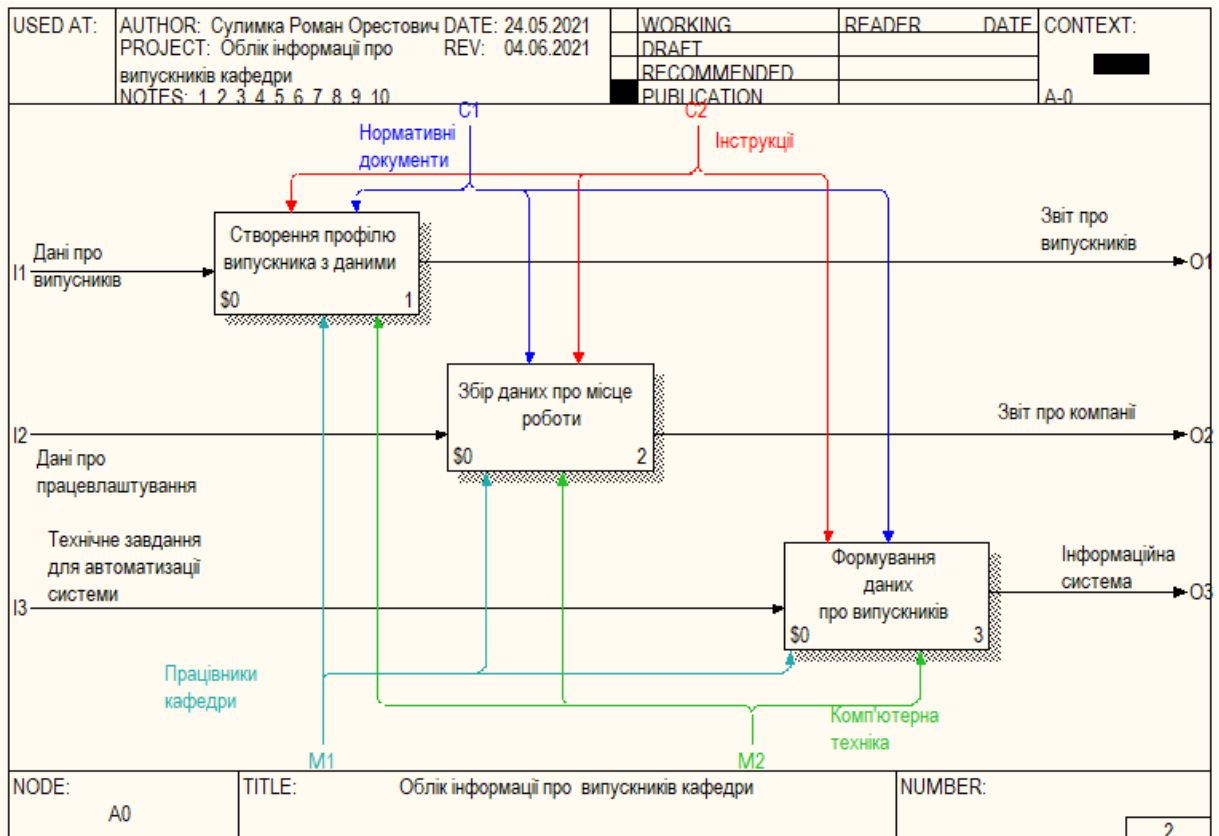


Рисунок 6. Модель AS-IS

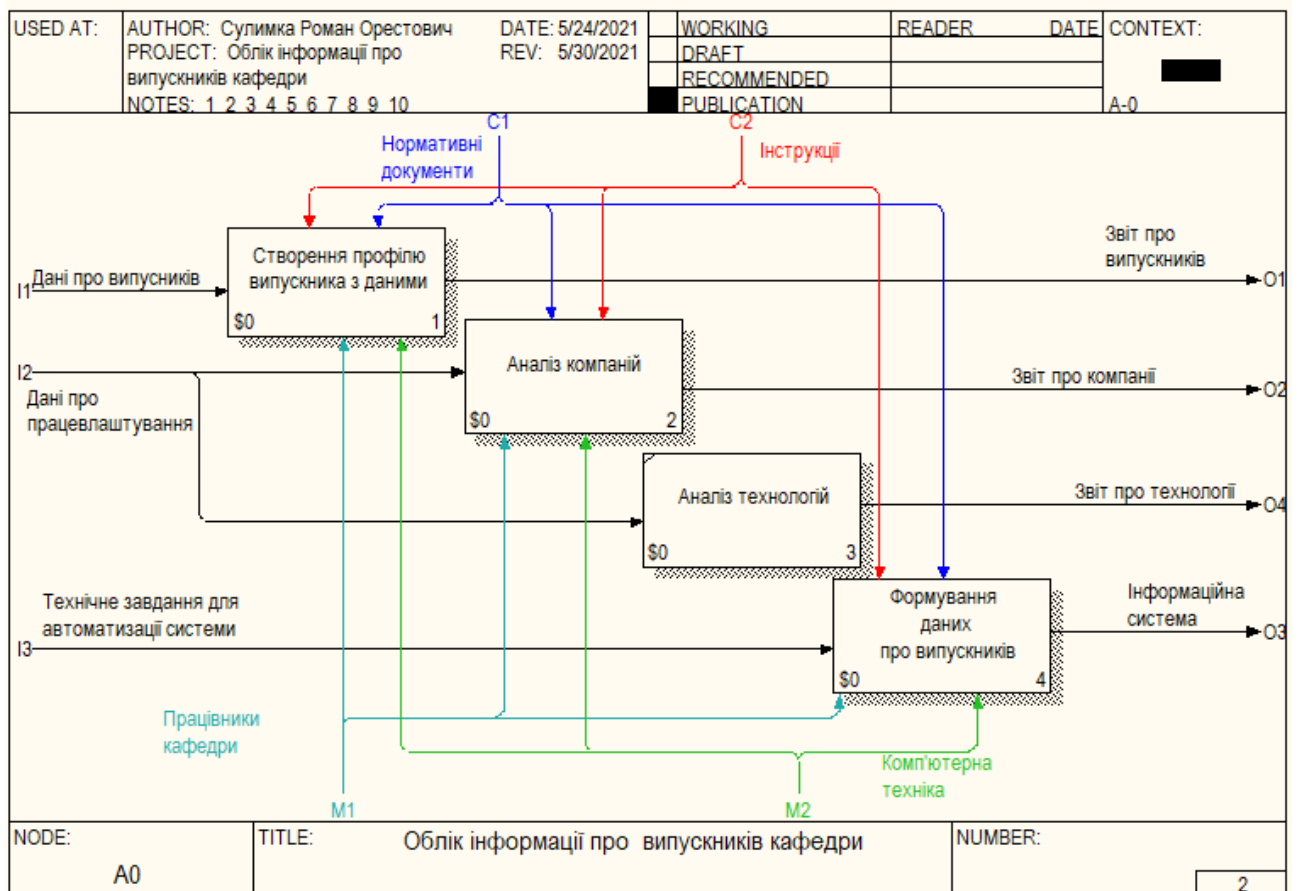


Рисунок 7. Модель To-Be.

## ДОДАТОК В. МОДЕЛІ ТА СИСТЕМИ БАЗИ ДАНИХ.

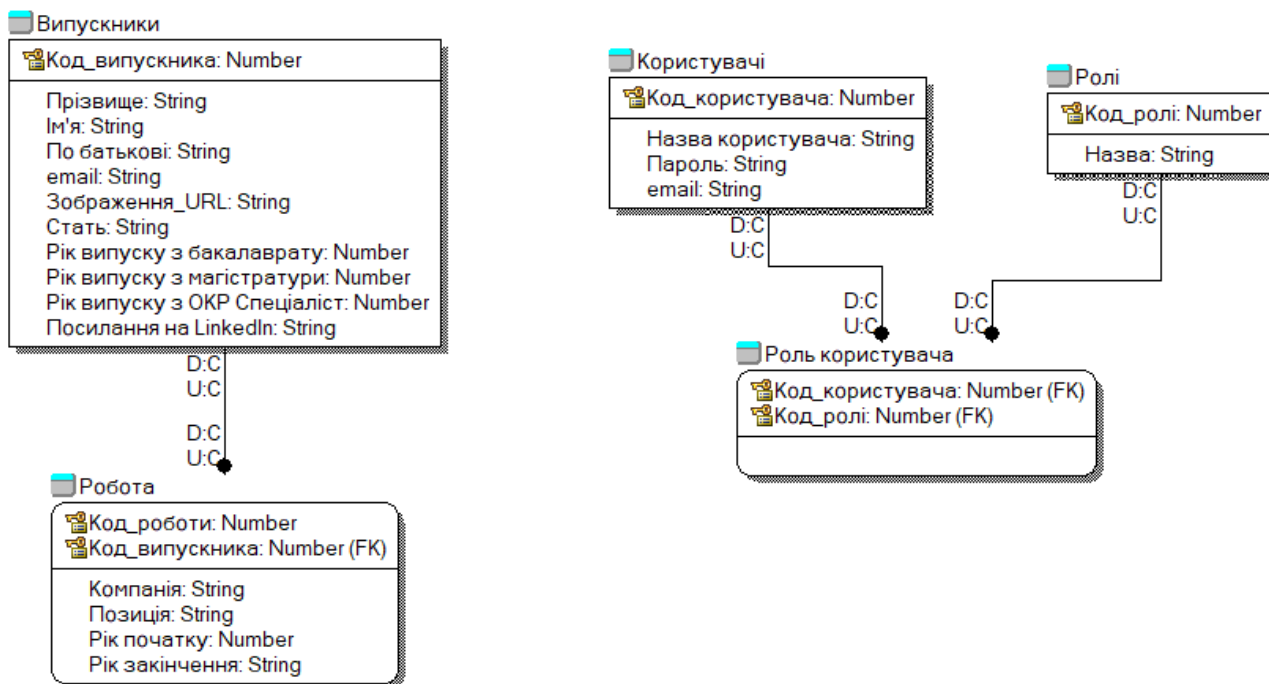


Рисунок 8. Логічна модель бази даних.

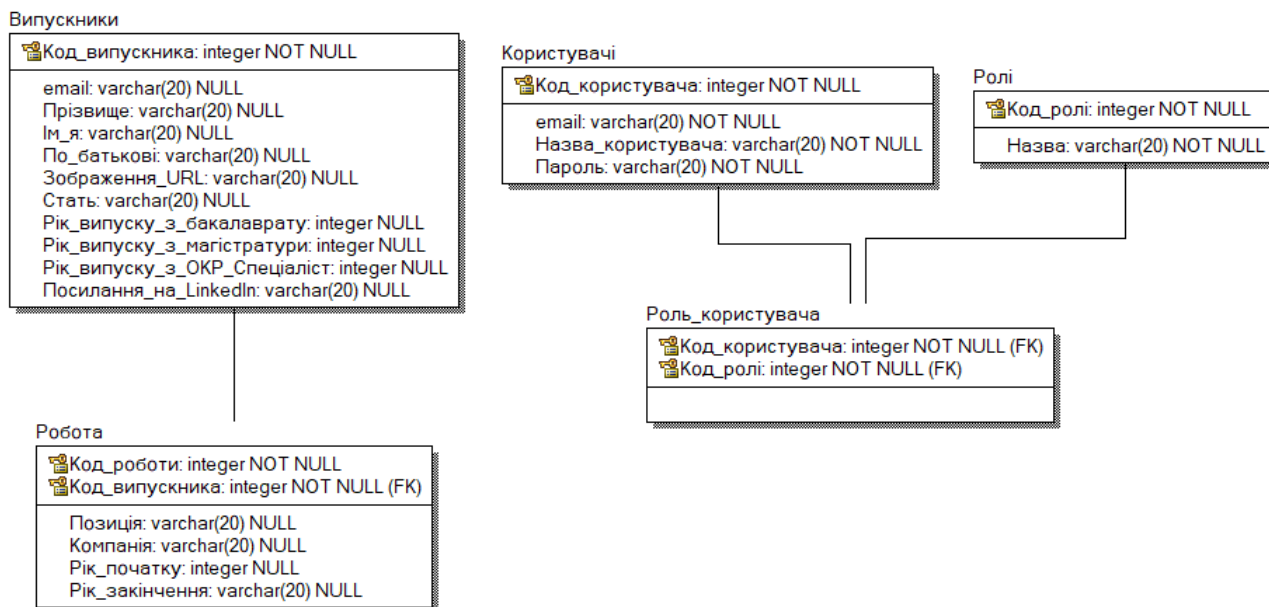


Рисунок 9. Фізична модель бази даних.

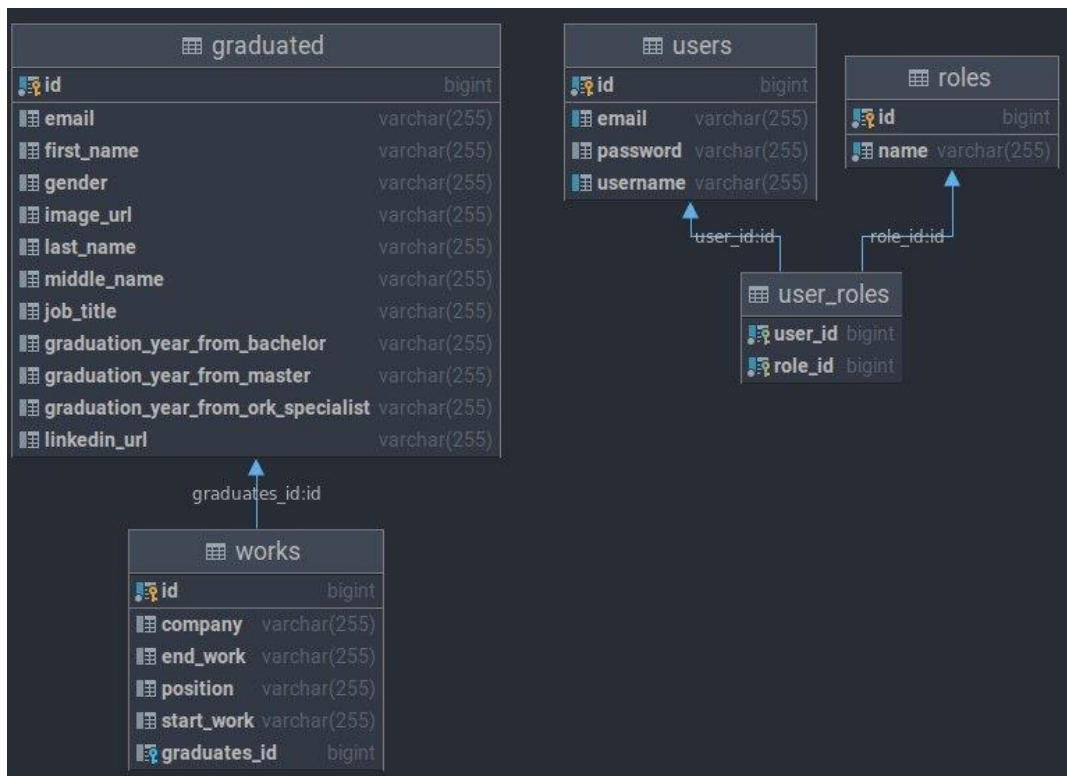


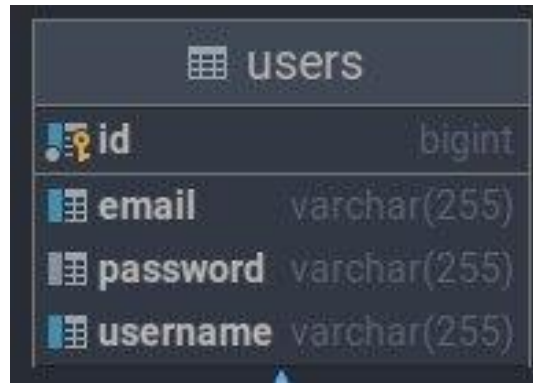
Рисунок 10. Структура БД.

graduated	
id	bigint
email	varchar(255)
first_name	varchar(255)
gender	varchar(255)
image_url	varchar(255)
last_name	varchar(255)
middle_name	varchar(255)
job_title	varchar(255)
graduation_year_from_bachelor	varchar(255)
graduation_year_from_master	varchar(255)
graduation_year_from_ork_specialist	varchar(255)
linkedin_url	varchar(255)

Рисунок 11. Таблица Выпускники.

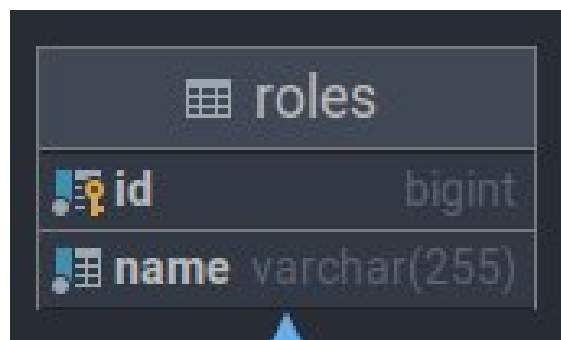
works	
id	bigint
company	varchar(255)
end_work	varchar(255)
position	varchar(255)
start_work	varchar(255)
graduates_id	bigint

Рисунок 12. Таблиця Робота.



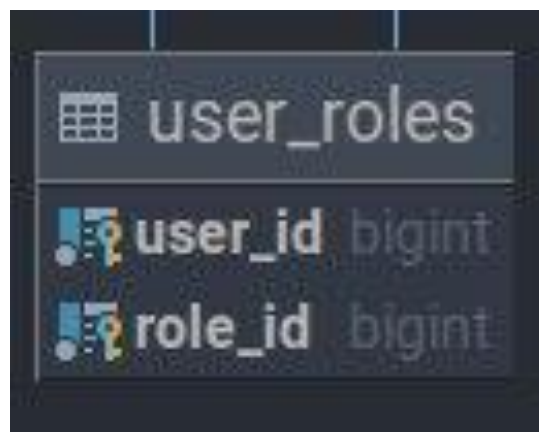
users	
id	bigint
email	varchar(255)
password	varchar(255)
username	varchar(255)

Рисунок 13. Таблиця Користувачі.



roles	
id	bigint
name	varchar(255)

Рисунок 14. Таблиця Ролі.



user_roles	
user_id	bigint
role_id	bigint

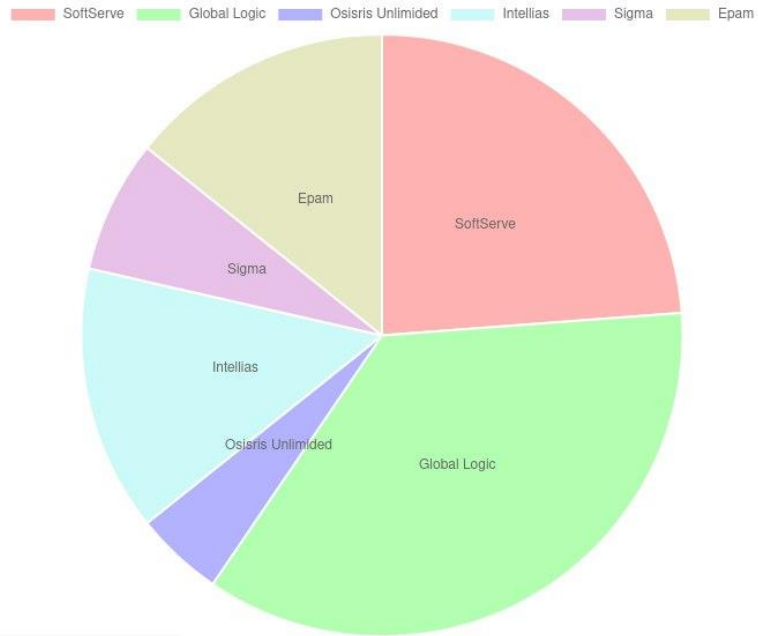
Рисунок 15. Таблиця розподіл по ролях

## ДОДАТОК Г. СКРІН-ШОТИ.

[Головна](#) / [Випускники](#) / [Діаграми](#)

Діаграма позицій

Діаграма технологій



Переключити легенду

Змінити позицію легенди

Рисунок 16. Вкладка з діаграмою компаній.

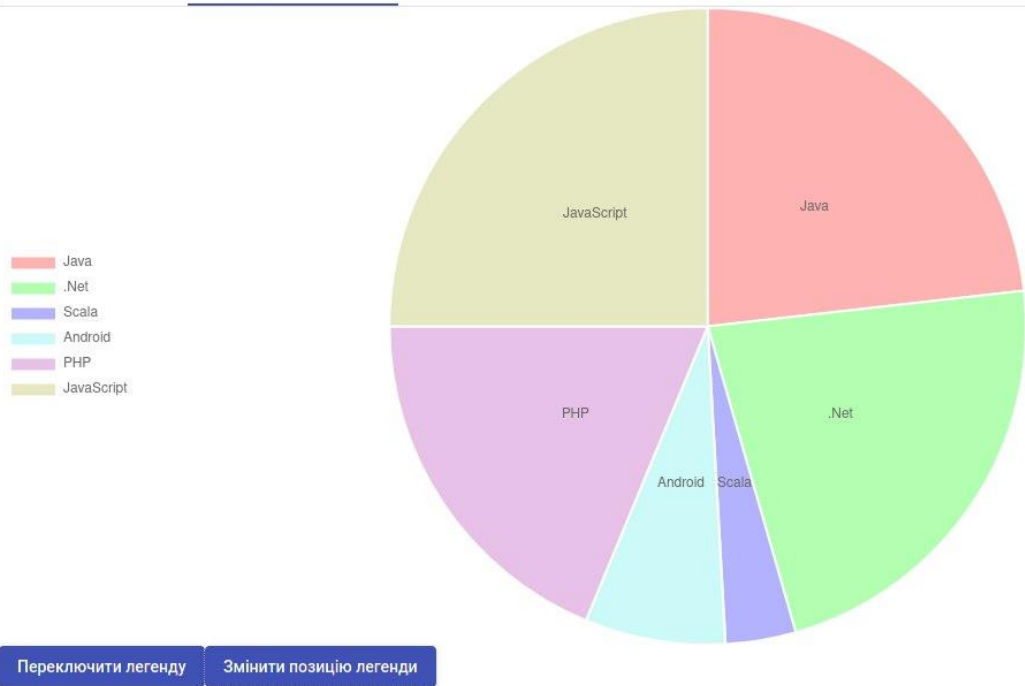


Рисунок 17. Вкладка з діаграмою технологій.



Студент групи 4-5ск:	Сулимка Роман Орестович
Технології:	Зараз технології приховано ▾
Контакти:	Контакти показано ▾



Доцент:	Костіков Микола Павлович
Контакти:	Контакти показано ▲
E-mail:	mikolaszk@gmail.com
LinkedIn:	<a href="#">Mykola Kostikov</a>
Telegram:	@mikolasz

Рисунок 18. Розширена інформація про автора.

Головна / Користувач / Адмін панель

Користувачі    Випускники    Робота

Знайти користувача  
admin

Id ↑	Ім'я користувача	E-mail	Роль	Оновити	Видалити
6	admin	roma.sulimak@gmail.com	ROLE_ADMIN		
7	admin2	admin@gmail.com	ROLE_ADMIN		

Items per page: 10    1 - 2 of 2    < >

Рисунок 19. Приклад пошуку користувачів за ім'ям.

Головна / Користувач / Адмін панель

Користувачі    Випускники    Робота

Знайти випускника  
Задор

Id	Ім'я	По батькові	Прізвище	E-mail	Компанія	Стать	Додати роботу	Оновити	Видалити
11	Андрій	Миколайович	Задорожній	oaja@gmail.com	Soft Serve	Чоловіча			

Items per page: 10    0 of 0    < >

Рисунок 20. Приклад пошуку випускників за прізвищем.

Головна / Користувач / Адмін панель

Користувачі    Випускники    Робота

Знайти роботу  
Java

Id	Позиція	Компанія	Рік початку	Рік закінчення	Оновити	Видалити
7	Java	Osisris Unlimided	2016	2018		
11	Java	EPAM	2016	2018		
18	Java Middle	Soft Serve	2015	-		
19	Java Developer	Innoveck	2014	2016		
20	Java	Soft Serve	2018	2019		

Items per page: 10    0 of 0    < >

Рисунок 21. Приклад пошуку за позицією.

The screenshot shows a web application window titled "Робота" (Work) with a sub-header "Користувач: admin" (User: admin). The form contains the following fields and elements:

- Id:** 6
- Ім'я користувача \*:** admin
- Password \*:** A password field with a masked input and a toggle icon.
- E-mail \*:** roma.sulimak@gmail.com
- Role Selection:** A dropdown menu with the following options: ROLE\_MODERATOR, ROLE\_USER, and ROLE\_ADMIN.

Рисунок 22. Вікно оновлення даних про користувача.

The screenshot shows a web application window titled "Створити нового користувача" (Create new user). The form contains the following fields and elements:

- Username \*:** A text input field.
- Пароль \*:** A password field with a masked input and a toggle icon.
- E-mail \*:** A text input field.
- Роль:** A dropdown menu.
- Buttons:** "Зберегти" (Save) and "Відмінити" (Cancel).

Рисунок 23. Вікно створення нового користувача.

Додати нову роботу

Компанія \*

Позиція \*

Рік початку роботи \*

Кінець роботи \*

Зберегти Скасувати

Рисунок 24. Вікно створення нової роботи.

Випусник: Nesvitaylo

Ім'я \*  
Daria

По батькові \*  
Volodymyrivna

Прізвище \*  
Nesvitaylo

E-mail \*  
isers@gmail.com

Стать \*  
Жіноча

Посилання на зображення \*  
<https://media-exp1.licdn.с>

Зберегти Скасувати

Рисунок 25. Вікно оновлення даних про випусника.

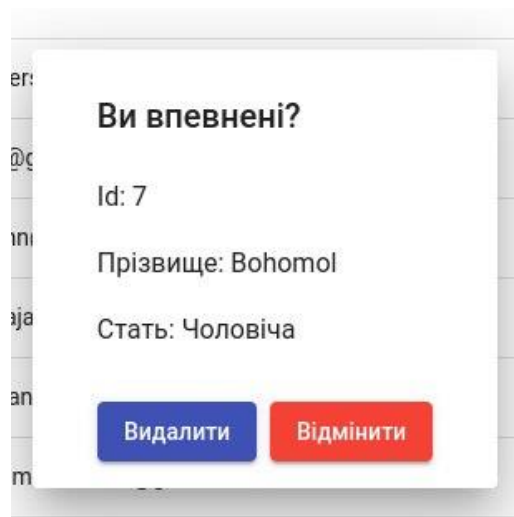


Рисунок 26. Вікно видалення випускника.

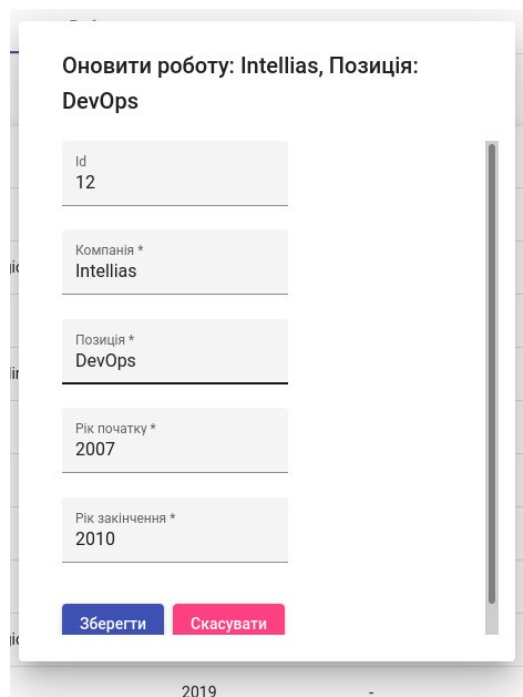


Рисунок 27. Вікно оновлення даних про місце роботи.

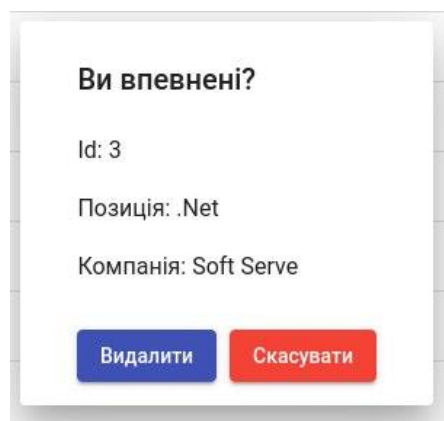


Рисунок 28. Вікно видалення місця роботи.

## ДОДАТОК Д. ЛІСТИНГ ПРОГРАМИ.

Контролер для передачі даних з сервера до клієнта для сторінки Випускники.

```
@CrossOrigin(origins="http://localhost:4200")
@RestController
@RequestMapping("/api/graduates")
public class GraduatedController {
    private final GraduatedService graduatedService;
    private final WorkService workService;
    private final GraduatedRepository graduatedRepository;

    public GraduatedController(GraduatedService graduatedService, WorkService workService,
GraduatedRepository graduatedRepository) {
        this.graduatedService = graduatedService;
        this.workService = workService;
        this.graduatedRepository = graduatedRepository;
    }

    @GetMapping("/all")
    public ResponseEntity<List<Graduated>> getAllGraduates(){
        List<Graduated> graduates = graduatedService.getAll();
        return new ResponseEntity<>(graduates, HttpStatus.OK);
    }

    @GetMapping("/find/{id}")
    public ResponseEntity<Graduated> getGraduatedById(@PathVariable("id") Long id){
        Graduated graduated = graduatedService.readById(id);
        return new ResponseEntity<>(graduated, HttpStatus.OK);
    }

    @PostMapping("/add")
    public ResponseEntity<Graduated> addGraduated(@RequestBody Graduated graduated){
        Graduated newGraduated = graduatedService.addGraduated(graduated);
        return new ResponseEntity<>(newGraduated, HttpStatus.CREATED);
    }

    @PutMapping("/update")
    public ResponseEntity<Graduated> updateGraduated(@RequestBody Graduated graduated){
        Graduated updateGraduated = graduatedService.updateGraduated(graduated);
        return new ResponseEntity<>(updateGraduated, HttpStatus.OK);
    }

    @DeleteMapping("/delete/{id}")
    public ResponseEntity<?> deleteGraduated(@PathVariable("id") Long id){
        graduatedService.deleteGraduated(id);
        return new ResponseEntity<>(HttpStatus.OK);
    }
}
```

```
}  
}
```

## Модель випускників

```
@AllArgsConstructor  
@NoArgsConstructor  
@EqualsAndHashCode  
@ToString  
@Data  
@Entity  
@Table  
public class Graduated implements Serializable {  
    @Id  
    @Column(nullable = false, updatable = false)  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
    private String firstName;  
    private String middleName;  
    private String lastName;  
    private String email;  
    private String gender;  
    private String imageUrl;  
    private String jobTitle;  
    private String graduationYearFromBachelor;  
    private String graduationYearFromMaster;  
    private String graduationYearFromOrkSpecialist;  
    private String linkedinUrl;  
    private String facebookUrl;  
    private String instagramUrl;  
    @Temporal(TemporalType.TIMESTAMP)  
    @JsonFormat(shape = JsonFormat.Shape.STRING, pattern = "yyyy-MM-dd")  
    private Date created;  
    @Temporal(TemporalType.TIMESTAMP)  
    @JsonFormat(shape = JsonFormat.Shape.STRING, pattern = "yyyy-MM-dd")  
    private Date updated;  
  
    @PrePersist  
    protected void onCreate() {  
        created = new Date();  
    }  
  
    @PreUpdate  
    protected void onUpdate() {  
        updated = new Date();  
    }  
}
```

```

}

@JsonManagedReference
@OneToMany(mappedBy = "graduated")
private List<Work> works;
}

```

## Сервіс для обробки випускників

```

@Service
@Transactional
public class GraduatedServiceImpl implements GraduatedService {
    private final GraduatedRepository graduatedRepository;

    @Autowired
    public GraduatedServiceImpl(GraduatedRepository graduatedRepository) {
        this.graduatedRepository = graduatedRepository;
    }

    @Override
    public Graduated addGraduated(Graduated graduated) {
        return graduatedRepository.save(graduated);
    }

    @Override
    public Graduated updateGraduated(Graduated graduated) {
        if (graduated != null) {
            Graduated oldGraduated = readById(graduated.getId());
            if (oldGraduated != null) {
                return graduatedRepository.save(graduated);
            }
        }
        throw new NullEntityReferenceException("User cannot be 'null'");
    }

    @Override
    public void deleteGraduated(Long id) {
        graduatedRepository.deleteGraduatedById(id);
    }

    @Override
    public Graduated readById(Long id) {
        return graduatedRepository.findGraduatedById(id)
            .orElseThrow(() -> new NullEntityReferenceException("Graduated by id " + id + " was not found"));
    }
}

```

```

@Override
public List<Graduated> getAll() {
    List<Graduated> graduates = graduatedRepository.findAll();
    return graduates.isEmpty() ? new ArrayList<>() : graduates;
}
}

```

Налаштування для захисту програми

```

@Configuration
@EnableWebSecurity
@EnableGlobalMethodSecurity(
    // securedEnabled = true,
    // jsr250Enabled = true,
    prePostEnabled = true)
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    private final UserDetailsServiceImpl userDetailsService;
    private final AuthEntryPointJwt unauthorizedHandler;

    public WebSecurityConfig(UserDetailsServiceImpl userDetailsService, AuthEntryPointJwt
unauthorizedHandler) {
        this.userDetailsService = userDetailsService;
        this.unauthorizedHandler = unauthorizedHandler;
    }

    @Bean
    public AuthTokenFilter authenticationJwtTokenFilter() {
        return new AuthTokenFilter();
    }

    @Override
    public void configure(AuthenticationManagerBuilder authenticationManagerBuilder) throws
Exception {
        authenticationManagerBuilder.userDetailsService(userDetailsService).passwordEncoder(passwordEncoder());
    }

    @Bean
    @Override
    public AuthenticationManager authenticationManagerBean() throws Exception {
        return super.authenticationManagerBean();
    }

    @Bean
    public PasswordEncoder passwordEncoder() {

```

```

        return new BCryptPasswordEncoder();
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.cors().and().csrf().disable()
            .exceptionHandling().authenticationEntryPoint(unauthorizedHandler).and()

            .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS).and()
            .authorizeRequests().antMatchers("/api/auth/**").permitAll()
            .antMatchers("/api/test/**").permitAll()
            .antMatchers("/graduates/**").permitAll()
            .antMatchers("/users/**").permitAll()
            .antMatchers("/works/**").permitAll()
            .antMatchers("/api/**").permitAll()
            .anyRequest().authenticated();

        http.addFilterBefore(authenticationJwtTokenFilter(),
UsernamePasswordAuthenticationFilter.class);
    }
}

```