

Я  
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

до кваліфікаційної роботи освітнього ступеня «бакалавр»  
за спеціальністю 121 «Інженерія програмного забезпечення»  
(освітня програма «Інженерія програмного забезпечення»)  
на тему:  
**«Сайт пошуку роботи з використанням штучного інтелекту»**

Виконав студент групи ІПЗ-21-5  
СЕМЕРИКОВ Нікіта Олексійович

Керівник роботи:  
ГРОМСЬКИЙ Олександр Олександрович

Рецензент:  
ВЛАСЕНКО Олег Васильович

Житомир – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ЗАТВЕРДЖУЮ

В.о. зав. каф. ПЗ, к.т.н., доц.

\_\_\_\_\_ Андрій МОРОЗОВ

(підпис)

«\_\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**

**на кваліфікаційну роботу**

Здобувач вищої освіти: Семериков Нікіта Олексійович

Керівник роботи: Громський Олександр Олександрович

Тема роботи: «Сайт пошуку роботи з використанням штучного інтелекту»,  
затверджена Наказом закладу вищої освіти від 23.02.2025р., №74/с

Об'єктом дослідження є використання інформаційних технологій для автоматизації процесів пошуку роботи та підбору персоналу. Предметом дослідження є використання технологій розробки вебзастосунків для створення платформи пошуку роботи з інтеграцією функцій штучного інтелекту.

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1			
2			
3			

РЕФЕРАТ

Кваліфікаційна робота освітнього ступеня «бакалавр» присвячена розробці сайту для пошуку роботи з використанням штучного інтелекту. Робота складається з програмного продукту та пояснювальної записки. Пояснювальна записка до кваліфікаційної роботи містить \_\_ сторінки, \_\_ ілюстрацій та \_\_ таблиць. Загальний обсяг \_\_ сторінок.

Метою роботи є створення сучасного, зручного сайту для пошуку роботи, який забезпечує автоматизацію процесів підбору кандидатів та взаємодії між роботодавцями і шукачами роботи.

У роботі поставлені основні завдання на розробку системи. Також було проведено аналіз аналогічних систем з метою виявлення їх переваг і недоліків. Було обрано клієнт-серверну архітектуру на основі фреймворку Next.js. Приведені основні сценарії використання системи, логіка взаємодії між модулями системи, діаграми потоків даних, структура бази даних та алгоритми роботи ключових модулів. База даних спроектована з використанням MongoDB для забезпечення гнучкості та масштабованості. Для інтеграції функціоналу штучного інтелекту використовувалося API Gemini AI, яке забезпечує автоматизацію аналізу текстових даних резюме та вакансій. Клієнтська і серверна частини реалізовані за допомогою Next.js, що дозволяє об'єднати серверний рендеринг та швидкодію інтерфейсу користувача.

Ключові слова: ПОШУК РОБОТИ, АВТОМАТИЗАЦІЯ, ШТУЧНИЙ ІНТЕЛЕКТ, NEXT.JS, GEMINI AI, MONGODB.

					ДУ «Житомирська політехніка».25.121.21.000 - ПЗ								
Змн.	Арк.	№ докум.	Підпис	Дата									
Розроб.		Семериков Н.О.			Пояснювальна записка			Літ.		Арк.		Аркушів	
Керівник		Громський О.О.								3		38	
								ФІКТ Гр. ІПЗ-21-5					
Н. контр.													
Затвердив													

## ABSTRACT

The qualification work for the bachelor's degree is dedicated to the development of a job search website utilizing artificial intelligence. The work consists of a software product and an explanatory note. The explanatory note includes 82 pages, 65 illustrations, and 20 tables, with a total volume of 90 pages.

The aim of the work is to create a modern, user-friendly job search website that automates the processes of candidate selection and interaction between employers and job seekers.

The work sets the main tasks for the system's development. Additionally, an analysis of similar systems was conducted to identify their advantages and disadvantages. A client-server architecture based on the Next.js framework was chosen. The main usage scenarios of the system, interaction logic between system modules, data flow diagrams, database structure, and algorithms for the operation of key modules are presented.

The database was designed using MongoDB to ensure flexibility and scalability. To integrate artificial intelligence functionality, the Gemini AI API was used, providing automation for analyzing text data from resumes and job postings. Both the client and server sides were implemented using Next.js, which combines server-side rendering with high interface performance.

**Keywords:** JOB SEARCH, AUTOMATION, ARTIFICIAL INTELLIGENCE, NEXT.JS, GEMINI AI, MONGODB.

		Семериков Н.О			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

# ЗМІСТ

		Семериков Н.О			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

API – Application Programming Interface

CQRS – command and query responsibility segregation

JSON – JavaScript Object Notation

JWT – JSON Web Token

RESTFull API – representational state transfer API

SPA – single page application

		Семериков Н.О			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

У сучасному світі швидкий розвиток інформаційних технологій та штучного інтелекту (ШІ) істотно впливає на всі аспекти суспільного життя, у тому числі й на ринок праці. Однією з ключових проблем сучасного ринку праці є ефективний пошук роботи та підбір персоналу. Традиційні методи, засновані на ручному аналізі резюме, часто є трудомісткими, потребують значних витрат часу та ресурсів і не завжди гарантують оптимальні результати. Водночас технології ШІ відкривають нові можливості для автоматизації та підвищення ефективності цих процесів.

Актуальність теми полягає у необхідності створення інноваційних рішень для ринку праці, які забезпечать ефективний пошук роботи для кандидатів та підбір персоналу для роботодавців. Використання технологій штучного інтелекту дозволяє автоматизувати аналіз резюме, забезпечуючи персоналізовані рекомендації для кандидатів щодо покращення їх професійних характеристик, а також допомагає роботодавцям знаходити найбільш підходящих кандидатів на основі їх кваліфікацій, досвіду та навичок. Це сприятиме скороченню часу на пошук роботи та підбір персоналу, а також підвищенню якості цих процесів.

Метою кваліфікаційної роботи є розробка сучасного веб-сайту для пошуку роботи з використанням технологій штучного інтелекту, який буде надавати роботодавцям інструменти для автоматизованого підбору персоналу, а користувачам — персоналізовані рекомендації щодо покращення їх резюме та підвищення шансів на працевлаштування. Основною метою є підвищення ефективності взаємодії між роботодавцями та кандидатами.

Об'єктом дослідження є процеси пошуку роботи та підбору персоналу на ринку праці за допомогою цифрових технологій.

Предметом дослідження є веб-сайт для пошуку роботи з інтеграцією штучного інтелекту, що забезпечує автоматизацію підбору персоналу та надає індивідуальні рекомендації для користувачів.

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		



## РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

### 1.1 Постановка задачі

Метою цієї роботи є розробка інноваційної веб-платформи для пошуку роботи, що використовує алгоритми штучного інтелекту (ШІ) для оптимізації процесів підбору персоналу та надання персоналізованих рекомендацій кандидатам.

Одним із ключових елементів функціоналу платформи є створення інтерактивного середовища для роботодавців, яке полегшує аналіз відгуків на вакансії. Система надаватиме наступні можливості:

- **Інтерактивний чат-помічник на основі ШІ.** Веб-платформа забезпечить роботодавцям доступ до віртуального HR-асистента. За допомогою чат-бота, подібного до ChatGPT, роботодавець зможе ставити запитання про кандидатів, отримувати детальну інформацію про їхній досвід, навички чи відповідність конкретним критеріям вакансії.
- **Пошук вакансій і кандидатів.** Користувачам надається можливість здійснювати пошук із застосуванням зручних фільтрів, таких як місце роботи, рівень заробітної плати, досвід тощо.
- **Збереження вакансій.** Користувачі можуть зберігати цікаві вакансії у своєму особистому кабінеті для подальшого перегляду. Це дозволяє ефективно управляти пошуком роботи, повертаючись до вибраних пропозицій у зручний час.
- **Персоналізовані рекомендації.** Для кандидатів платформа генерує індивідуальні рекомендації щодо покращення резюме та професійних навичок. Ці рекомендації базуються на аналізі резюме, вакансій та актуальних трендів ринку праці.
- **Модуль аналітики для адміністраторів.** Цей модуль надасть адміністраторам веб-сайту інструменти для моніторингу активності користувачів, аналізу взаємодії роботодавців та кандидатів, а також оцінки ефективності роботи платформи.

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Основні етапи розробки сайту для пошуку роботи з використанням штучного інтелекту:

1. Проектування платформи:

- Розробка архітектури системи, що включає серверну, клієнтську частини та базу даних.
- Проектування користувацького інтерфейсу з урахуванням потреб різних груп користувачів (роботодавців і кандидатів).

2. Реалізація функціоналу сайту:

- Інтеграція моделі штучного інтелекту Gemini для аналізу резюме, автоматизації підбору персоналу та формування персоналізованих рекомендацій.
- Налаштування параметрів і адаптація готової моделі під специфічні завдання платформи, такі як ранжування кандидатів, оцінка відповідності вакансій і надання рекомендацій.
- Реалізація механізмів пошуку та фільтрації вакансій і кандидатів.
- Впровадження системи збереження вакансій і профілів.
- Створення системи авторизації з підтримкою різних ролей користувачів.

3. Тестування сайту:

- Функціональне тестування усіх компонентів.
- Перевірка точності роботи алгоритмів ШІ.
- Тестування інтерфейсу для забезпечення зручності використання.

4. Розробка дизайну вебсайту:

- Створення загальної концепції дизайну сайту.
- Розробка зручного інтерфейсу для користувачів різних ролей (роботодавців, кандидатів, адміністраторів).
- Оптимізація UI/UX для забезпечення інтуїтивної навігації.

Результатом розробки стане веб-сайт, який полегшить процес підбору персоналу для роботодавців і підвищить ефективність пошуку роботи для

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

кандидатів, забезпечуючи точний аналіз даних, персоналізовані рекомендації та можливості аналітики для адміністраторів.

## 1.2 Аналіз аналогів програмного продукту

Для розробки сучасного сайту з пошуку роботи з використанням штучного інтелекту важливо проаналізувати існуючі рішення, їхні особливості та функціонал. У цьому розділі розглядаються три провідні платформи в Україні: robota.ua, work.ua та djinni.co.

### Robota.ua

Robota.ua — одна з найстаріших платформ для пошуку роботи в Україні, яка орієнтована на широкий спектр вакансій. Вона дозволяє роботодавцям і кандидатам взаємодіяти в межах єдиної екосистеми.

### Переваги:

1. Фільтри пошуку: користувачі можуть шукати вакансії за категоріями, місцем роботи, рівнем зарплати, досвідом.
2. Створення резюме: кандидатам надається можливість створити резюме за шаблоном безпосередньо на платформі.
3. Повідомлення про вакансії: автоматичне надсилання сповіщень про нові вакансії відповідно до налаштувань користувача.
4. Аналітика для роботодавців: кількість переглядів і відгуків на вакансії.

### Недоліки:

1. Відсутність функцій персоналізованих рекомендацій для кандидатів.
2. Недостатньо інструментів для взаємодії між роботодавцями та пошукачами.
3. Модуль аналітики обмежений базовими показниками.

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

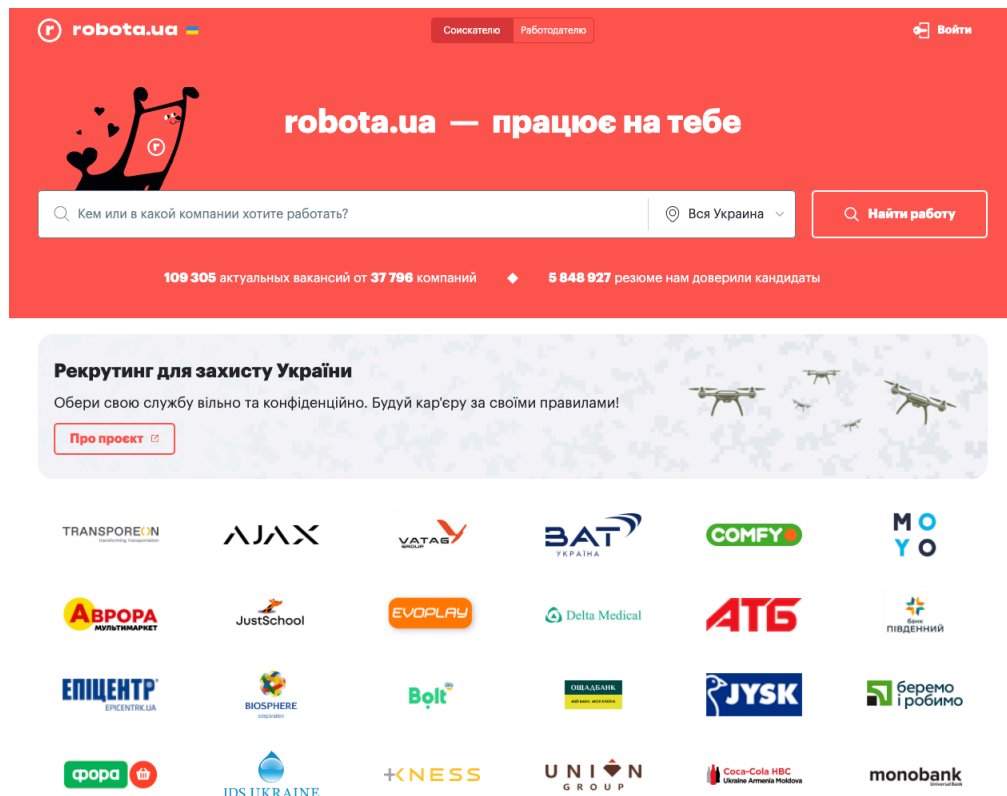


Рис. 1.1. Інтерфейс сайту Robota.ua

## Work.ua

Work.ua — ще одна провідна платформа, яка конкурує з Robota.ua, пропонуючи подібний функціонал. Вона надає можливість публікації вакансій, швидкого пошуку роботи, а також доступ до корисних матеріалів і порад щодо працевлаштування.

### Переваги:

1. Функціональний пошук: широкий вибір фільтрів для зручного підбору вакансій.
2. Розділ із корисними порадами: поради щодо складання резюме, проходження співбесід тощо.
3. Швидкий відгук на вакансії: можливість відправити резюме роботодавцю прямо з сайту.
4. Оцінка компаній: користувачі можуть залишати відгуки про роботодавців.

### Недоліки:

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Відсутність інтерактивного помічника на основі ШІ.
2. Недостатня персоналізація рекомендацій для кандидатів і роботодавців.
3. Можливості аналітики для роботодавців також обмежені.

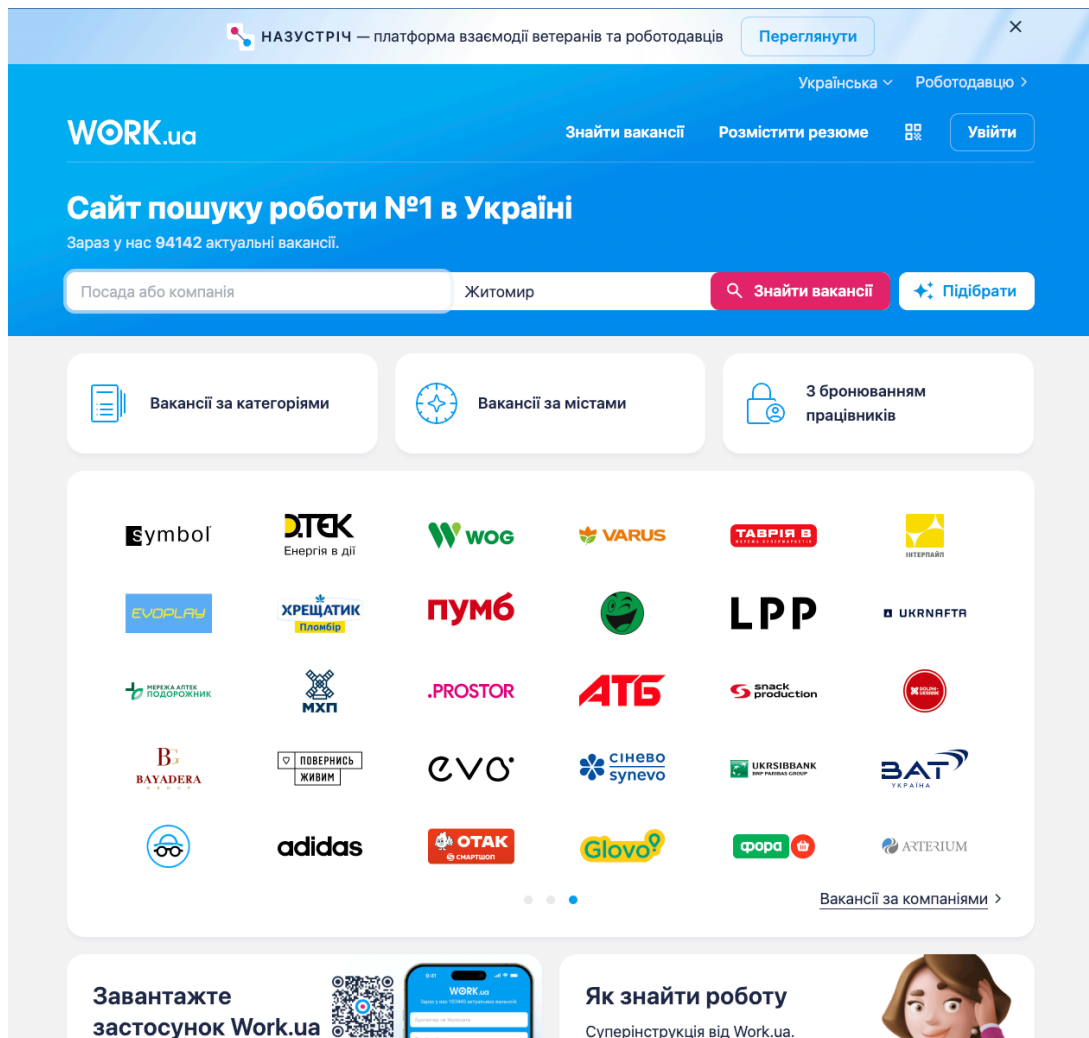


Рис. 1.2. Інтерфейс сайту Work.ua

Djinni.co

Djinni.co — спеціалізована платформа для IT-фахівців, яка відрізняється фокусом на конфіденційність кандидатів і автоматичному підборі вакансій. Вона надає можливість роботодавцям отримувати профілі кандидатів із системи, а кандидатам — залишатися анонімними до моменту встановлення контакту.

Переваги:

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Анонімний профіль: кандидати можуть подаватися на вакансії, не розголошуючи свої особисті дані.
2. Фокус на ІТ: вакансії переважно для технічних спеціалістів.
3. Інтелектуальний підбір: система пропонує вакансії, які найбільше відповідають профілю кандидата.
4. Прямий контакт: роботодавці безпосередньо спілкуються з кандидатами через платформу.

Недоліки:

1. Обмежений сектор (переважно ІТ).
2. Відсутність універсальних функцій для широкого ринку праці.
3. Недостатній функціонал для створення аналітичних звітів.

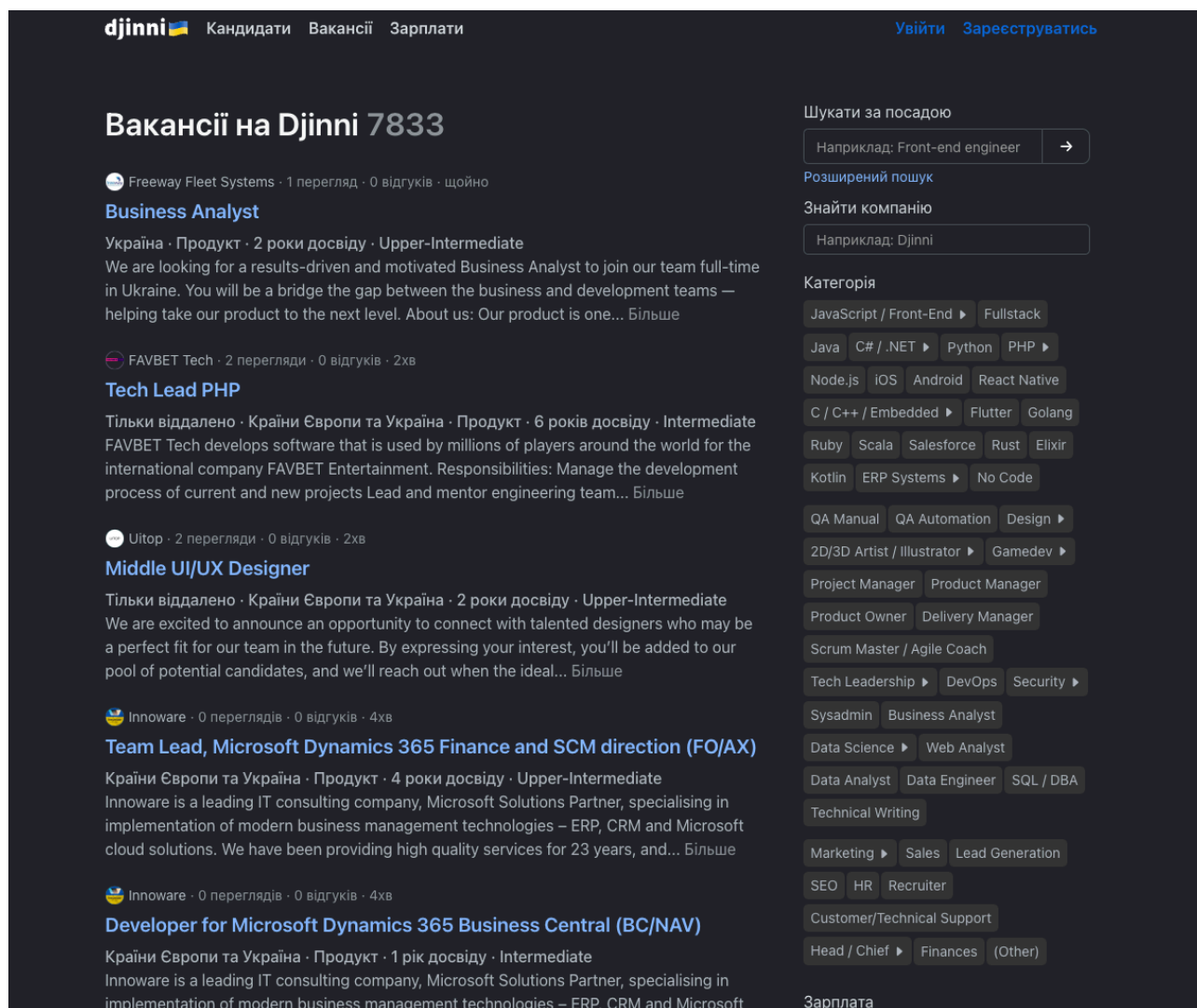


Рис. 1.2. Інтерфейс сайту Djinni.co

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Проведений аналіз показав, що популярні платформи мають багатий функціонал, однак не використовують повною мірою можливості штучного інтелекту для автоматизації процесів. Основною відмінністю розроблюваного сайту стане інтеграція ШІ для аналізу резюме, рекомендацій щодо кандидатів і аналітики, що дозволить забезпечити персоналізований підхід до кожного користувача. Це підвищить ефективність взаємодії між роботодавцями та кандидатами.

### 1.3. Вибір архітектури платформи для надання юридичної допомоги з ші

Для створення сайту з пошуку роботи, що використовує штучний інтелект, було обрано клієнт-серверну архітектуру, в якій клієнтська та серверна частини реалізовані за допомогою Next.js. Ця архітектура дозволяє інтегрувати фронтенд і бекенд в єдиному середовищі, забезпечуючи простоту обслуговування, швидкість роботи і зручність взаємодії користувачів із системою.

Унікальність підходу полягає у використанні функціоналу Server Actions, який дозволяє виконувати серверні операції безпосередньо в рамках компонентів Next.js. Це сприяє зменшенню складності розробки та забезпечує високу продуктивність сайту.

Клієнтська частина сайту, розроблена на базі Next.js, забезпечує швидкий доступ користувачів до функцій, таких як перегляд вакансій, отримання рекомендацій і взаємодія з іншими можливостями. Вона відповідає за збір даних від користувачів (наприклад, резюме чи налаштувань пошуку), надсилання запитів до серверної частини та відображення результатів обробки. Використання підходу SPA (Single Page Application) забезпечує плавний досвід роботи без необхідності перезавантаження сторінок. Завдяки адаптивному дизайну сайт коректно працює на різних пристроях: мобільних телефонах, планшетах і комп'ютерах. Основна задача клієнтської частини — забезпечити зручний інтерфейс і обмін даними із серверною частиною для виконання складних операцій.

		Семериков Н.О			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

Серверна частина сайту, також реалізована з використанням Next.js, виконує обробку запитів і реалізацію бізнес-логіки. За допомогою технології Server Actions сервер аналізує резюме користувачів, співставляє їх із вакансіями та формує персоналізовані рекомендації на основі алгоритмів штучного інтелекту. Сервер також зберігає дані, включаючи профілі користувачів, історію взаємодій і вакансії, забезпечуючи надійність та швидкий доступ до інформації. Він управляє правами доступу до функцій сайту, включаючи преміум-опції, такі як розширена аналітика чи рекомендації. Серверна частина є центральним вузлом обробки даних і гарантує ефективну роботу всього сайту.

#### 1.4 Обґрунтування вибору інструментальних засобів

Для реалізації сайту пошуку роботи з використанням штучного інтелекту було обрано сучасний і продуктивний стек технологій, який забезпечує ефективну розробку, масштабованість та інтеграцію з інструментами ІІІ. Обрані засоби дозволяють досягти поставлених цілей, забезпечуючи високу продуктивність, зручність розробки та якісний користувацький досвід.

- Next.js. Фреймворк Next.js було обрано для побудови як клієнтської, так і серверної частини сайту. Next.js підтримує сучасні підходи до розробки, такі як Server-side Rendering (SSR), Static Site Generation (SSG) та Server Actions, що дозволяє інтегрувати серверну логіку безпосередньо у фронтенд-компоненти. Його можливості спрощують розробку SPA-додатків із високою продуктивністю та зручністю для користувачів.
- NextAuth.js. Для реалізації системи автентифікації обрано NextAuth.js, оскільки цей інструмент ідеально інтегрується з Next.js і забезпечує гнучку підтримку різних методів автентифікації (Google, GitHub, email, та інші). NextAuth.js дозволяє легко управляти сесіями користувачів, включаючи підтримку токенів доступу для безпечної взаємодії з API.
- TypeScript. Для забезпечення високої надійності та зручності у розробці було обрано TypeScript як основну мову програмування. TypeScript є надбудовою над JavaScript, яка додає строгий типізаційний контроль, що

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				15
Змн.	Арк.	№ докум.	Підпис	Дата		



дозволяє виявляти помилки на етапі компіляції. Це зменшує кількість багів у коді та полегшує підтримку проєкту. TypeScript інтегрується з Next.js, забезпечуючи можливість типізації компонентів, API, моделей бази даних і бізнес-логіки, що сприяє більш структурованій розробці.

- **Mongoose.** Для роботи з базою даних використовується Mongoose — ORM-бібліотека для MongoDB. Вона спрощує маніпуляції з базою даних, забезпечуючи зручну модель даних, валідацію та гнучкі механізми запитів. Використання Mongoose дозволяє ефективно управляти профілями користувачів, історією взаємодій та іншими даними, необхідними для роботи сайту.
- **Vercel AI SDK.** Для інтеграції можливостей штучного інтелекту використовується Vercel AI SDK, який спрощує взаємодію з мовними моделями та іншими алгоритмами ШІ. Цей інструмент дозволяє створювати функції, що забезпечують персоналізовані рекомендації для користувачів, аналіз резюме та автоматизоване формування відповідей для покращення взаємодії з користувачами.
- **Vercel/blob.** Для роботи з файлами, такими як резюме користувачів у форматі PDF або інших документах, використовується Vercel/blob. Цей інструмент дозволяє легко завантажувати, зберігати та обробляти файли в межах платформи Vercel. Використання Vercel/blob спрощує керування файлами, забезпечує швидкий доступ до них і інтегрується з іншими частинами сайту, такими як система аналізу резюме.
- **Lucide-react** Для створення іконок у користувацькому інтерфейсі обрано бібліотеку Lucide-react. Її переваги включають сучасний дизайн, високу кастомізованість та просту інтеграцію з React-компонентами. Використання цієї бібліотеки дозволяє створювати привабливий і функціональний інтерфейс для сайту.

Обраний стек технологій забезпечує швидкість розробки, масштабованість і стабільність завдяки використанню Next.js, TypeScript та інших інструментів. Інтеграція з Vercel AI SDK, Vercel/blob і Mongoose

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяє створювати функціональний, безпечний та інноваційний сайт із сучасним дизайном і підтримкою III.

Для реалізації сайту пошуку роботи з використанням штучного інтелекту було обрано базу даних MongoDB у поєднанні з бібліотекою Mongoose. MongoDB забезпечує зручне зберігання даних у форматі JSON/BSON, що ідеально підходить для роботи з динамічними об'єктами, такими як резюме, профілі користувачів і вакансії. Щоб обґрунтувати вибір, розглянемо порівняння MongoDB з іншими популярними системами управління базами даних (СУБД): PostgreSQL і MySQL.

Для оцінки основних характеристик СУБД важливо враховувати тип даних, що підтримується, можливості масштабування, продуктивність і гнучкість у роботі з даними. У таблиці 1.1 наведено порівняння ключових характеристик MongoDB, PostgreSQL і MySQL.

Таблиця 1.1

Характеристика	MongoDB	PostgreSQL	MySQL
Тип даних	Документно-орієнтована (JSON/BSON)	Реляційна	Реляційна
Масштабованість	Горизонтальна	Горизонтальна та вертикальна	Переважно вертикальна
Гнучкість схеми	Динамічна, без чітко визначеної схеми	Жорстко визначена схема	Жорстко визначена схема
Продуктивність	Висока для обробки нереляційних даних	Ефективна для складних реляційних зв'язків	Висока для простих реляційних запитів
Підтримка JSON	Природна, основний формат	JSONB, менш	Обмежена

	зберігання	продуктивний	підтримка
Простота інтеграції	Висока завдяки Mongoose	Стандартна для SQL	Стандартна для SQL

Для визначення відповідності СУБД до вимог проєкту, таких як робота з великою кількістю користувачів або обробка динамічних даних, було проаналізовано їхні можливості. У таблиці 1.2 наведено відповідність MongoDB, PostgreSQL та MySQL ключовим вимогам проєкту.

Таблиця 1.2.

вимога	MongoDB	PostgreSQL	MySQL
Гнучкість у зміні структури даних	Висока	Обмежена	Низька
Масштабування для великої кількості користувачів	Горизонтальне	Горизонтальне	Переважно вертикальне
Обробка нереляційних даних	Дуже ефективна	Обмежена	Складно реалізувати
Продуктивність із великим обсягом даних	Відмінна	Висока	Висока
Легка інтеграція з JavaScript	Ідеальна	Складна	Складна
Підтримка складних зв'язків	Обмежена	Повна	Повна

Для оцінки характеристик баз даних важливо врахувати такі параметри, як підтримуваний обсяг даних і можливості захисту. У таблиці 1.3 наведено порівняння MongoDB, PostgreSQL і MySQL за цими критеріями.

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.2.

Характеристика	MongoDB	PostgreSQL	MySQL
Максимальний обсяг бази даних	16 ТБ на колекцію	Необмежений	Необмежений
Розмір документа/рядка	До 16 МБ	До 1.6 ТБ на таблицю	До 4 ГБ на таблицю
Масштабованість	Горизонтальна (sharding)	Горизонтальна та вертикальна	Переважно вертикальна
Захист даних	Шифрування на рівні поля і TLS	Шифрування на рівні поля і TLS	Шифрування через сторонні засоби
Резервне копіювання	Інтегрована підтримка	Інтегрована підтримка	Залежить від реалізації

На основі аналізу MongoDB було обрано як оптимальну базу даних для цього проєкту завдяки її гнучкості, підтримці великих обсягів даних, надійності та засобам захисту. Як показано у таблицях 1.1–1.3, MongoDB перевершує PostgreSQL і MySQL у роботі з динамічними структурами даних, масштабованістю та інтеграцією із сучасними технологіями.

### Висновки до першого розділу

У цьому розділі визначено мету роботи, сформульовано основні функціональні вимоги до платформи та проведено аналіз аналогів існуючих рішень. Основною перевагою розроблюваної системи стане інтеграція штучного інтелекту для автоматизації процесів і персоналізації рекомендацій. Обґрунтовано вибір архітектури та інструментальних засобів, що забезпечують ефективність, масштабованість і зручність використання платформи.

РОЗДІЛ 2. ПРОЄКТУВАННЯ САЙТУ З ПОШУКУ РОБОТИ З  
ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

2.1. Визначення варіантів використання та структури системи

Для забезпечення функціональності сайту пошуку роботи, були реалізовані модулі для управління вакансіями, обробки резюме, а також комунікації між роботодавцями та шукачами роботи. Сайт надає можливість зберігати дані про вакансії, кандидатів, роботодавців, що значно спрощує процес пошуку роботи та підбору персоналу. Основний функціонал включає перегляд, публікація та редагування вакансій, перегляд резюме а також автоматизовану обробку текстових даних за допомогою штучного інтелекту, реалізованого через Gemini AI.

Типи користувачів сайту

На сайті передбачені різні ролі користувачів із відповідними правами доступу (табл. 2.1).

Таблиця 2.1

Тип користувача	Короткий опис
Адміністратор	Керує сайтом, включаючи управління користувачами, модерацію контенту та перевірку вакансій.
Роботодавець	Має можливість публікувати вакансії, переглядати резюме та взаємодіяти з кандидатами.
Шукач роботи	Може, переглядати вакансії, подавати відгуки на вакансії, отримувати поради по ваканції, щоб мати більше шансів отримати бажану вакансію та завантажувати своє резюме.

На (рис. 2.1) зображено побудовану діаграму базових варіантів використання для кожного типу користувачів.

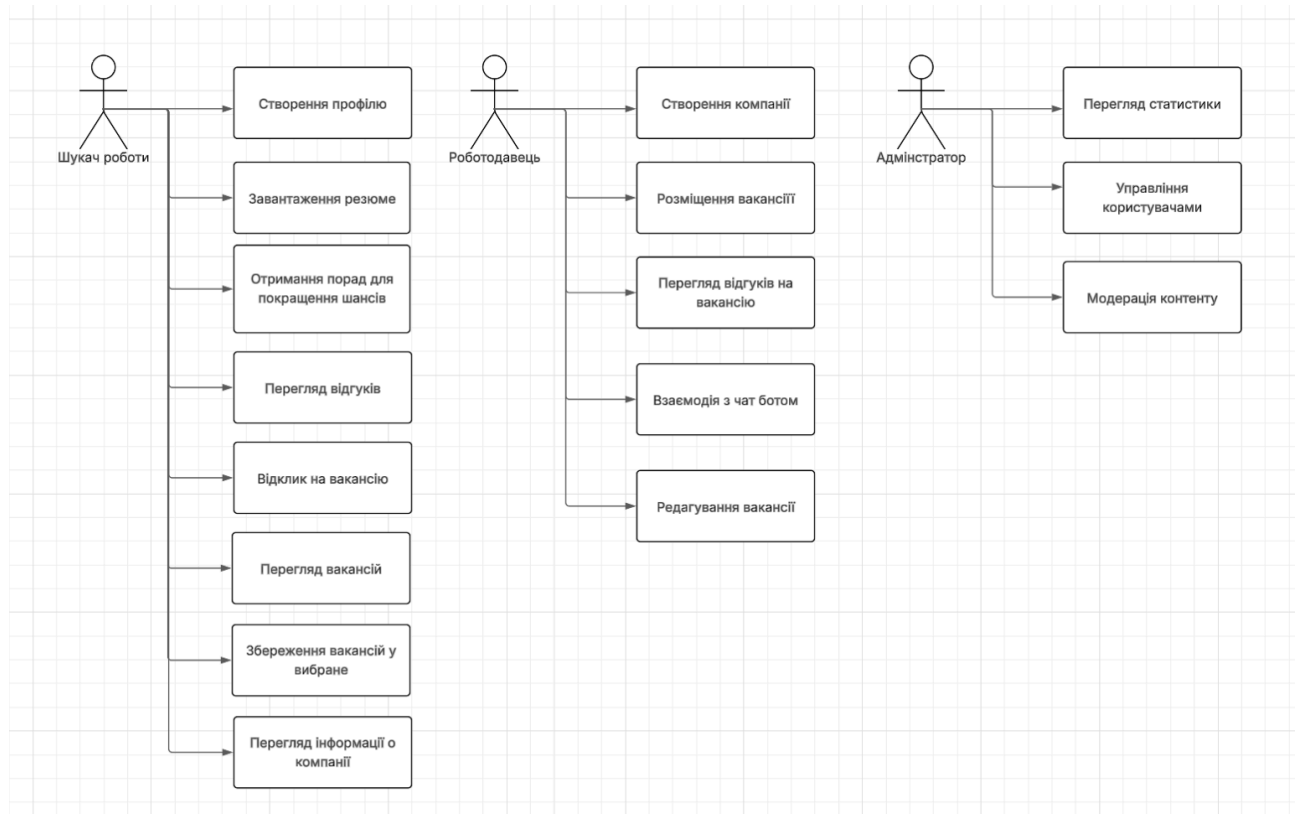


Рис 1.1. Варіанти використання сайту по пошуку роботи з використанням штучного інтелекту

Основними функціями користувачів сайту, які мають ролі шукачів роботи, є такі:

1. Можливість створення та редагування профілю користувача для демонстрації своїх професійних навичок та досвіду.
2. Зручний пошук вакансій за ключовими словами, містом та категорією, що дозволяє швидко знайти потрібну роботу.
3. Подача заявок на вакансії через простий та інтуїтивний інтерфейс.
4. Перегляд історії поданих заявок для відстеження прогресу у процесі працевлаштування.
5. Завантаження резюме у систему для його використання при подачі заявок.

6. Можливість переглядати інформацію про роботодавців, включаючи їх відгуки, рейтинг, та розмір компанії, для прийняття обґрунтованого рішення.
7. Збереження цікавих вакансій у список "Вибране" для зручного доступу в майбутньому.
8. Відстеження статусу поданих заявок, включаючи етапи розгляду заявки роботодавцем.
9. Отримання персоналізованих порад щодо підвищення шансів на працевлаштування, рекомендацій з покращення резюме або підготовки до співбесіди на основі аналізу даних за допомогою штучного інтелекту.

Основними функціями користувачів сайту, які мають ролі роботодавців, є такі:

1. Створення та редагування вакансій, включаючи опис роботи, вимоги до кандидатів та умови праці.
2. Перегляд кандидатів, які подали заявки на вакансії, та можливість сортування їх за статусом або іншими критеріями.
3. Використання штучного інтелекту для попереднього відбору резюме, щоб скоротити час на ручний перегляд.
4. Перегляд аналітики щодо вакансій, таких як кількість переглядів, поданих заявок та ефективність публікацій.
5. Створення облікового запису компанії з додаванням логотипу, опису та іншої інформації для залучення кандидатів.
6. Публікація вакансій з можливістю налаштування рівня приватності, включаючи приховування компанії, якщо це потрібно.

Основними функціями користувачів сайту, які мають роль адміністратора, є такі:

1. Модерація контенту, включаючи вакансії та компанії, для забезпечення дотримання правил сайту.

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Відстеження скарг користувачів, їх перевірка та прийняття відповідних заходів для вирішення проблем.

#### **Функціональні вимоги:**

1. **Авторизація у системі:** Система повинна забезпечувати реєстрацію користувачів із присвоєнням відповідної ролі (шукач роботи, роботодавець, адміністратор).
2. **Можливість входу в систему:** Користувачі повинні мати можливість входити до системи через власний акаунт.
3. **Створення та редагування профілю:** Користувачі повинні мати можливість заповнювати, редагувати та видаляти особисту інформацію, включаючи резюме, навички.
4. **Пошук вакансій:**  
Система повинна підтримувати пошук вакансій за ключовими словами, категоріями, містом та рівнем зарплати.
5. **Створення вакансій:**  
Роботодавці повинні мати можливість створювати вакансії, додаючи детальну інформацію, таку як вимоги, умови праці, зарплата та переваги.
6. **Подання заявок:** Шукачі роботи повинні мати можливість подавати заявки на вакансії з можливістю додавання супровідного листа.
7. **Управління заявками:** Роботодавці повинні мати можливість переглядати, відхиляти або затверджувати заявки, додавати коментарі та змінювати статус заявки (розглянуто, відібрано, відхилено).
8. **Фільтрація та сортування:**  
Користувачі повинні мати можливість фільтрувати вакансії або кандидатів за різними критеріями (зарплата, місце розташування, досвід роботи).
9. **Відстеження статусу заявок:**  
Шукачі роботи повинні мати можливість переглядати статус своїх поданих заявок.

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				23
Змн.	Арк.	№ докум.	Підпис	Дата		



**10. Аналітика та звіти:** Роботодавці повинні мати доступ до аналітики щодо переглядів вакансій, кількості поданих заявок, рейтингу компанії тощо.

**11. Збереження в обране:** Користувачі повинні мати можливість додавати вакансії або профілі до списку "Обраного" для подальшого перегляду.

**12. Модерація контенту:**  
Адміністратори повинні мати інструменти для модерації вакансій, резюме та коментарів.

**13. Поради для шукачів роботи на основі штучного інтелекту:** Сайт повинен надавати персоналізовані поради шукачам роботи щодо покращення резюме та підготовки до співбесід на основі аналізу їхніх профілів.

**14. Чат з штучним інтелектом для роботодавців:** Роботодавці повинні мати можливість взаємодіяти з чат-ботом на основі штучного інтелекту для аналізу резюме кандидатів, отримання рекомендацій щодо відбору найкращих кандидатів та оптимізації процесу рекрутингу.

## **Нефункціональні вимоги:**

### **1. Сприйняття**

- Час, необхідний для навчання звичайних користувачів – 2-3 години, а для досвідчених користувачів – до 1 години.
- Час відгуку для виконання типових задач (пошук вакансій, перегляд резюме) – не більше 10 секунд; для складних завдань (генерація рекомендацій, аналітика) – не більше 20 секунд.
- Інтерфейс повинен бути інтуїтивно зрозумілим, із детальними підказками та прикладами використання.

### **2. Надійність**

- Доступність системи повинна становити не менше 98%, з максимально допустимим часом обслуговування – до 2% загального часу роботи.

		Семериков Н.О.			ДУ «Житомирська політехніка». 25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

- Середній час безвідмовної роботи системи повинен бути не менше 30 робочих днів.
- Максимально допустима норма помилок – 1 помилка на 20 000 рядків коду.

### 3. Продуктивність

- Система повинна підтримувати одночасну роботу мінімум 1000 активних користувачів.
- Обробка запитів до бази даних повинна виконуватися за час, що не перевищує 5 секунд для простих запитів та 10 секунд для складних.
- Максимальний час завантаження сторінки – до 3 секунд при стандартній швидкості з'єднання.

### 4. Можливість експлуатації

- Масштабованість:  
Система повинна підтримувати збільшення кількості користувачів без зниження продуктивності шляхом додавання серверних потужностей.
- Адаптивність:  
Інтерфейс повинен коректно працювати на різних пристроях (десктопи, смартфони, планшети) та підтримувати всі сучасні браузері.

### 5. Безпека

- Дані користувачів повинні бути зашифровані під час зберігання та передачі.
- Необхідно впровадити багаторівневу систему автентифікації (паролі).
- Регулярне резервне копіювання даних з мінімальною частотою раз на добу.

Для розробки сайту були використані наступні технології: фреймворк Next.js для побудови клієнтської та серверної частини, база даних MongoDB для зберігання даних, а також API Gemini AI та Ai SDK від Vercel для реалізації функцій автоматичного аналізу текстових даних резюме та вакансій. Сайт побудований з урахуванням функціонального підходу, який дозволяє

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

реалізовувати бізнес-логіку у вигляді окремих функцій без прив'язки до об'єктно-орієнтованих структур.

Діаграма структури компонентів сайту (рис. 2.2) відображає зв'язки між основними функціональними блоками та потоками даних, які забезпечують реалізацію функцій пошуку роботи, управління вакансіями, обробки резюме та взаємодії між кандидатами і роботодавцями.

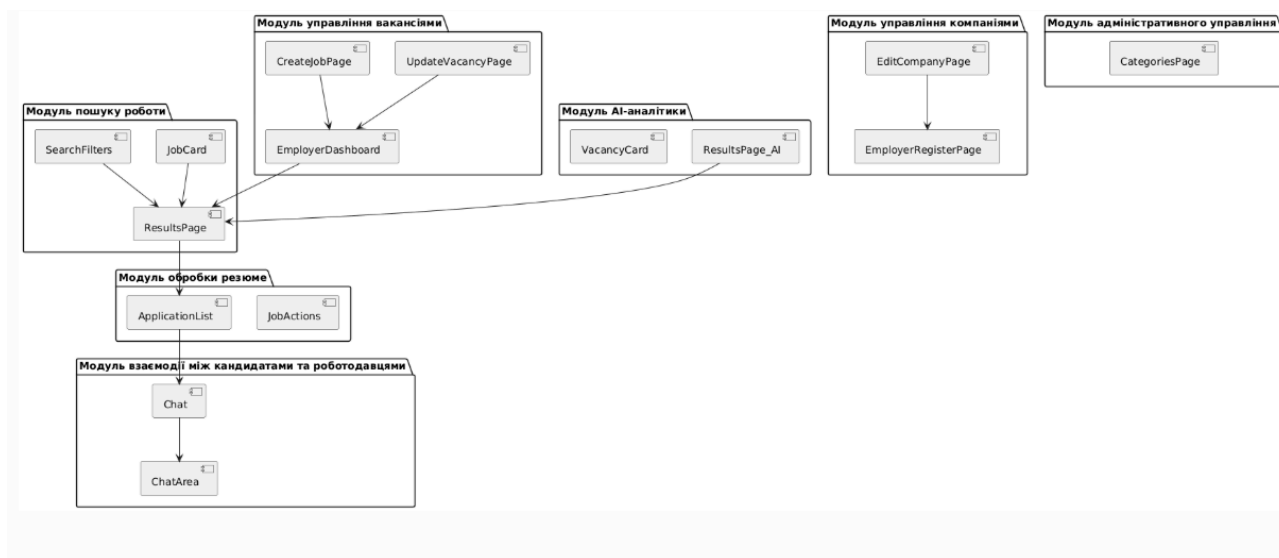


Рис. 2.2 Діаграма структури компонентів сайту

Модуль Пошук роботи відповідає за фільтрацію, пошук та відображення вакансій, де основними компонентами є фільтри пошуку, картки вакансій та сторінка результатів. Через фільтрацію користувачі можуть швидко знайти потрібні вакансії за містом, зарплатою або категорією.

Модуль Управління вакансіями дозволяє роботодавцям створювати, редагувати та деактивувати вакансії. Компоненти цього модуля пов'язані з панеллю управління роботодавця, де здійснюється контроль над усіма створеними вакансіями.

Модуль Обробки резюме реалізує функціонал завантаження резюме кандидатами, подання заявок на вакансії та відображення списку поданих заявок для роботодавців. Він тісно пов'язаний із результатами пошуку вакансій і дозволяє кандидатам швидко відгукуватися на пропозиції.

		Семериков Н.О			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

Модуль Взаємодії кандидатів і роботодавців забезпечує комунікацію між користувачами через інтегрований чат. Після подання заявки кандидат може взаємодіяти з роботодавцем безпосередньо через чат-інтерфейс.

Модуль Управління компаніями дозволяє роботодавцям створювати профілі своїх компаній та редагувати основну інформацію. Він пов'язаний із реєстрацією роботодавців та управлінням вакансіями.

Модуль Адміністративного управління призначений для адміністраторів платформи та надає можливість створювати, редагувати та видаляти категорії вакансій для підтримки структури сайту.

Окремо виділено модуль ІІІ, який інтегрується з модулями пошуку, обробки резюме та управління вакансіями для аналізу відповідності кандидатів вимогам вакансій та генерації персоналізованих рекомендацій за допомогою штучного інтелекту.

Таким чином, усі модулі взаємопов'язані, що забезпечує безперервний потік даних між ними та повний цикл роботи платформи з пошуку роботи.

## 2.2. Розробка бази даних системи

Дані в системі зберігаються у базі даних MongoDB, яка забезпечує гнучке зберігання та ефективний доступ до інформації. Основними колекціями бази даних є 8 колекцій, які реалізують основний функціонал сайту з пошуку роботи (табл. 2.2).

Таблиця 2.2

Назва колекції	Призначення
User	Колекція користувачів
Company	Колекція компаній
JobVacancy	Колекція вакансій
Application	Колекція заявок на вакансії

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

Category	Колекція категорій вакансій
Favorite	Колекція обраних вакансій
Message	Колекція повідомлень чату з ІІІ
Resume	Зберігання резюме користувачів (інкапсульовано у User)

Структура бази даних наочно представлена на (рис. 2.3).

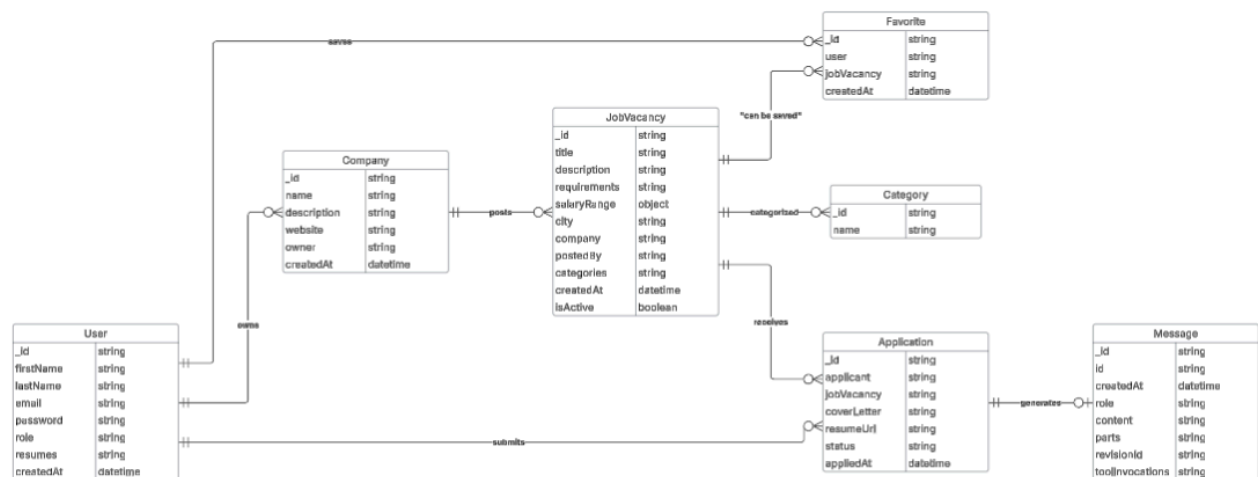


Рис 2.3 Структура бази даних

Колекція User слугує для збереження даних користувачів системи. Опис основних полів наведено у таблиці 2.3.

Таблиця 2.3

Назва поля	Тип даних	PK	FK	Опис
id	ObjectId	+	-	Унікальний ідентифікатор користувача
firstName	String	-	-	Ім'я користувача
lastName	String	-	-	Прізвище користувача
email	String	-	-	Електронна пошта
password	String	-	-	Пароль користувача (захищений)
role	String	-	-	Роль користувача (шукач роботи, роботодавець, адміністратор)
resumes	Array	-	-	Масив резюме користувача
createdAt	Date	-	-	Дата реєстрації

Колекція «Company» використовується для збереження даних про компанії, які реєструються на платформі для розміщення своїх вакансій. Опис полів колекції наведено нижче (табл. 2.4).

Таблиця 2.4

Назва поля	Тип даних	PK	FK	Опис
id	ObjectId	+	-	Унікальний ідентифікатор компанії
name	String	-	-	Назва компанії
description	String	-	-	Опис діяльності компанії
website	String	-	-	Офіційний вебсайт компанії
owner	ObjectId	-	User. id	Ідентифікатор власника компанії
createdAt	Date	-	-	Дата створення профілю компанії

Колекція «JobVacancy» зберігає дані про вакансії, які створюють роботодавці. Вона є однією з ключових у системі, оскільки на пряму взаємодіє з користувачами та кандидатами. Опис структури наведено в табл. 2.5.

Таблиця 2.5

Назва поля	Тип даних	PK	FK	Опис
id	ObjectId	+	-	Унікальний ідентифікатор вакансії
title	String	-	-	Назва вакансії
description	String	-	-	Опис обов'язків та завдань
requirements	Array	-	-	Перелік вимог до кандидата
salaryRange	Object	-	-	Діапазон зарплати (мінімальна та максимальна)
city	String	-	-	Місто розміщення вакансії
company	ObjectId	-	Company. id	Компанія, яка створила вакансію
postedBy	ObjectId	-	User. id	Користувач-роботодавець
categories	Array	-	Category. id	Категорії вакансії
createdAt	Date	-	-	Дата публікації
isActive	Boolean	-	-	Статус активності вакансії

Колекція «Application» зберігає дані про подані заявки кандидатів на відкриті вакансії. Вона фіксує зв'язок між шукачем роботи та вакансією. Детальний опис полів подано в табл. 2.6.

Таблиця 2.6

Назва поля	Тип даних	PK	FK	Опис
id	ObjectId	+	-	Унікальний ідентифікатор заявки

applicant	ObjectId	-	User. id	Кандидат, який подав заявку
jobVacancy	ObjectId	-	JobVacancy. id	Вакансія, на яку подано заявку
coverLetter	String	-	-	Супровідний лист
resumeUrl	String	-	-	Посилання на резюме кандидата
status	String	-	-	Статус розгляду заявки (очікує, прийнято, відхилено)
appliedAt	Date	-	-	Дата подачі заявки

Колекція «Category» використовується для класифікації вакансій за професійними напрямками. Це допомагає покращити фільтрацію при пошуку. Опис полів — у табл. 2.7.

Таблиця 2.7

Назва поля	Тип даних	PK	FK	Опис
id	ObjectId	+	-	Унікальний ідентифікатор категорії
name	String	-	-	Назва категорії вакансій

Колекція «Favorite» містить інформацію про вакансії, які шукачі роботи зберегли у "вибране" для подальшого перегляду. Її структура подана в табл. 2.8.

Таблиця 2.8

Назва поля	Тип даних	PK	FK	Опис
_id	ObjectId	+	-	Унікальний ідентифікатор збереження
user	ObjectId	-	User. _id	Користувач, який додав у вибране
jobVacancy	ObjectId	-	JobVacancy. id	Вакансія, додана у вибране
createdAt	Date	-	-	Дата додавання

Колекція «Message» відповідає за зберігання повідомлень між користувачами та штучним інтелектом у чаті. Її використання дозволяє проводити аналіз резюме та надавати рекомендації. Опис полів — у табл. 2.9.

Таблиця 2.9

Назва поля	Тип даних	PK	FK	Опис
id	ObjectId	+	-	Унікальний ідентифікатор повідомлення
id	String	-	-	Внутрішній ідентифікатор повідомлення
createdAt	Date	-	-	Дата створення повідомлення
role	String	-	-	Роль автора повідомлення (користувач або ШІ)
content	String	-	-	Основний текст повідомлення

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

parts	Array	-	-	Частини повідомлення (наприклад, текстові блоки)
revisionId	String	-	-	Ідентифікатор ревізії
toolInvocations	Array	-	-	Викликані інструменти у процесі обробки

Використовуючи такі колекції, як User, Company та JobVacancy, було побудовано логіку для зберігання основної інформації про користувачів, роботодавців та їхні вакансії, із розподіленням доступів відповідно до ролей користувачів. Завдяки колекціям Application та Favorite реалізовано функціонал подання заявок на вакансії та можливість збереження цікавих пропозицій у "вибране". Колекції Category та Message використовуються для забезпечення зручної класифікації вакансій за напрямками діяльності та для організації чату зі штучним інтелектом, який допомагає у аналізі резюме й наданні рекомендацій шукачам роботи та роботодавцям.

### 2.3 Проєктування та реалізація алгоритмів роботи системи

Для реалізації сайту з пошуку роботи була обрана клієнт-серверна архітектура з використанням сучасного фреймворку **Next.js**. Такий підхід забезпечує ефективний розподіл навантаження між клієнтською та серверною частинами застосунку, а також підтримує серверний рендеринг (SSR) і статичну генерацію сторінок (SSG). Ключовою особливістю реалізації є застосування **Server Actions** — нового механізму Next.js, який дозволяє напряму викликати серверні функції з клієнтських компонентів без використання окремих API-ендпойнтів або RESTful запитів.

Використання Server Actions має такі переваги:

- Логіка обробки даних виконується безпосередньо на сервері без проміжного мережевого протоколу HTTP (з точки зору розробника).
- Підвищення безпеки за рахунок ізольованого виконання коду на сервері.

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				31
Змн.	Арк.	№ докум.	Підпис	Дата		



- Спрощення розробки та обслуговування системи завдяки уникненню ручного створення API-роутів.
- Оптимізація продуктивності за рахунок зменшення кількості проміжних рівнів обробки запитів.

На (рис. 2.4) представлено загальну схему взаємодії клієнта з сервером за допомогою Server Actions у Next.js.

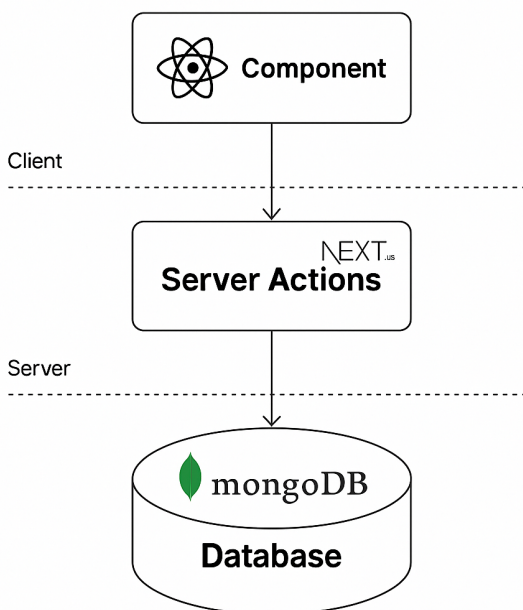


Рисунок 2.4 – Загальна схема взаємодії клієнта, серверних дій та бази даних у системі на основі Next.js

Також для ілюстрації алгоритму роботи сайту побудовано структурну діаграму (рис. 2.5), на якій зображені основні потоки виконання логіки для користувачів із різними ролями.

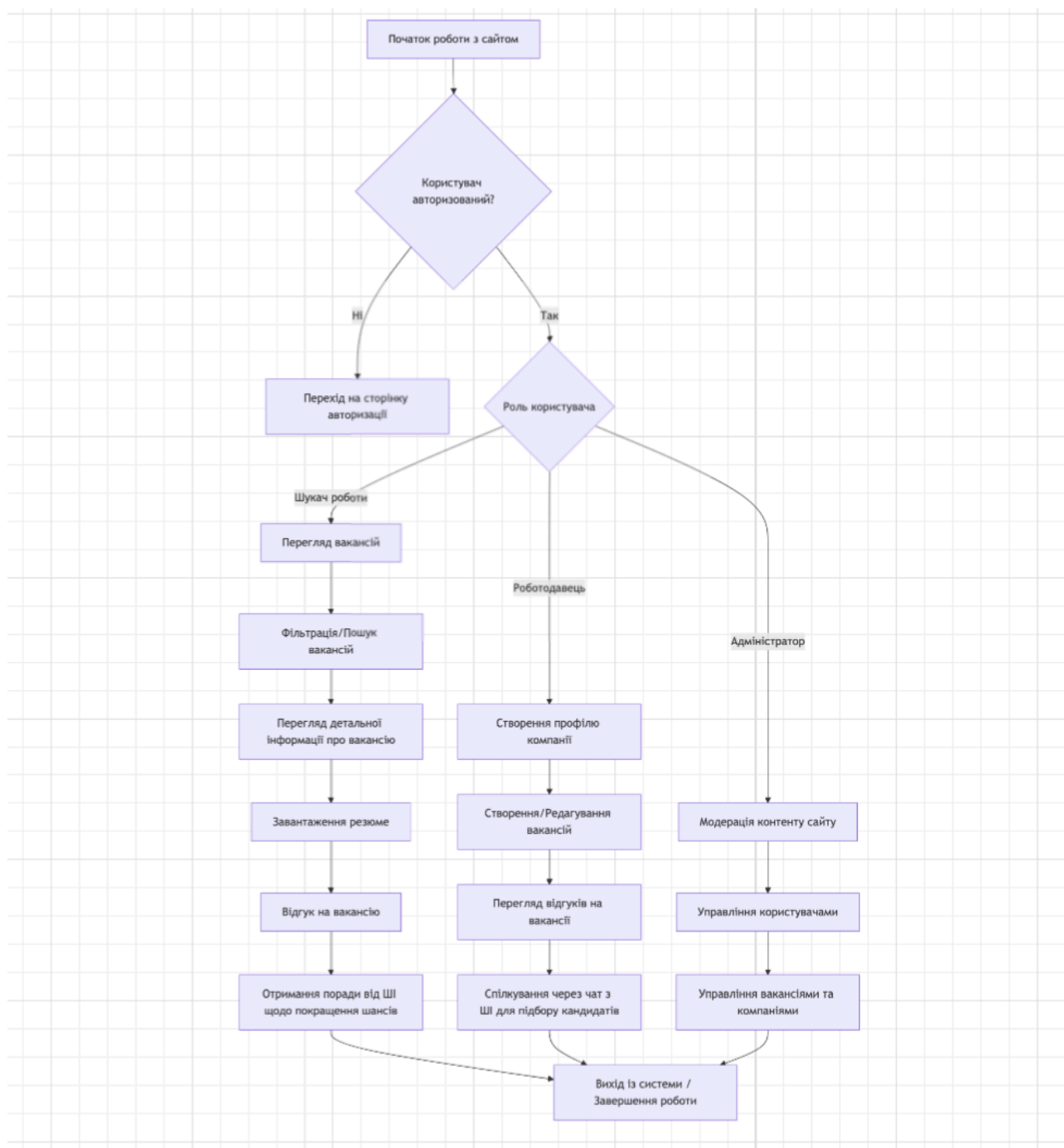


Рис 2.5 Структурна діаграма системи

### Опис отриманої діаграми:

- Перевірка авторизації користувача (якщо авторизація відсутня – перехід на сторінку входу; якщо авторизований – визначення ролі користувача).
- Отримання прав доступу відповідно до ролі:
  - Для шукача роботи – відкривається можливість перегляду та фільтрації вакансій, перегляду детальної інформації про вакансії,

завантаження резюме, подачі заявок на вакансії та отримання порад від системи штучного інтелекту.

- Для роботодавця – можливість створення профілю компанії, створення, редагування та деактивації вакансій, перегляду відгуків кандидатів, а також спілкування із ШІ у чаті для допомоги у відборі претендентів.
- Для адміністратора – надання доступу до модерації контенту сайту, управління користувачами, вакансіями та компаніями.
- Завершення роботи користувача із системою через вихід або завершення сеансу.

Також для відображення архітектурної побудови системи було побудовано діаграму компонентів (рис. 2.6), на якій зображено основні модулі клієнтської частини, серверної частини та взаємодію з базою даних.

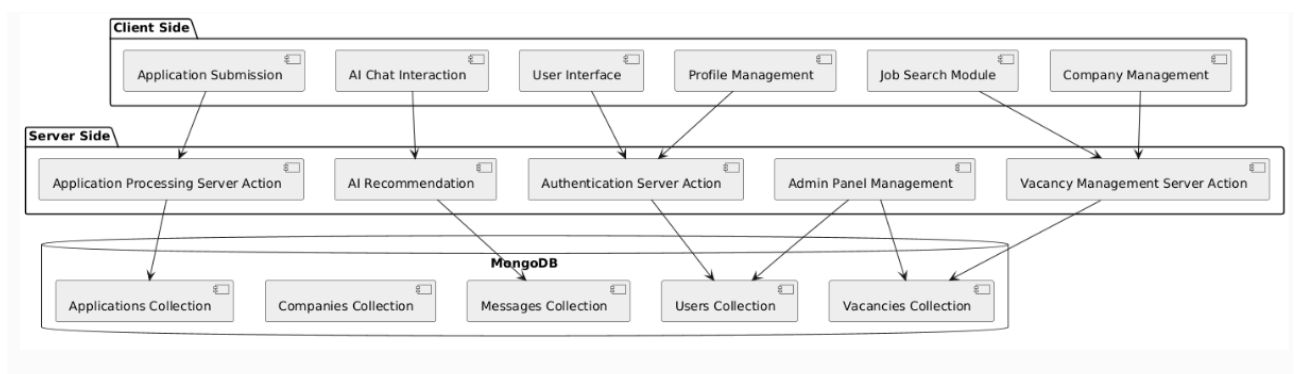


Рисунок 2.6 – Діаграма компонентів сайту

На діаграмі показано такі компоненти:

- User Interface – відповідальний за відображення інтерфейсу користувача, забезпечує взаємодію користувача з основними можливостями сайту.
- Job Search Module – модуль для пошуку вакансій за різними критеріями (ключові слова, місто, категорія, рівень зарплати).
- Profile Management – компонент, що дозволяє користувачам створювати, редагувати або видаляти свої профілі, а також завантажувати резюме.

- Application Submission – реалізує процес подання заявок на вакансії, включаючи надсилання резюме та супровідних листів.
- Company Management – модуль для створення, редагування та керування профілями компаній роботодавцями.
- AI Chat Interaction – інтеграція чату з штучним інтелектом для аналізу резюме та надання рекомендацій кандидатам і роботодавцям.
- Authentication Server Action – обробляє реєстрацію, авторизацію та перевірку прав доступу користувачів.
- Vacancy Management Server Action – відповідає за створення, редагування, деактивацію вакансій роботодавцями.
- Application Processing Server Action – обробляє подані заявки від шукачів роботи, зміну їх статусу та зберігання даних.
- AI Recommendation – обробляє запити до модуля штучного інтелекту для надання порад користувачам.
- Admin Panel Management – компонент, який дозволяє адміністраторам здійснювати модерацію контенту, управління користувачами та компаніями.
- Users Collection – зберігає інформацію про зареєстрованих користувачів, їхні профілі та завантажені резюме.
- Companies Collection – містить дані про компанії, що публікують вакансії на платформі.
- Vacancies Collection – база даних відкритих вакансій, їх описів, вимог та зарплат.
- Applications Collection – колекція заявок, поданих шукачами роботи на вакансії.
- Messages Collection – зберігає історію спілкування користувачів із системою рекомендацій на основі штучного інтелекту.

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

Далі розглянемо алгоритм роботи модуля подання відгуку на вакансію. Для наочного зображення послідовності взаємодії між основними компонентами системи було побудовано діаграму послідовності (рис. 2.7).

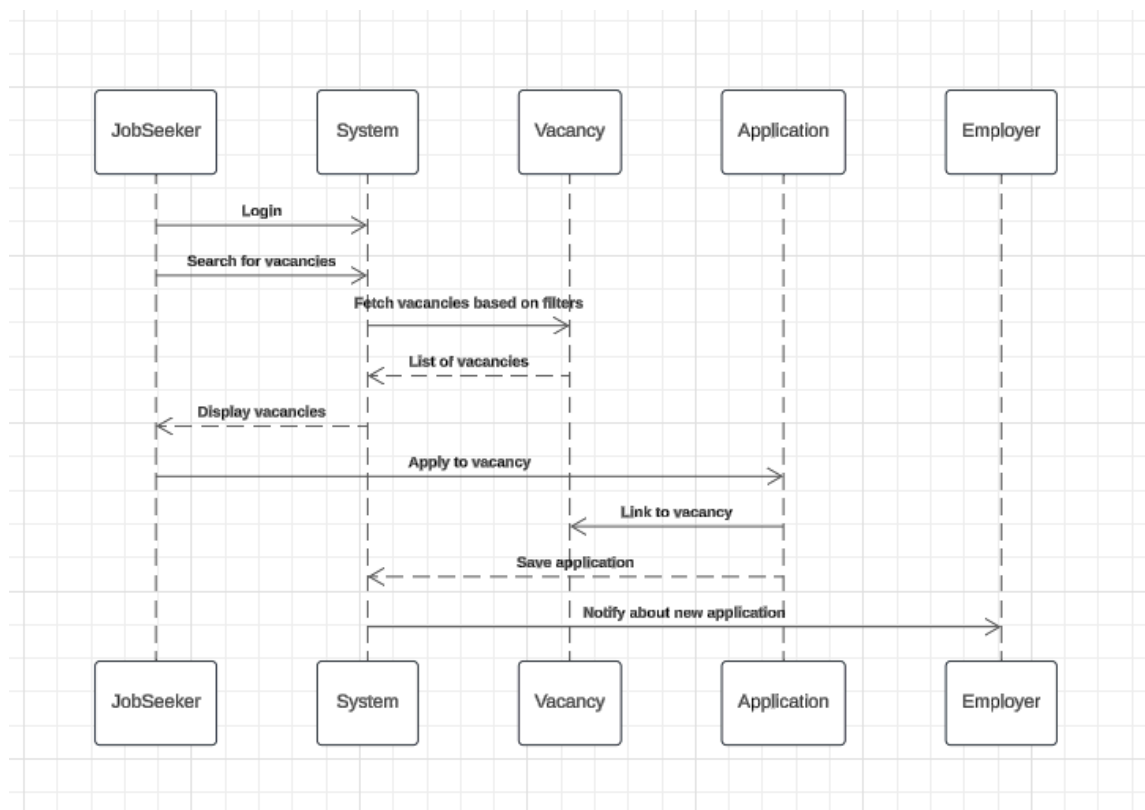


Рисунок 2.7 – Діаграма послідовності процесу подання заявки на вакансію

На даній діаграмі приведено основний потік виконання при роботі з модулем подання заявки на вакансію. Спочатку шукач роботи авторизується у системі, після чого має можливість перейти на сторінку пошуку вакансій. Користувач вводить критерії пошуку (наприклад, місто, ключові слова або категорію), після чого система здійснює пошук відповідних вакансій на основі заданих фільтрів. Отриманий список вакансій відображається користувачу для перегляду. Після вибору вакансії, користувач має можливість подати заявку, додавши своє резюме та супровідний лист. Система зберігає інформацію про подану заявку та встановлює зв'язок між заявкою і відповідною вакансією.

Після успішного створення заявки роботодавець отримує сповіщення про надходження нового відгуку для подальшого розгляду кандидата.

## 2.4 Реалізація функціоналу платформи сайту з пошуку роботи з використанням штучного інтелекту

Додаток складається з клієнтської частини, серверних функцій та панелі адміністратора. Усі частини реалізовані в одному проєкті на основі фреймворку Next.js. Серверна логіка побудована за допомогою Server Actions, що дозволяє викликати серверні функції напряму з клієнтських компонентів. Вхідною точкою додатку є компоненти маршрутизації layout.tsx та серверні функції з директивою "use server". База даних реалізована на основі MongoDB із використанням Mongoose для роботи з основними колекціями.

У даній функції реалізується підключення до бази даних MongoDB за допомогою бібліотеки mongoose. При першому виклику створюється нове з'єднання, яке кешується для подальшого використання без повторного встановлення з'єднання.

```
import mongoose, { Connection } from "mongoose";

let cachedConnection: Connection | null = null;

export async function dbConnect() {
  if (cachedConnection) {
    console.log("Using cached db connection");
    return cachedConnection;
  }
  try {
    const cnx = await mongoose.connect(process.env.MONGODB_URL!);

    cachedConnection = cnx.connection;
    console.log("New mongodb connection established");
    return cachedConnection;
  } catch (error) {
    console.log(error);
    throw error;
  }
}
```

Також у системі реалізовано middleware для контролю доступу користувачів до захищених маршрутів відповідно до їх ролі.

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

Middleware перевіряє наявність токена авторизації, отриманого через next-auth, і звіряє роль користувача з вимогами доступу до маршруту. Якщо роль користувача не відповідає потрібній для маршруту, система виконує перенаправлення на відповідну сторінку авторизації. Повний код middleware наведено в додатку А.

```
export async function middleware(req: NextRequest) {
  const token = await getToken({ req, secret: process.env.NEXTAUTH_SECRET });

  const protectedRoutes: { [key: string]: string } = {
    "/admin": "admin",
    "/employer": "employer",
    "/jobseeker": "job_seeker",
  };

  const pathname = req.nextUrl.pathname;
  const requiredRole = Object.keys(protectedRoutes).find((route) =>
    pathname.startsWith(route)
  );

  if (
    pathname.startsWith("/admin/signin") ||
    pathname.startsWith("/employer/signin") ||
    pathname.startsWith("/jobseeker/signin") ||
    pathname.startsWith("/employer/register") ||
    pathname.startsWith("/jobseeker/register")
  ) {
    return NextResponse.next();
  }

  if (requiredRole) {
    const userRole = protectedRoutes[requiredRole];

    if (!token || token.role !== userRole) {
      const redirectUrl = new URL(`/${userRole}/signin`, req.url);
      return NextResponse.redirect(redirectUrl);
    }
  }

  return NextResponse.next();
}
```

Далі розглянемо функціонал подання та обробки заявок шукачів роботи на вакансії, було створено окремий серверний модуль, що використовує можливості Server Actions у Next.js.

```
"use server";

import { dbConnect } from "@/config/db";
```

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import ApplicationModel, { IApplication } from "@models/Application";
import { authOptions } from "@app/api/auth/[...nextauth]/route";
import { getServerSession } from "next-auth";
import { IApplicationCreate } from "@types/application";

export async function createApplication(data: IApplicationCreate) {
  try {
    await dbConnect();

    const session = await getServerSession(authOptions);

    if (!session || !session.user) {
      throw new Error("User is not authenticated");
    }

    const newApplication = new ApplicationModel({
      ...data,
      applicant: session.user.id,
    });

    await newApplication.save();

    return {
      success: true,
      message: "Application submitted successfully",
    };
  } catch (error) {
    return {
      success: false,
      error: error instanceof Error ? error.message : "Unknown error",
    };
  }
}

export async function getApplicationById(applicationId: string) {
  await dbConnect();
  return await ApplicationModel.findById(applicationId);
}

export async function updateApplication(
  applicationId: string,
  data: Partial<Omit<IApplication, "applicant">>
) {
  await dbConnect();

  const session = await getServerSession(authOptions);

  if (!session || !session.user) {
    throw new Error("User is not authenticated.");
  }

  const application = await ApplicationModel.findById(applicationId);
  if (application?.applicant.toString() !== session.user.id) {

```

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				39
Змн.	Арк.	№ докум.	Підпис	Дата		



```

    throw new Error("You are not authorized to update this application.");
  }

  return await ApplicationModel.findByIdAndUpdate(applicationId, data, {
    new: true,
  });
}

export async function deleteApplication(applicationId: string) {
  await dbConnect();

  const session = await getServerSession(authOptions);

  if (!session || !session.user) {
    throw new Error("User is not authenticated.");
  }

  const application = await ApplicationModel.findById(applicationId);
  if (application?.applicant.toString() !== session.user.id) {
    throw new Error("You are not authorized to delete this application.");
  }

  return await ApplicationModel.findByIdAndDelete(applicationId);
}

export async function getApplicationsByJobVacancyId(vacancyId: string) {
  try {
    await dbConnect();

    const applications = await ApplicationModel.find({
      jobVacancy: vacancyId,
    })
      .populate("applicant", "firstName lastName email")
      .populate("jobVacancy", "title")
      .lean();

    if (!applications || applications.length === 0) {
      return {
        success: false,
        error: "No applications found for this job vacancy.",
      };
    }

    const plainApplications = applications.map((application) => ({
      ...application,
      _id: application._id.toString(),
      applicant: {
        ...application.applicant,
        _id: application.applicant._id.toString(),
      },
      jobVacancy: {
        ...application.jobVacancy,
        _id: application.jobVacancy._id.toString(),
      },
    }));
  }
}

```

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    },
    appliedAt: application.appliedAt.toISOString(),
  }));

  return {
    success: true,
    data: plainApplications,
  };
} catch (error) {
  return {
    success: false,
    error:
      error instanceof Error ? error.message : "An unknown error occurred",
  };
}
}

```

Для реалізації функціоналу подання та обробки заявок шукачів роботи на вакансії було створено окремий серверний модуль, що використовує можливості Server Actions у Next.js.

У цьому файлі описані всі основні функції для роботи з заявками, а саме `createApplication`, `getApplicationById`, `updateApplication`, `deleteApplication` та `getApplicationsByJobVacancyId`. Функція `createApplication` використовується для створення нової заявки, де автоматично прив'язується ідентифікатор авторизованого користувача та зберігається інформація у колекції бази даних. `getApplicationById` дозволяє отримати детальну інформацію про конкретну заявку за її унікальним ідентифікатором, тоді як `updateApplication` забезпечує можливість редагування заявки лише її автором після проходження перевірки прав доступу. Видалення заявки реалізовано у функції `deleteApplication`, що також передбачає перевірку авторизації користувача. Функція `getApplicationsByJobVacancyId` виконує пошук усіх заявок на певну вакансію та збагачує їх даними про заявника і вакансію через механізм `populate`, повертаючи їх у зручному для обробки форматі. Усі функції модуля працюють із попереднім встановленням з'єднання з базою даних за допомогою `dbConnect`, що забезпечує стабільну роботу із збереженням кешованого з'єднання.

		Семериків Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

Для реалізації механізму автентифікації користувачів у системі використано бібліотеку NextAuth з провайдером CredentialsProvider. У цьому модулі описана логіка обробки даних для входу в систему через електронну пошту, пароль та роль користувача. Функція authorize виконує підключення до бази даних MongoDB, знаходить користувача за введеною електронною поштою, перевіряє правильність пароля за допомогою бібліотеки bcrypt, а також звіряє роль користувача для запобігання несанкціонованому доступу. При успішній автентифікації створюється JWT-токен, який містить основну інформацію про користувача (ID, ім'я, прізвище, email та роль) і використовується для управління сесією під час подальшої роботи з сайтом. Повний код реалізації механізму автентифікації наведено в додатку А.

```
export const authOptions: NextAuthOptions = {
  providers: [
    CredentialsProvider({
      name: "Credentials",
      credentials: {
        email: { label: "Email", type: "email" },
        password: { label: "Password", type: "password" },
        role: { label: "Role", type: "text" },
      },
      async authorize(credentials) {
        if (!credentials) {
          return null;
        }

        await dbConnect();

        const user = await UserModel.findOne({ email: credentials.email });

        if (user) {
          const isPasswordValid = await bcrypt.compare(
            credentials.password,
            user.password
          );
          if (isPasswordValid) {
            if (user.role !== credentials.role) {
              throw new Error("Role mismatch");
            }

            return {
              id: user._id.toString(),
              email: user.email,
              firstName: user.firstName,
              lastName: user.lastName,
            };
          }
        }
      },
    })
  ],
}
```

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        role: user.role,
      };
    }
  }
  return null;
},
}),
],
...

```

Для реалізації функціоналу рекомендацій та аналізу резюме роботодавцями були створені окремі серверні кінцеві точки на базі Next.js із використанням Google Gemini 1.5 та AI SDK. У папці `app/api` реалізовано два маршрути: перший виконує аналіз відповідності одного резюме вимогам вакансії, другий здійснює порівняння кількох резюме для вибору найкращого кандидата. Обробка запитів відбувається виключно на серверній стороні, що забезпечує безпечну взаємодію з персональними даними кандидатів. Передані резюме у форматі PDF аналізуються моделлю штучного інтелекту, яка формує рекомендації для покращення шансів кандидата або допомагає роботодавцю прийняти рішення. Повний код реалізації інтеграції з штучним інтелектом наведено у додатку А.

```

let { messages, id, jobVacancyId } = await req.json();

if (!jobVacancyId) {
  const chat = await getChatById(id);

  if (!chat) {
    throw new Error("Chat not found");
  }
  jobVacancyId = chat.jobVacancy;
}

const [vacancy, jobVacancyApplications] = await Promise.all([
  getJobVacancyById(jobVacancyId),
  getApplicationsByJobVacancyId(jobVacancyId),
]);

if (!vacancy.success || !vacancy.data) {
  throw new Error("Vacancy not found");
}
if (
  !jobVacancyApplications.success ||
  !jobVacancyApplications.data?.length
) {

```

		Семериков Н.О			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    throw new Error("No resumes found");
  }

  const resumeFiles = jobVacancyApplications.data
    .filter((application) => application.resumeUrl)
    .map((application) => ({
      type: "file",
      data: new URL(application.resumeUrl),
      mimeType: "application/pdf",
    }));

  const requestMessages = [
    {
      role: "system",
      content:
        "Ти фахівець з управління персоналом. Проаналізуй надані резюме для цієї  
вакансії і вкажи, який кандидат найкраще підходить.",
    },
    {
      role: "user",
      content: `Назва вакансії: "${vacancy.data.title}".  
Опис: "${vacancy.data.description}".  
Вимоги: "${vacancy.data.requirements.join(", ")}".`,
    },
    {
      role: "user",
      content: [
        { type: "text", text: "These are the submitted resumes:" },
        ...resumeFiles,
      ],
    },
  ],
  ...messages,
];

const result = streamText({
  model: google("gemini-1.5-pro-latest"),
  messages: requestMessages,
  experimental_generateMessageId: createIdGenerator({
    prefix: "msgs",
    size: 16,
  }),
  async onFinish({ response }) {
    await saveChat({
      id,
      messages: appendResponseMessages({
        messages: messages,
        responseMessages: response.messages,
      }),
    });
  },
});

```

		Семериков Н.О.			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

## Висновки до розділу 2

У другому розділі було здійснено проектування та реалізацію основних компонентів сайту з пошуку роботи з використанням штучного інтелекту. Було визначено варіанти використання системи для різних ролей користувачів, розроблено структуру бази даних на основі MongoDB та побудовано основні модулі клієнтської та серверної частин за допомогою Next.js і Server Actions. Реалізовано ключовий функціонал для реєстрації, пошуку вакансій, подання заявок, управління вакансіями та інтеграції модуля штучного інтелекту для рекомендацій і аналізу резюме. Створена архітектура системи забезпечує гнучкість, масштабованість та високу якість користувацького досвіду, що є основою для подальшого розвитку та удосконалення платформи.

		Семериков Н.О			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

ДЖЕРЕЛА ІНФОРМАЦІЇ

		Семериков Н.О			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

# ДОДАТКИ

		Семериков Н.О			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				47
Змн.	Арк.	№ докум.	Підпис	Дата		



		Семериков Н.О			ДУ «Житомирська політехніка».25.121.21.000 - ПЗ	Арк.
		Громський Н.О.				48
Змн.	Арк.	№ докум.	Підпис	Дата		