

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра мультимедійних систем

Розробка соціального веб-застосунку для пошуку роботи з використанням  
React та Node.js  
Текстова частина до курсової роботи за спеціальністю 121 «Інженерія  
програмного забезпечення»

Керівник курсової роботи  
ст. викладач Борозенний С. О.  
\_\_\_\_\_ (підпис)  
“ \_\_\_\_ ” \_\_\_\_\_ 2024 р.

Виконала студентка  
Лиса Анастасія Миколаївна  
“ \_\_\_\_ ” \_\_\_\_\_ 2024 р.

Київ 2024

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра мультимедійних систем і технологій

ЗАТВЕРДЖУЮ

Зав.кафедри мультимедійних систем,  
доцент, к.ф-м.н.

\_\_\_\_\_ О. П. Жежерун (підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студентці Лисій Анастасії Миколаївні факультету інформатики 3 курсу

ТЕМА: Розробка соціального веб-застосунку для пошуку роботи з

використанням React та Node.js

Зміст ТЧ до курсової роботи:

Зміст

Анотація

Вступ

1. Огляд наявних рішень
2. Огляд використаних технологій
3. Схема бази даних
4. Функціональність застосунку

Висновок

Список використаних джерел

Дата видачі “ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

Керівник \_\_\_\_\_ (підпис)

Завдання отримав \_\_\_\_\_ (підпис)

**Календарний план виконання роботи**

№	Назва етапу курсового проекту (роботи)	Термін виконання етапу
1	Отримання завдання на курсову роботу	20.09.2023
2	Визначення актуальності та аналіз наявних рішень	10.01.2024
3	Проектування бази даних та продумування архітектури застосунку	17.01.2024
4	Розробка веб-застосунку	25.01.2024
5	Написання текстової частини курсової роботи	02.04.2024
6	Створення презентації для доповіді	06.05.2024

Лиса А. М. \_\_\_\_\_

Борозенний О. С. \_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_

## Зміст

АНОТАЦІЯ .....	5
ВСТУП .....	6
РОЗДІЛ 1. ОГЛЯД НАЯВНИХ РІШЕНЬ .....	7
1.1 WORK.UA .....	7
1.2 LOBBY X .....	7
РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ .....	9
2.1 REACT .....	9
2.1.1 Особливості React .....	9
2.1.2 React Router .....	10
2.1.3 Порівняння Angular, Vue та React .....	11
2.2 NODE.JS .....	12
2.2.1 Особливості Node.js .....	12
2.3 EXPRESS.JS .....	13
2.4 NESTJS .....	13
2.5 TYPEORM .....	13
2.6 POSTGRESQL .....	14
РОЗДІЛ 3. СХЕМА БАЗИ ДАНИХ .....	15
3.1 ER-МОДЕЛЬ .....	15
РОЗДІЛ 4. ФУНКЦІОНАЛЬНІСТЬ ЗАСТОСУНКУ .....	16
4.1 РЕЄСТРАЦІЯ ТА АВТОРИЗАЦІЯ .....	16
4.2 СТВОРЕННЯ ПРОФІЛЮ .....	18
4.3 ПЕРЕГЛЯД ВАКАНСІЙ .....	21
4.4 ПЕРЕГЛЯД ПІДРОЗДІЛІВ .....	22
4.5 ФУНКЦІОНАЛЬНІСТЬ ШУКАЧА .....	23
4.5.1 Створення резюме .....	23
4.5.2 Подача заявки на вакансію .....	24
4.6 ФУНКЦІОНАЛЬНІСТЬ РОБОТОДАВЦЯ .....	24
4.6.1 Пошук кандидатів .....	24
4.6.2 Розміщення вакансій .....	26
4.6.3 Опрацювання заявок .....	26
4.7 ФУНКЦІОНАЛЬНІСТЬ АДМІНІСТРАТОРА .....	28
ВИСНОВОК .....	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	30

## Анотація

Під час написання курсової роботи було створено веб-застосунок для пошуку роботи в ЗСУ. Він покликаний спростити процес подання заявок для добровольців.

Інтерфейс веб-сайту простий та інтуїтивно зрозумілий для користувачів, дизайн лаконічний з використанням кольорів військової форми Збройних сил України. Застосунок створено за допомогою таких основних інструментів: React, Node.js, Express.js, NestJS, TypeORM, PostgreSQL.

На даному веб-сайті є можливість викладати вакансії, а також відгукуватися на них, легко шукати потрібні пропозиції щодо роботи та підбирати кандидатів.

## Вступ

Як зараз можна допомогти армії? Один з варіантів - стати її частиною. Але як це зробити? Чи можна обрати посаду або військовий підрозділ? Більшість добровольців з такими питаннями звертаються у найближчий військовий комісаріат. Даний застосунок надає можливість залишити заявку на ту чи іншу посаду не виходячи з дому. Також є можливість обрати як цивільну професію так і військову службу. Перед подачею заявки можна легко ознайомитися з усіма вимогами й ухвалити правильне рішення.

З огляду на ситуацію в Україні актуальність теми не ставиться під сумнів, тож було вирішено створити веб-застосунок який би полегшив процес набору до війська.

Головною метою створення даного продукту було максимально автоматизувати цей процес та спростити його для пересічного громадянина, а також надати можливість обирати підрозділ та посаду. Застосунок є корисним не лише для людей які шукають роботу, але й для військових підрозділів, оскільки надає можливість пошуку кандидата на ту чи іншу посаду.

Для створення додатку використовувалися популярні технології такі як React, Node.js, Express.js, NestJS та TypeORM для роботи з базою даних PostgreSQL. Процес розробки було розділено на кілька основних етапів: порівняння наявних рішень, дослідження технологій для розробки веб-сайтів, проектування бази даних, створення backend-частини та розробка веб-інтерфейсу.

## **Розділ 1. Огляд наявних рішень**

### **1.1 Work.ua**

Work.ua – це один з найвідоміших сайтів для пошуку роботи в Україні. Загалом на ньому розміщено понад 100 тисяч вакансій та більше 3 мільйонів кандидатів шукають роботу саме на Work.ua[1]. Загалом це досить зручна платформа, але є деякі недоліки саме в контексті військової служби:

- 1) Оскільки Work.ua розміщує не лише військові професії, то це може бути дещо незручно для людей, які прагнуть вступити в лави Збройних сил України.
- 2) Work.ua надає можливість пошуку військових професій саме за підрозділами, тобто щоб знайти бажану вакансію спершу потрібно обрати підрозділ, це не зручно якщо людина не знає де саме хоче служити.
- 3) На сайті немає чіткого розділення на військові професії (тобто ті, які передбачають службу в армії) та цивільні, які фактично є звичайною роботою, але на користь Збройних сил України.

Отже, Work.ua є досить корисним веб-застосунком для пошуку роботи, він надає велику функціональність, але має недоліки які ускладнюють пошук саме військових вакансій.

### **1.2 Lobby X**

Lobby X – це платформа з працевлаштування, яка спрямована на пошук фахівців до державних установ та громадських організацій [2]. Даний веб-сайт

дійсно зручний для пошуку як цивільної професії так і військової, але попри це має деякі недоліки:

- 1) На платформі немає можливості шукати професії за населеним пунктом, проте можна здійснювати пошук по спеціальності та підрозділу.
- 2) Вагомим недоліком є відсутність особистого кабінету кандидата, де користувач міг би відслідковувати свої відгуки на вакансії та резюме.

У цілому, Lobby X має гармонійний дизайн та простий зрозумілий інтерфейс, але відсутність аутентифікації може дещо ускладнити процес пошуку роботи.

## Розділ 2. Огляд використаних технологій

### 2.1 React

React – це одна з найпопулярніших JavaScript бібліотек, що використовується для створення інтерфейсів веб-застосунків.

Перший прототип даної бібліотеки побачив світ у 2010 році завдяки Джордану Воку, який на той час працював у Facebook. Лише у 2013 році React став доступним для загального використання [3].

#### 2.1.1 Особливості React

- Використання Virtual DOM: Virtual DOM – це представлення реального DOM в пам'яті, яке забезпечує швидке оновлення інтерфейсу користувача, оскільки рендерить лише ті компоненти що змінилися (див. Рис. 1) [4].

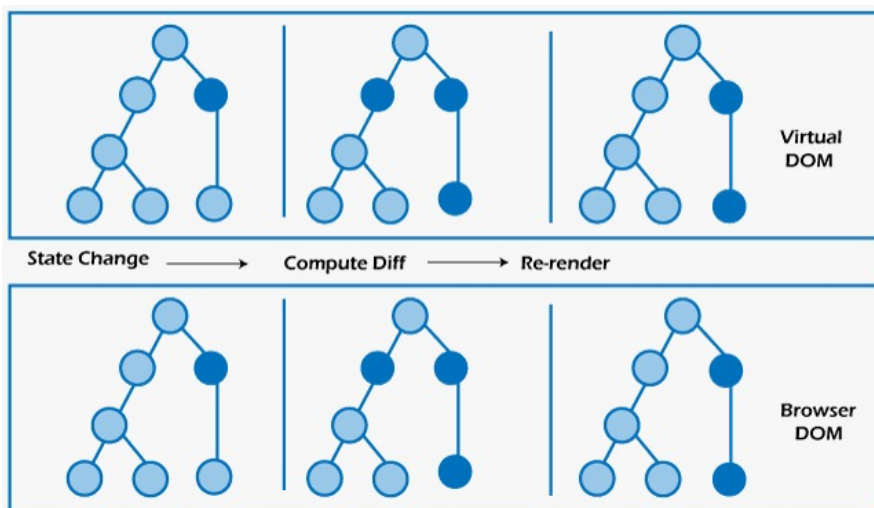


Рисунок 1. Схема роботи Virtual DOM [5]

- Розділення на окремі компоненти: Такий підхід дає можливість значно зменшити кількість дублювання коду. Розділення інтерфейсу на окремі модулі полегшує підтримку програми та значно економить час на її створення (див. Рис. 2).

```
const MyComponent = (props) => {  
  const { title = "Default Title", description = "Default Description" } = props;  
  
  return (  
    <div className="my-component-container">  
      <h1 className="my-component-title">{title}</h1>  
      <p className="my-component-description">{description}</p>  
    </div>  
  );  
};
```

Рисунок 2. Приклад створення React компонента [6]

- Підтримка SEO: React дозволяє рендерити розмітку на стороні сервера, це дає можливість пошуковим системам легко індексувати веб-застосунок. Також в React підтримуються мета теги, що забезпечує SEO просування сторінки.
- JSX: кодування JSX (JavaScript XML) дозволяє записувати HTML прямо в JavaScript і при цьому надає захист від XSS атак. Для відображення в браузері JSX компілюється в звичайний JavaScript код.
- Використання хуків: хуки – це функції, які допомагають змінювати стани компонентів і не потребують використання класів. Одні з найпоширеніших хуків: useEffect, useState, useReducer, useContext. За потреби можна також сворити власний хук [7].

### 2.1.2 React Router

React Router – це бібліотека React, головна ціль якої – спрощення маршрутизації та синхронізація url з інтерфейсом веб-застосунку. React Router потрібен для створення SPA (односторінкового додатку).

BrowserRouter – забезпечує декларативний спосіб маршрутизації використовуючи HTML5 без перезавантаження сторінки [11].

### 2.1.3 Порівняння Angular, Vue та React

Для кращої наочності порівняння оформлено як таблицю (див. Таблиця 1)

Критерії	React	Angular	Vue
Перший випуск	2013	2016	2014
Тип	Бібліотека	Фреймворк	Фреймворк
Засновник	Facebook	Google	Еван Ю. – працівник Google
DOM	Virtual DOM	Browser DOM	Virtual DOM
Написаний на	JavaScript	TypeScript	JavaScript
Легкість вивчення	Простий	Складний	Простий
Компанії, що використовують	Facebook, Instagram, WhatsApp	Microsoft Office, Gmail, PayPal	Xiaomi, Adobe, Behance
Ідеальний для	Створення невеликих користувацьких інтерфейсів	Створення масштабних проєктів з широким функціоналом	Створення невеликих користувацьких інтерфейсів

Таблиця 1. Порівняння Angular, Vue та React [8]

Загалом як React так і Angular і Vue є хорошими інструментами для створення користувацького інтерфейсу, проте мають свої недоліки. Angular є досить важким для вивчення і частіше використовується саме для великих проєктів, а Vue хоч і схожий на React проте менш популярний і має меншу кількість бібліотек. З огляду на перелічені недоліки було прийнято рішення створювати застосунок на основі React.

## 2.2 Node.js

Node.js – це середовище виконання JavaScript, яке було розроблене в 2009 році Райаном Далом. Node.js надає можливість писати код на серверній стороні.

### 2.2.1 Особливості Node.js

- Асинхронність: бібліотеки Node.js асинхронні, тому не виникає блокування, тобто сервер не очікує доки API поверне дані.
- Швидкість: Node.js створений на JavaScript Engine V8, що забезпечує високу швидкість роботи.
- Відсутність буферизації: дані не буферизуються, а виводяться фрагментами.
- Висока масштабованість: Node.js створює один потік, в якому циклічно обробляються події, що дозволяє опрацьовувати набагато більшу кількість запитів ніж традиційні сервери, які створюють певну кількість потоків. [9]

## 2.3 Express.js

Express.js – один з найпопулярніших фреймворків для Node.js. Він використовується для створення як односторінкових так і багатосторінкових сайтів та розробки backend.

Express.js надає інструменти для обробки HTTP запитів, маршрутизації, обробки помилок та NPM – інтерфейс командного рядка [12].

## 2.4 NestJS

NestJS – це фреймворк, створений для проектування об'ємних додатків, написаний на TypeScript. NestJS надає додатковий рівень абстракції над фреймворками Node.js [13].

Основні переваги NestJS: підтримка TypeScript, структурованість коду, вбудована ін'єкція залежностей [14].

## 2.5 TypeORM

ORM (Object Relational Mapper) – це частина програмного забезпечення що пов'язує дані програми з тими що знаходяться в базі даних [15].

TypeORM – це ORM, що використовується з JavaScript та TypeScript, працює на таких платформах: NodeJS, React Native, Browser, PhoneGap, Cordova, Electron [16]. TypeORM підтримує велику кількість баз даних: PostgreSQL, MySQL, SQLite, Oracle, MS SQL Server, WebSQL [17].

TypeORM значно спрощує роботу з базою даних, щоб створити сутність потрібно просто створити модель з декоратором @Entity (див. Додаток 1).

```
@Entity( options: { name: 'job-category' })
export class JobCategory extends BaseEntity {
  @PrimaryGeneratedColumn( strategy: 'uuid')
  id: string;

  @Column( options: { type: 'varchar', length: 100, unique: true })
  name: string;
```

### Додаток 1. Створення сутності в TypeORM

## 2.6 PostgreSQL

PostgreSQL – це об’єктно-реляційна система управління баз даних, що використовує мову SQL. PostgreSQL відповідає щонайменше 170 вимогам з 179 які необхідні для відповідності стандарту SQL:2023 [18]. Важливою характеристикою цієї системи управління баз даних є ACID-сумісність. Тобто виконуються основні вимоги для забезпечення надійності транзакцій: атомарність, узгодженість, ізолюваність та довговічність [19].

Основні переваги PostgreSQL[20]:

- Швидкість
- Масштабованість
- Підтримка паралельної роботи
- Підтримка багатьох мов програмування
- Відкритий код



## Розділ 4. Функціональність застосунку

### 4.1 Реєстрація та авторизація

У застосунку є можливість зареєструватися як шукач або як роботодавець (див. Рис. 4). Лише зареєстрований користувач може відгукнутися на вакансію.

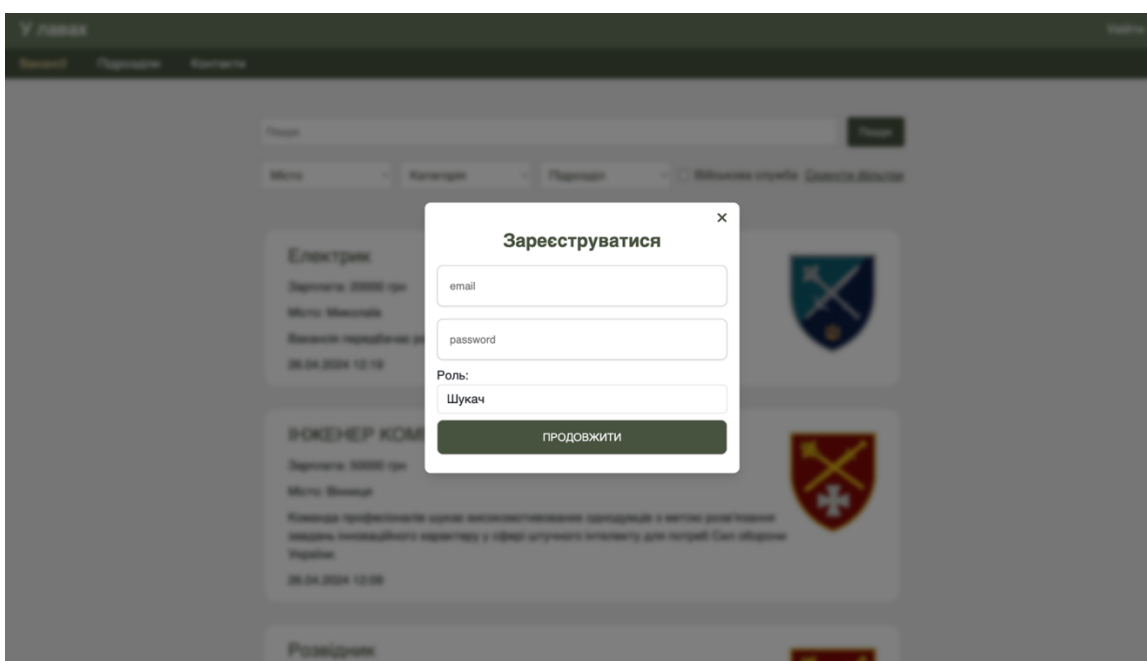


Рисунок 4. Сторінка реєстрації

Реєстрація створена з використанням бібліотеки bcrypt, що допомагає шифрувати паролі [10]. За допомогою методу hashSync (див. Додаток 2), що автоматично генерується хеш.

Хешування – це перетворення пароля на рядок з символів та чисел. Використання хешування мінімізує шанс викрадення даних та використання

їх в зловмисницьких цілях, оскільки в базі даних паролі зберігаються у захешованому вигляді й не підлягають розшифруванню.

```
@Post( path: 'registration')
async registration(@Body() authDto: AuthDto, @Res() res: Response) {
  try {
    if (await findUserByEmail(authDto)) {
      return res.status( code: 400).send( body: 'user with this email already exist');
    }
    authDto.password = bcrypt.hashSync(authDto.password, saltOrRounds: 5);
    return res.json(authDto);
  } catch (error) {
    return res.status( code: 400).send( body: 'Invalid registration');
  }
}
```

Додаток 2. Код для реєстрації

Також у застосунку є вікно аутентифікації (див. Рис. 5). При вході користувач має обрати одну з ролей: шукач, роботодавець, адміністратор (див. Рис. 6). Залежно від ролі користувачу надаються різні функціональні можливості.

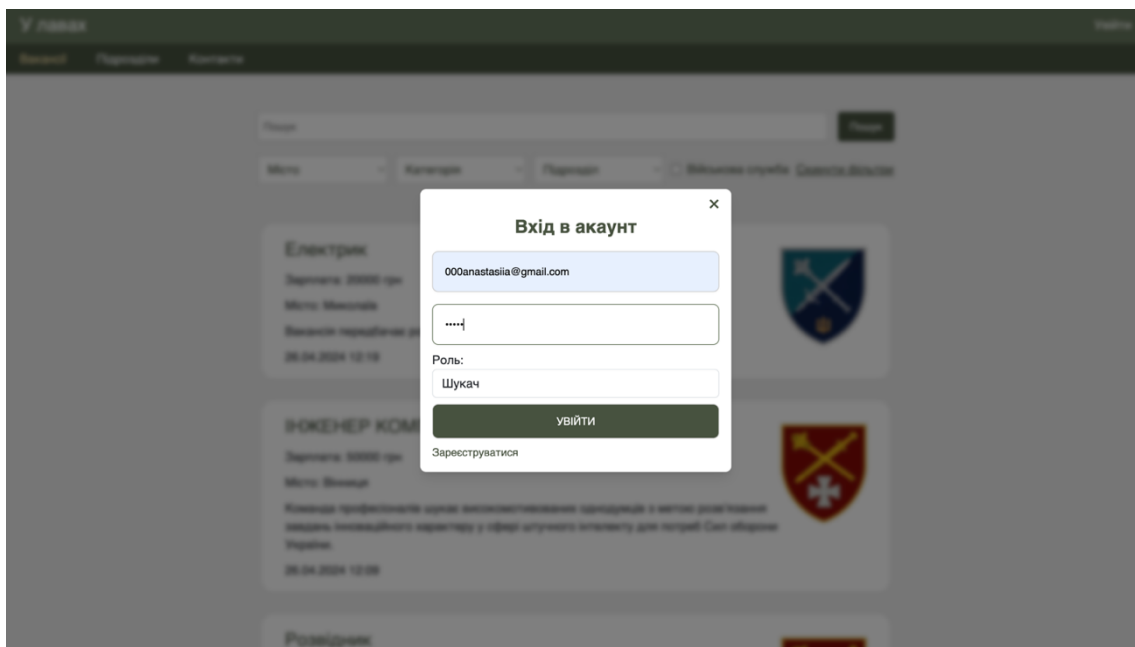


Рисунок 5. Сторінка логіну

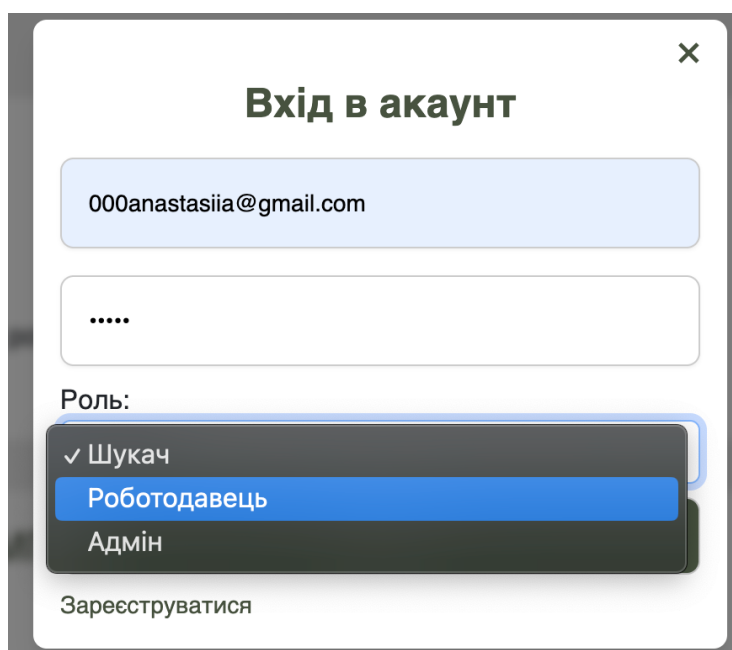
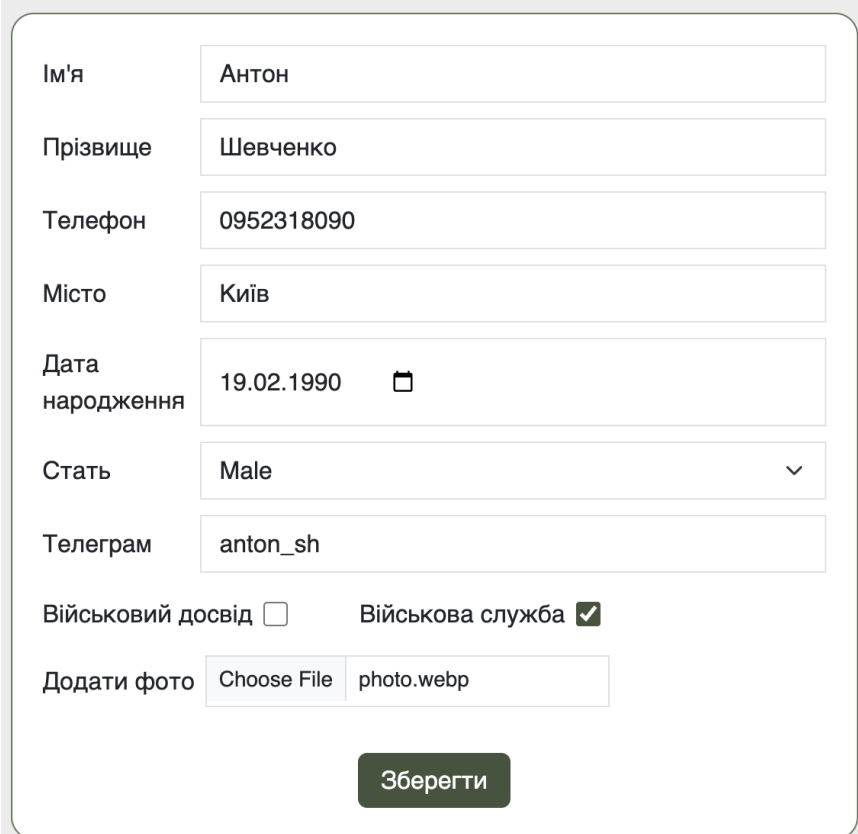


Рисунок 6. Доступні ролі користувачів

## 4.2 Створення профілю

Після реєстрації в ролі шукача, користувач має заповнити особистий профіль, за бажанням може завантажити фото (див. Рис. 7) .



The image shows a registration form for a user profile. The form is enclosed in a light gray border and contains the following fields and options:

- Ім'я:** Input field containing "Антон".
- Прізвище:** Input field containing "Шевченко".
- Телефон:** Input field containing "0952318090".
- Місто:** Input field containing "Київ".
- Дата народження:** Input field containing "19.02.1990" with a calendar icon to its right.
- Стать:** Dropdown menu with "Male" selected and a downward arrow.
- Телеграм:** Input field containing "anton\_sh".
- Військовий досвід:** A checkbox that is currently unchecked.
- Військова служба:** A checkbox that is currently checked.
- Додати фото:** A file upload section with a "Choose File" button and a text box containing "photo.webp".
- Зберегти:** A dark green button with white text, centered at the bottom of the form.

Рисунок 7. Форма для заповнення профілю шукача

Після успішного збереження користувач може переглянути свій профіль та редагувати його за потреби (див. Рис. 8) .

## Антон Шевченко

Стать: Чоловік  
Місто: Київ  
Вік: 34  
Телефон: 0952318090  
Email: anton@gmail.com  
Telegram: anton\_sh  
Військовий досвід: ✘  
Згода на військову службу: ✔

Редагувати


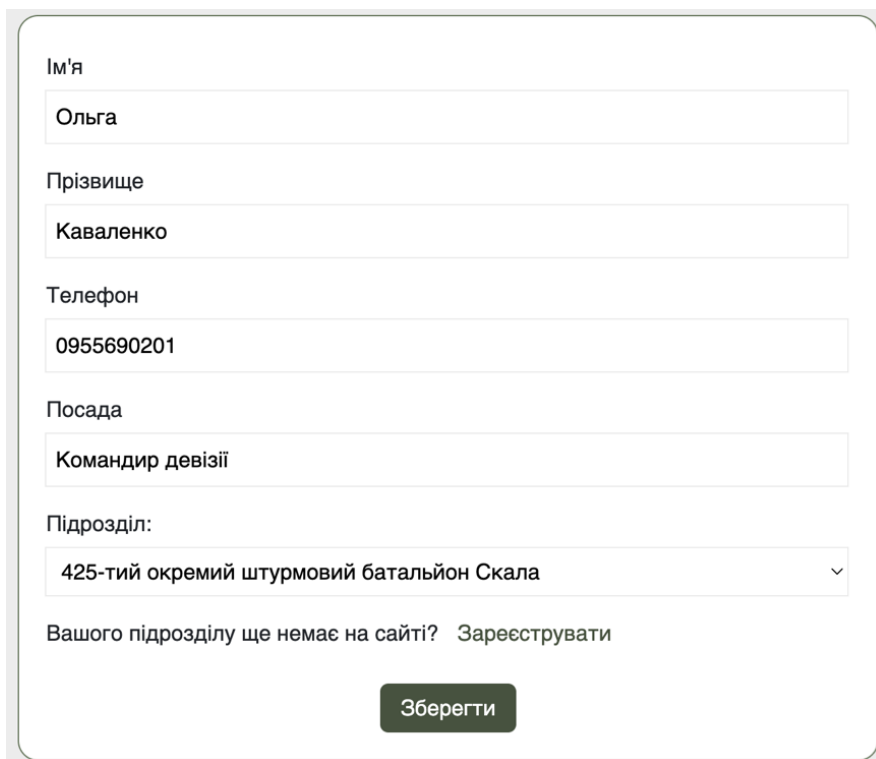


Рисунок 8. Профіль шукача

Якщо ж користувач зареєструвався як роботодавець то має заповнити профіль роботодавця (див. Рис. 9) вказавши при цьому посаду та підрозділ, якщо підрозділ ще не зареєстрований на сайті то користувач має його створити натиснувши на “Зареєструвати”.



Ім'я  
Ольга

Прізвище  
Каваленко

Телефон  
0955690201

Посада  
Командир дівізії

Підрозділ:  
425-тий окремий штурмовий батальйон Скала

Вашого підрозділу ще немає на сайті? [Зареєструвати](#)

**Зберегти**

Рисунок 9. Форма для заповнення профілю роботодавця

### 4.3 Перегляд вакансій

Переглядати вакансії можуть усі користувачі навіть незареєстровані. Також за потреби є можливість здійснити пошук вакансій по назві/опису, та фільтрувати за містом, категорією, підрозділом, військовою службою (див. Рис. 10). Детальну інформацію можна почитати натиснувши на назву вакансії.

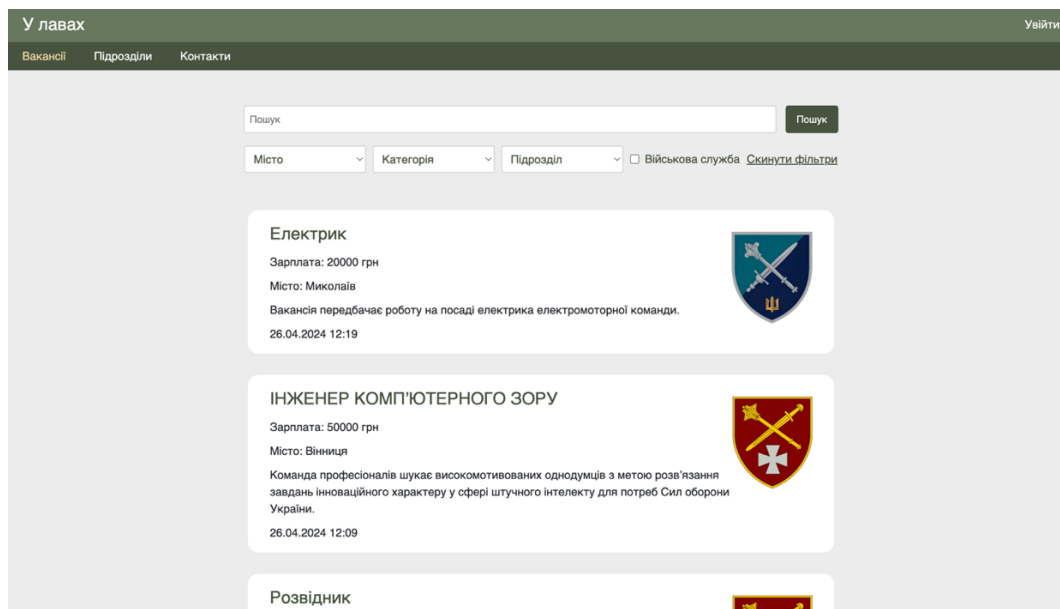


Рисунок 10. Список вакансій

#### 4.4 Перегляд підрозділів

Користувач може переглянути всі підрозділи представлені на сайті та почитати детальну інформацію про них (див. Рис. 11).

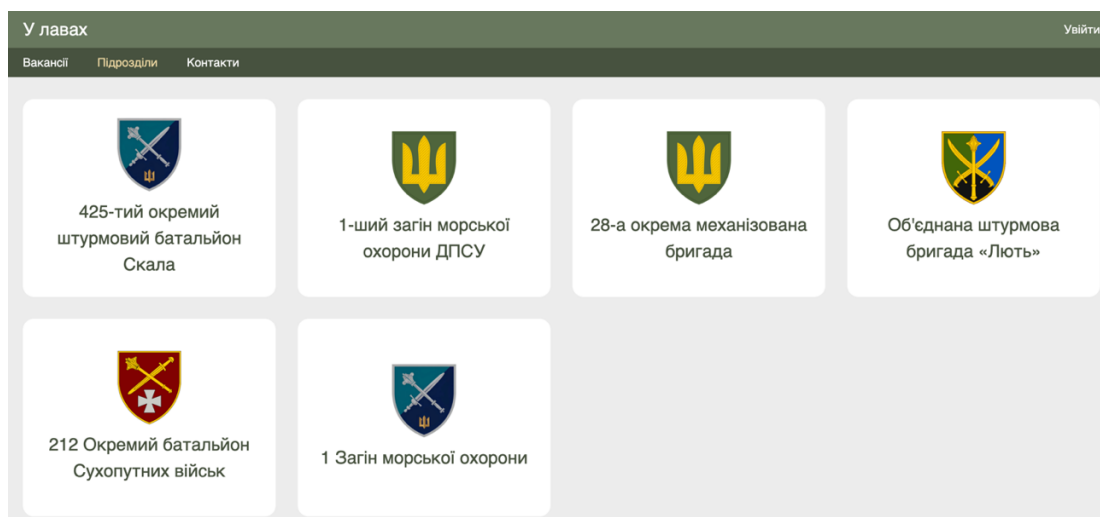


Рисунок 11. Сторінка з підрозділами

## 4.5 Функціональність шукача

### 4.5.1 Створення резюме

Щоб відгукнутися на роботу шукач має створити резюме, їх може бути декілька (див. Рис. 12). Під час створення резюме обов'язково потрібно обрати категорію та зазначити бажану посаду. Шукач може прикріпити резюме у форматі pdf або ж розписати свій досвід, освіту та навички в формі (див. Рис. 13). Також користувач може редагувати та видаляти свої резюме.

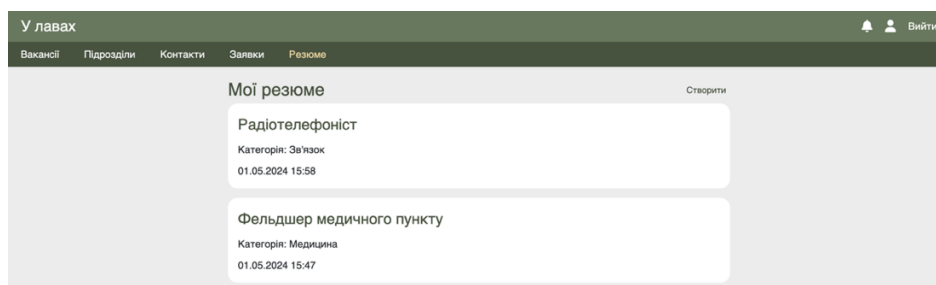


Рисунок 12. Список резюме

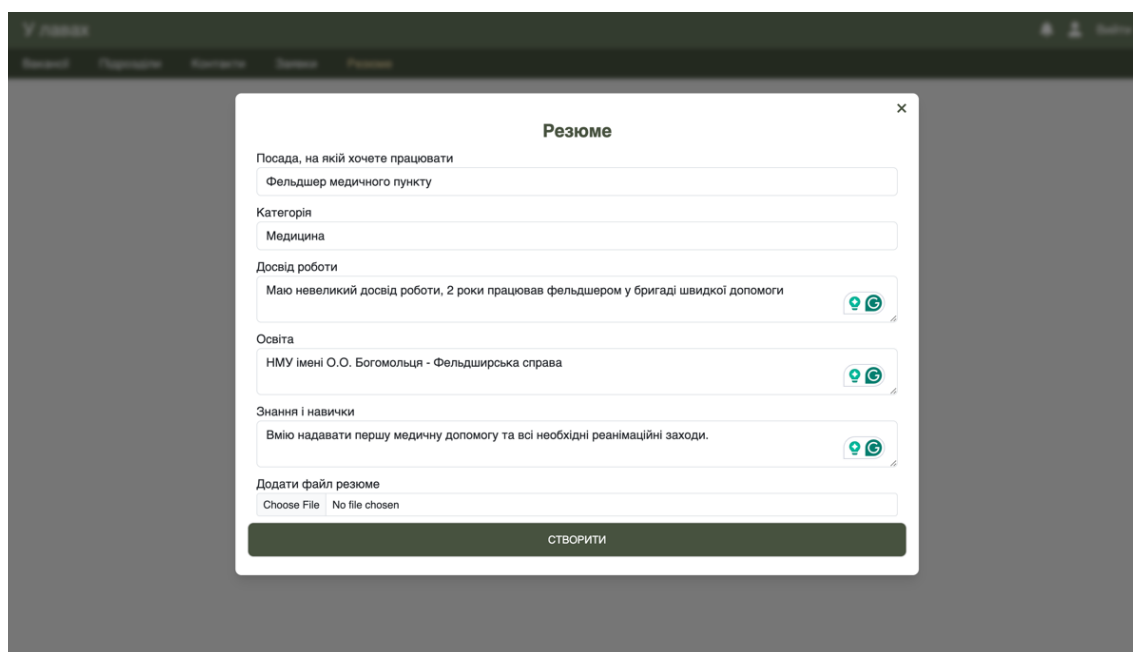
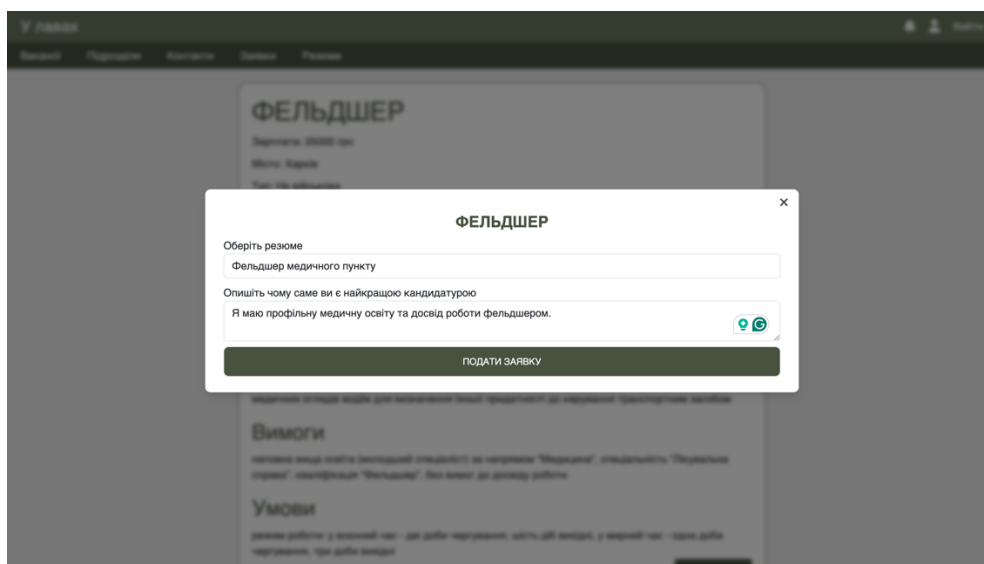


Рисунок 13. Форма створення резюме

## 4.5.2 Подача заявки на вакансію

Щоб подати заявку на ту чи іншу вакансію користувач має натиснути на кнопку “Відгукнутися”, що знаходиться в детальній інформації про вакансію, обрати резюме на подати заявку (див. Рис. 14). Також можна скасувати заявку за потреби.



The image shows a mobile application interface for applying to a job. The job title is 'ФЕЛЬДШЕР' (Paramedic). The form is titled 'ФЕЛЬДШЕР' and has a close button (X) in the top right corner. It contains the following elements:

- A dropdown menu labeled 'Оберіть резюме' (Select resume) with the selected option 'Фельдшер медичного пункту' (Paramedic of the medical point).
- A text area labeled 'Опишіть чому саме ви є найкращою кандидатурою' (Describe why you are the best candidate) with the text 'Я маю профільну медичну освіту та досвід роботи фельдшером.' (I have specialized medical education and experience as a paramedic.)
- A green button labeled 'ПОДАТИ ЗАЯВКУ' (Apply).

Рисунок 14. Форма для подачі заявки

## 4.6 Функціональність роботодавця

### 4.6.1 Пошук кандидатів

Роботодавець може переглянути резюме всіх шукачів та фільтрувати їх для пошуку потрібних кандидатів (див. Рис. 15). Якщо резюме зацікавить роботодавця то він може запропонувати вакансію відповідному користувачу через телеграм або email (див. Рис. 16).

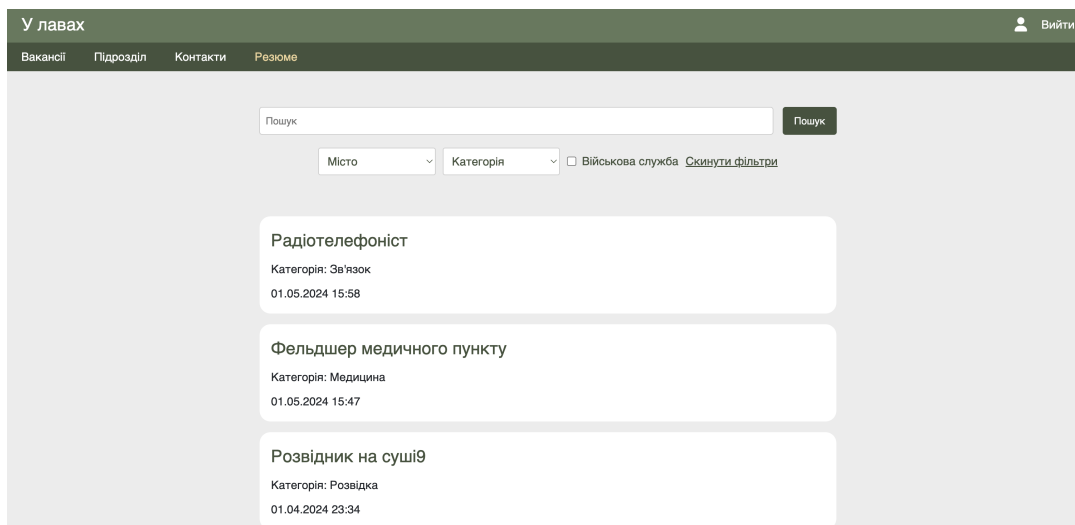


Рисунок 15. Сторінка з резюме

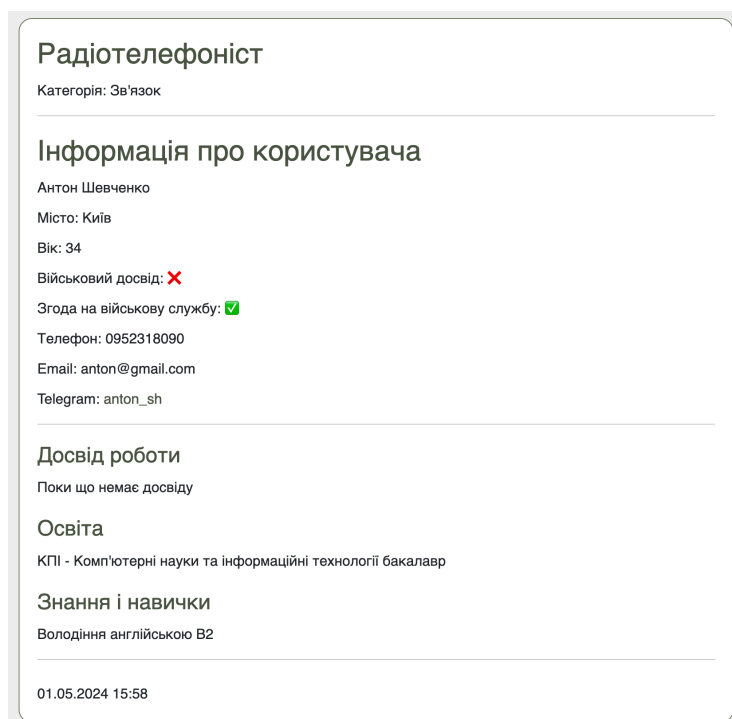


Рисунок 16. Резюме шукача

## 4.6.2 Розміщення вакансій

Одна з ключових функцій роботодавця - це розміщення вакансій. Для цього він має заповнити форму (див. Рис. 17). Після чого вакансія відразу з'явиться у переліку всіх вакансій і шукачі можуть з легкістю відгукнутися на неї.

The image shows a web form for creating a job vacancy. It contains the following fields and content:

- Назва:** Авіаційний механік
- Категорія:** Авіація
- Зарплата:** 50000
- Місто:** Одеса
- Опис:** Служба передбачає вміння працювати з авіаційною технікою та досвід технічного обслуговування
- Обов'язки:** технічне обслуговування авіаційної техніки, утримання техніки в належному стані, виконання наказів командирів, дотримання технічних норм і стандартів
- Вимоги:** наявність профільної освіти, придатність до служби за станом здоров'я, досвід військової служби буде перевагою, відсутність судимостей
- Умови:** мобілізація до кінця воєнного стану або служба за контрактом, соціальні гарантії, речове й продовольче забезпечення
- Військова служба:**

At the bottom of the form is a button labeled "Зберегти".

Рисунок 17. Форма створення вакансії

## 4.6.3 Опрацювання заявок

За необхідності представник підрозділу може редагувати та видаляти вакансії, але основне завдання - це опрацювання заявок, зробити це можна натиснувши на кнопку “Заявки” (див. Рис. 18).

## ФЕЛЬДШЕР

Редагувати
Видалити

Зарплата: 25000 грн

Місто: Харків

Тип: Не військова

Категорія: Медицина

Підрозділ: Об'єднана штурмова бригада «Лють»

### Опис

Вакансія передбачає службу у складі рятувального відділення 63-тньої державної пожежно-рятувальної частини

### Обов'язки

прийом постраждалих у зоні надзвичайної ситуації та надання першої медичної допомоги забезпечення медичного супроводу поранених, проведення передрейсових і післярейсових медичних оглядів водіїв для визначення їхньої придатності до керування транспортним засобом

### Вимоги

неповна вища освіта (молодший спеціаліст) за напрямом "Медицина", спеціальність "Лікувальна справа", кваліфікація "Фельдшер", без вимог до досвіду роботи

### Умови

режим роботи: у воєнний час - дві доби чергування, шість днів вихідні, у мирний час - одна доба чергування, три доби вихідні

25.04.2024 23:38 Заявки

Рисунок 18. Детальна інформація про вакансію

Переглянувши заявку кандидата роботодавець може прийняти або відхилити її (див. Рис. 19) та згодом зв'язатися з користувачем і домовитися про співбесіду.

## Антон Шевченко

---

### Чим відгукнулись

Резюме Фельдшер медичного пункту

Я маю профільну медичну освіту та досвід роботи фельдшером.

01.05.2024 17:02 Прийняти Відхилити

Рисунок 19. Заявка на роботу

Після опрацювання заявки її статус в кабінеті шукача зміниться (див. Рис. 20) і з'явиться сповіщення, що роботодавець прийняв чи відхилив її (див. Рис. 21).

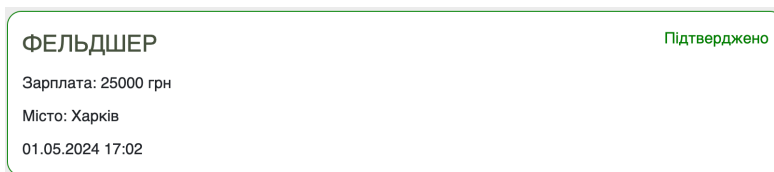


Рисунок 20. Заявка шукача зі статусом

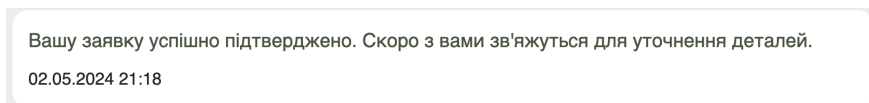


Рисунок 21. Сповідження про опрацювання заявки

## 4.7 Функціональність адміністратора

Адміністратор має доступ до майже всіх даних системи, має можливість редагувати видаляти та додавати дані за потреби (див. Рис. 22).

Головна задача – модерація контенту, а саме перевірка оголошень про вакансії та заявок на дотримання правил платформи. Також адміністратор має підтверджувати нових роботодавців попередньо зконтактувавши з ними.

Для створення адмін-панелі використано фреймворк react-admin, що працює на основі REST, GraphQL, React та Material Design [21].

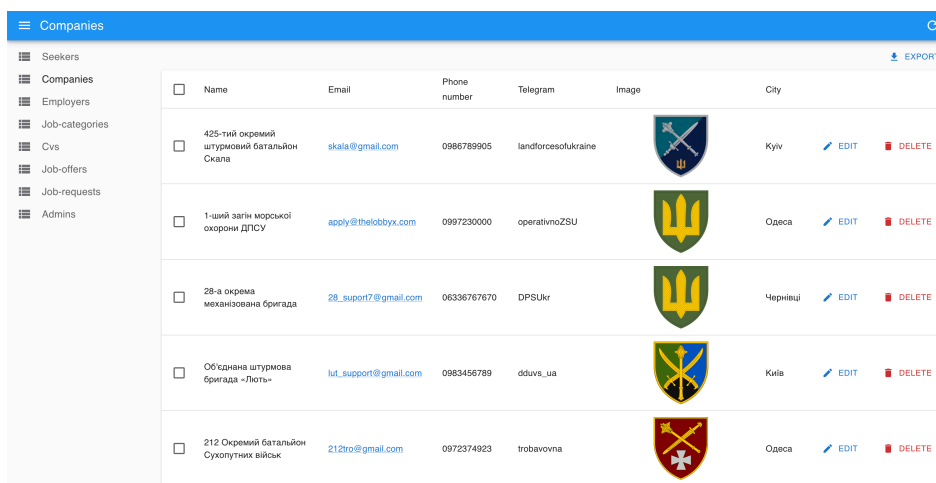


Рисунок 22. Панель адміністратора

## Висновок

Під час дослідження наявних веб-сайтів для пошуку роботи було виявлено значні недоліки, проаналізовано їх та зроблено висновки. Перед створенням застосунку враховано специфіка предметної області, а саме - пошук роботи в ЗСУ.

Перед початком роботи було досліджено найпопулярніші інструменти для розробки відповідних систем. Веб-сайт створено на основі React, Node.js, Express.js, NestJS, TypeORM та PostgreSQL, що є дуже поширеним стеком технологій. Робота була виконана в кілька основних етапів: порівняння схожих веб-сайтів, проектування бази даних, створення backend та розробка frontend. Застосунок побудований на основі REST, що забезпечує зручну взаємодію клієнта з сервером та надає можливість для масштабування в майбутньому. Веб-сайт має необхідний функціонал і є зручним не лише для кандидатів, але й для роботодавців. Дизайн стриманий та лаконічний, інтерфейс інтуїтивно зрозумілий кожному.

Оскільки на даний момент у держави є велика необхідність у військових, розробка веб-застосунку має прикладний характер. Під час створення застосунку було досягнуто основної мети розробки – спрощення етапу подачі заявки для добровольців. Даний проект має великий потенціал для розвитку в майбутньому, адже його актуальність з кожним днем лише зростає.

## Список використаних джерел

1. Офіційна сторінка сайту з пошуку роботи Work.us. URL: <https://www.work.ua/about-us/>
2. Сторінка Вікіпедія Lobby X. URL: [https://uk.wikipedia.org/wiki/Lobby\\_X](https://uk.wikipedia.org/wiki/Lobby_X)
3. The History of React.js: A Story of Innovation and Communit. URL: <https://www.linkedin.com/pulse/history-reactjs-story-innovation-community-l-anderson>
4. Advantages of React JS. URL: <https://www.linkedin.com/pulse/advantages-react-js-thescholarhat-v8c3f>
5. What is Dom in React? URL: <https://www.javatpoint.com/what-is-dom-in-react>
6. A Step-by-Step Guide to Creating a Component in React. URL: <https://medium.com/@AryanVora/a-step-by-step-guide-to-creating-a-component-in-react-151c0d875ee2>
7. Офійна документація React. Огляд хуків. URL: <https://uk.legacy.reactjs.org/docs/hooks-overview.html>
8. Angular Vs React Vs Vue: Which One To Choose. URL: <https://www.tatvasoft.com/blog/angular-vs-react-vs-vue/>
9. Node.js – Introduction. URL: [https://www.tutorialspoint.com/nodejs/nodejs\\_introduction.htm](https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm)
10. Офіційна документація bcrypt.js. URL: <https://www.npmjs.com/package/bcryptjs>
11. BrowserRouter in React. URL: <https://www.geeksforgeeks.org/browserrouter-in-react/>
12. What is Express.js? Everything you should know. URL: <https://kinsta.com/knowledgebase/what-is-express-js/>

13. Офіційна документація NestJS. URL: <https://docs.nestjs.com/>
14. Nest.js vs Express.js – Unveiling the Tale of Two Titans. URL: <https://www.linkedin.com/pulse/nestjs-vs-expressjs-unveiling-tale-two-titans-hamza-zaidi--u7euf/>
15. Документація Prisma: What is ORM? URL: <https://www.prisma.io/dataguide/types/relational/what-is-an-orm>
16. Офіційна документація TypeORM. URL: <https://typeorm.io/>
17. TypeORM - Quick Guide. URL: [https://www.tutorialspoint.com/typeorm/typeorm\\_quick\\_guide.htm](https://www.tutorialspoint.com/typeorm/typeorm_quick_guide.htm)
18. Офіційна документація PostgreSQL. URL: <https://www.postgresql.org/about/>
19. AWS What is PostgreSQL? URL: <https://aws.amazon.com/ru/rds/postgresql/what-is-postgresql/>
20. What is PostgreSQL? URL: <https://www.ibm.com/topics/postgresql>
21. Офіційна документація по react-admin. URL: <https://marmelab.com/react-admin/Readme.html>