

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
(повне найменування вищого навчального закладу)
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ
(повне найменування інституту, назва факультету (відділення))
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ І ТЕХНОЛОГІЙ
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

бакалавра

(освітньо-кваліфікаційний рівень)

на тему: «Розробка web-платформи для пошуку роботи»

Виконав: здобувач 4 курсу, групи 4ПРЗ
напряму підготовки (спеціальності)
121 «Інженерія програмного забезпечення»
(шифр і назва напряму підготовки, спеціальності)

Кучеренко Євгеній Сергійович
(прізвище та ініціали)

Керівник доцент Величко Ю. І.
(прізвище та ініціали)

Рецензент к.т.н., доцент Вишемирська С. В.
(прізвище та ініціали)

Хмельницький – 2025 рік

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут, факультет, відділення Інформаційних технологій та дизайну
Кафедра, циклова комісія Програмних засобів і технологій
Освітньо-кваліфікаційний рівень бакалавр
ОПП Програмна інженерія
(шифр і назва)
Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри, голова циклової комісії програмних засобів і технологій

_____ к.т.н., доцент О.Е. Огнева
«___» _____ 2025 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Кучеренко Євгеній Сергійович

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) «Розробка web-платформи для пошуку роботи»

керівник проекту (роботи) доцент Величко Юрій Ігорович

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

затверджені наказом вищого навчального закладу від «20» січня 2025 року № 96-с

2. Строк подання здобувачем проекту(роботи) 20.06.2025

3. Вихідні дані до проекту (роботи) _____

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

1. Дослідження та аналіз існуючих технологій та веб-додатків,

2. Аналіз вимог та розробка проектних специфікацій

3. Проектування програмного продукту

4. Розробка та застосування програмного продукту

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Створення комп'ютерної презентації.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

* КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання	09.02.2025	Виконано
2	Підбір літератури	10.02.2025 – 25.02.2025	Виконано
3	Аналіз предметної області	28.02.2025 – 10.03.2025	Виконано
4	Розробка концептуальної моделі	12.03.2025 – 22.03.2025	Виконано
5	Розробка та проектування системи	25.03.2025 – 30.03.2025	Виконано
6	Розробка та проектування бази даних	02.04.2025 – 10.04.2025	Виконано
7	Розробка інтерфейсу додатка	12.04.2025 – 20.04.2025	Виконано
8	Написання вихідного коду програми	21.04.2025 – 20.05.2025	Виконано
9	Оформлення пояснювальної записки	21.05.2025- 01.06.2025	Виконано
10	Захист кваліфікаційної роботи	20.06.2025	Виконано

Здобувач _____
(підпис)

Є.С. Кучеренко _____
(прізвище та ініціали)

Керівник проекту (роботи) _____
(підпис)

Ю.І.Величко _____
(прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 114 сторінок, 22 рисунка, 3 додатки, 15 джерел.

Мета роботи — Створення зручного, функціонального та візуально привабливого клієнтського веб-додатку для інноваційної платформи професійного пошуку роботи у сфері інформаційних технологій. Основна увага приділяється не лише реалізації сучасного дизайну, але й забезпеченню високого рівня доступності, інтуїтивної зрозумілості, кросплатформеності та адаптивності інтерфейсу для максимально комфортної взаємодії користувачів з системою в умовах стрімкого розвитку цифрових сервісів та зростання вимог до якості користувацького досвіду.

Об'єкт дослідження – інтерфейс веб-додатків для пошуку роботи в IT-сфері.

Предмет дослідження – методи проектування та реалізації клієнтської частини веб-застосунків з використанням фреймворку Next.js.

Методи дослідження – аналіз інтерфейсних рішень, проектування UI/UX, компонентний підхід до розробки, адаптивна вёрстка, REST API-інтеграція.

Результат роботи:

- Розроблено користувацький інтерфейс веб-платформи;
- Реалізовано адаптивний дизайн та мобільну підтримку;
- Налагоджено взаємодію з серверною частиною через API;
- Забезпечено зручність та інтуїтивну навігацію на сайті.

Наукова новизна роботи – застосування принципів сучасного UI/UX дизайну та підходів із JAMstack-архітектури у розробці клієнтської частини веб-застосунку для пошуку роботи в IT.

Ключові слова: *React, Next.js, серверний рендеринг, SEO, TypeScript, веб-інтерфейс, пошук роботи, UX/UI, Next.js, адаптивний дизайн, інтерактивний інтерфейс, фронтенд-розробка, API.*

АНОТАЦІЯ

Кваліфікаційна робота бакалавра складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. У першому розділі досліджено ключові підходи до розробки інтерфейсів у сучасному вебсередовищі. Проаналізовано переваги та недоліки клієнтського (CSR) і серверного (SSR, SSG, ISR) рендерингу в контексті продуктивності, SEO та досвіду користувача. Розглянуто особливості фреймворків React і Next.js, наведено їх порівняльну характеристику. Обґрунтовано вибір архітектури на основі Next.js як оптимального рішення для створення масштабованих SPA/MPA-застосунків.

У другому розділі визначено функціональні та нефункціональні вимоги до застосунку. Сформульовано технічне завдання, виділено основні модулі системи, зокрема: маршрутизацію, типізацію, управління станом, SEO-оптимізацію та перевірку якості коду. Наведено специфікацію проєктної структури, зокрема структуру директорій, підхід до розділення компонентів і типів рендерингу для різних сторінок.

У третьому розділі реалізовано ключові програмні компоненти вебзастосунку. Створено шаблони сторінок з використанням App Router, реалізовано навігацію, динамічні маршрути, та впроваджено типізацію з TypeScript. Налаштовано ESLint і Prettier для забезпечення чистоти коду. Здійснено підключення Redux для глобального управління станом. Упроваджено підтримку SVG-графіки і відповідність принципам UX/UI-дизайну.

Четвертий розділ присвячено тестуванню, демонстрації функціональності та SEO-аналізу. Надано приклади рендерингу сторінок із різними типами завантаження, проаналізовано результати перевірки швидкодії та відповідності пошуковим вимогам. Розглянуто поведінку застосунку у реальних умовах, а також вплив використаних технологій на

якість інтерфейсу та досвід користувача.

У результаті виконано повний цикл розробки клієнтської частини вебзастосунку, що відповідає вимогам сучасних вебтехнологій та стандартів розробки. Робота демонструє інтеграцію найкращих практик з архітектури інтерфейсів, типізації, оптимізації та дизайну.

.

ABSTRACT

The bachelor's qualification work consists of an introduction, four sections, conclusions, a list of sources used and appendices.

The first section examines key approaches to interface development in a modern web environment. The advantages and disadvantages of client (CSR) and server (SSR, SSG, ISR) rendering in the context of performance, SEO and user experience are analyzed. The features of the React and Next.js frameworks are considered, their comparative characteristics are given. The choice of architecture based on Next.js as the optimal solution for creating scalable SPA/MPA applications is justified.

The second section defines functional and non-functional requirements for the application. The technical task is formulated, the main modules of the system are highlighted, in particular: routing, typing, state management, SEO optimization and code quality control. The specification of the project structure is given, in particular the directory structure, the approach to separating components and rendering types for different pages.

The third section implements the key software components of the web application. Page templates are created using App Router, navigation, dynamic routes are implemented, and typing with TypeScript is implemented. ESLint and Prettier are configured to ensure code cleanliness. Redux is connected for global state management. SVG graphics support and compliance with UX/UI design principles are implemented.

The fourth section is devoted to testing, functionality demonstration, and SEO analysis. Examples of rendering pages with different types of loading are provided, the results of speed testing and compliance with search engine requirements are analyzed. The behavior of the application in real conditions is considered, as well as the impact of the technologies used on the quality of the interface and user experience.

As a result, a full cycle of development of the client part of the web application is completed, which meets the requirements of modern web technologies and development standards. The work demonstrates the integration of best practices in interface architecture, typing, optimization, and design.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	11
ВСТУП	12
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ ІНТЕРФЕЙСУ ВЕБ-ПЛАТФОРМИ.....	15
1.1 Аналіз предметної області з позиції користувача (UX)	15
1.1.1 Типи користувачів платформи: кандидат та HR-компанії	15
1.1.2 Основні сценарії використання платформи користувачами .	17
1.2 Аналіз сучасних технологій для розробки фронтенду	18
1.2.1 Порівняння фронтенд-фреймворків: Next.js, React.js, Angular	19
1.2.2 Огляд технологій обраного фреймворка Next.js.....	21
1.2.3 CSS-фреймворки та бібліотеки: Tailwind CSS.....	23
1.2.4 Огляд бібліотек для валідації форм (Zod та альтернативи)...	23
1.2.5 Інструменти для розробки та підтримки проєкту: ESLint, Prettier.....	24
1.3 Порівняння методів управління станом додатка	24
1.3.1 React Context: суть, переваги та сценарії використання	25
1.3.2 Redux: огляд переваг та недоліків.....	25
1.3.3 Zustand та MobX як прості альтернативи	26
1.3.4 Вибір рішення для проєкту.....	26
1.4 Аналіз оптимізації інтерфейсу для продуктивності та SEO	27
1.4.1 Важливість продуктивності для користувацького досвіду ...	28

1.4.2 Оптимізація для SEO: чому це критично важливо для платформи пошуку роботи	28
1.4.3 Як Next.js сприяє SEO-оптимізації	29
1.4.4 Технічна доступність і структура HTML документа	29
1.4.5 Метрики та аудит (Lighthouse, Core Web Vitals)	30
1.5 Висновки до розділу 1	30
РОЗДІЛ 2. АНАЛІЗ ВИМОГ ДО КЛІЄНТСЬКОЇ ЧАСТИНИ ТА ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ	32
2.1. Вимоги до клієнтської частини веб-платформи.....	32
2.2 Вимоги до UI/UX дизайну.....	35
2.3 Специфікації основних функцій інтерфейсу	42
2.4 Висновки до розділу 2	46
РОЗДІЛ 3. ПРОЄКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	47
3.1. Архітектурна модель клієнтської частини застосунку.....	47
3.2 Опис компонентної моделі.....	50
3.3. Розробка моделей даних	52
3.4. Використання локального збереження даних на стороні клієнта.....	55
3.5. Реалізація основних компонентів	56
3.6: UX-інтеракції: валідація, сповіщення, статуси	70
3.7. Інтеграція з API та типізація запитів у клієнтській частині.....	75
3.8. Висновки до розділу 3	76
РОЗДІЛ 4. РЕЗУЛЬТАТИ РЕАЛІЗАЦІЇ КЛІЄНТСЬКОЇ ЧАСТИНИ ПРОГРАМНОГО ПРОДУКТУ	78
4.1. Опис виконаних робіт	78
4.2. Перевірка працездатності реалізованих компонентів	79

4.3. Висновки до розділу 4	91
ВИСНОВОК	93
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	95
ДОДАТОК А	97
Файл Header.tsx:	97
ДОДАТОК В	100
Файл UkraineMap.tsx:	100
Файл svgUkraineMap.css:	104
ДОДАТОК С	105
Файл page.tsx:	105
Файл components/vacancy/VacancyDetails.tsx:	107
Файл components/vacancy/CompanyCard.tsx:	109
Файл components/vacancy/SimilarJobs.tsx:	112

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API (Application Programming Interface) – інтерфейс прикладного програмування, набір методів для взаємодії між програмними компонентами.
UI (User Interface) – інтерфейс користувача, візуальна частина програми, з якою взаємодіє користувач.

UX (User Experience) – досвід користувача, загальне враження від використання інтерфейсу.

HTML (HyperText Markup Language) – мова розмітки, що використовується для створення структури веб-сторінок.

CSS (Cascading Style Sheets) – каскадні таблиці стилів, мова для оформлення зовнішнього вигляду HTML-елементів.

JS (JavaScript) – мова програмування для створення інтерактивності на веб-сторінках.

SSR (Server-Side Rendering) – рендеринг на стороні сервера, спосіб формування HTML-коду до його надсилання клієнту.

SPA (Single Page Application) – односторінковий застосунок, веб-додаток, що динамічно оновлює контент без перезавантаження сторінки.

SEO (Search Engine Optimization) – оптимізація веб-сторінок для покращення їх видимості у пошукових системах.

SSG (Static Site Generation) – статична генерація сторінок, при якій HTML створюється під час збірки, а не під час запиту користувача.

REST (Representational State Transfer) – архітектурний стиль взаємодії між клієнтом і сервером через HTTP з використанням стандартних методів (GET, POST тощо).

HR (Human Resources) – відділ управління персоналом, у контексті дипломної – компанії або фахівці, які займаються підбором кадрів.

ВСТУП

У сучасних умовах стрімкого розвитку цифрових технологій особливу роль у комунікації між шукачами роботи та роботодавцями відіграють веб-платформи для професійного працевлаштування. Якість користувацького досвіду під час взаємодії з такими платформами безпосередньо впливає на ефективність пошуку та відбору кадрів. Традиційні системи, попри функціональність, часто мають застарілі або незручні інтерфейси, не оптимізовані для мобільних пристроїв, а також не відповідають сучасним вимогам до SEO та швидкості роботи.

Клієнтська частина веб-додатку є критично важливою складовою таких систем, оскільки саме через неї користувач взаємодіє з усіма основними функціями. Актуальним є створення зручного, адаптивного та SEO-оптимізованого інтерфейсу з урахуванням сучасних стандартів веб-розробки, що забезпечує не лише привабливість і зрозумілість, а й технічну ефективність роботи додатку.

Кваліфікаційна робота присвячена розробці клієнтської частини веб-платформи для професійного пошуку роботи у сфері інформаційних технологій з використанням фреймворку Next.js.

Актуальність теми. Значна частина взаємодії користувача з системою працевлаштування відбувається через браузер. Умови сучасного ринку вимагають високої швидкодії, зручності навігації, підтримки мобільних пристроїв та пошукової оптимізації (SEO). Реалізація таких функціональних можливостей у клієнтській частині безпосередньо впливає на привабливість і конкурентоспроможність веб-продукту. Тому розробка ефективного, адаптивного та доступного інтерфейсу є актуальною задачею у сфері прикладного програмування.

Мета роботи. Розробка інтерфейсу користувача веб-платформи для професійного пошуку роботи в IT-сфері з акцентом на UX/UI-дизайн,

адаптивність, зручність, інтерактивність і SEO-оптимізацію за допомогою інструментів сучасної фронтенд-розробки.

Об’єкт дослідження – веб-платформи у сфері працевлаштування.

Предмет дослідження – методи та технології реалізації клієнтської частини веб-додатків з використанням Next.js.

Методи дослідження – аналіз існуючих рішень у сфері веб-інтерфейсів, проектування UI/UX, використання фреймворку Next.js для SSR та SPA-логіки, адаптивна верстка, SEO-оптимізація, REST API-інтеграція.

Основні завдання роботи:

- проаналізувати вимоги до сучасного веб-інтерфейсу для системи працевлаштування;
- обґрунтувати вибір фреймворку Next.js як інструменту для створення клієнтської частини;
- розробити дизайн користувацького інтерфейсу відповідно до принципів UX/UI;
- реалізувати адаптивну верстку з підтримкою мобільних пристроїв;
- забезпечити пошукову оптимізацію сторінок (SEO);
- реалізувати інтеграцію з серверною частиною через API-запити;
- протестувати інтерфейс на предмет зручності, стабільності та сумісності з популярними браузерами.

Наукова новизна одержаних результатів. Уперше застосовано поєднання SSR і SPA-архітектури у контексті платформи для пошуку роботи, що дозволяє досягти високої продуктивності, зручності користування та оптимізації для пошукових систем. Запропоновано підхід до створення адаптивного інтерфейсу з урахуванням сценаріїв використання у сфері IT-працевлаштування.

Практичне значення одержаних результатів. Результатом роботи є повноцінна клієнтська частина веб-платформи, готова до інтеграції з серверною логікою. Розроблений інтерфейс відповідає сучасним вимогам до

якості веб-додатків і може бути використаний у реальних умовах для взаємодії з кандидатами та роботодавцями у сфері інформаційних технологій. Створений продукт може бути основою для комерційного впровадження або як шаблон для аналогічних рішень у суміжних галузях.

Структура: робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел, додатків. Кваліфікаційна робота бакалавра складається з 114 сторінок, 22 рисунка, 3 додатків, 15 джерел.